Dissertation

Eine kanonische Form zur Darstellung äquivalenter Codes

– Computergestützte Berechnung und ihre Anwendung in der Codierungstheorie, Kryptographie und Geometrie –

Von der Universität Bayreuth
zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Abhandlung

von

Thomas Feulner

geboren am 16. Februar 1982
in Bayreuth

Vorwort

Die vorliegende Arbeit entstand in den Jahren 2008 bis 2013 an der Universität Bayreuth unter Betreuung von Herrn Prof. Dr. Adalbert Kerber. Seiner beherzten Initiative ist es zu verdanken, dass ich nach dem Diplom meine Forschungsarbeit zur Kanonisierung linearer Codes fortsetzen konnte. Meinem Doktorvater möchte ich für seine Unterstützung und das allzeit entgegengebrachte Vertrauen herzlich danken.

Als weiterer Glücksfall erwies sich für mich, dass nach der Emeritierung von Herrn Prof. Dr. Kerber die Arbeitsgruppe durch Prof. Dr. Alfred Wassermann und PD Dr. Axel Kohnert fortgeführt wurde. Ihnen ist es zu verdanken, dass die Finanzierung meiner Arbeit, zunächst über ein Stipendium der Bayerischen Eliteförderung und schließlich im Rahmen des DFG Schwerpunktprogramms 1489, sichergestellt werden konnte. Zutiefst betroffen bin ich immer noch darüber, dass ich Dir, Axel, diesen Dank nicht mehr persönlich aussprechen kann.

Für eine konstruktive und stets unkomplizierte Zusammenarbeit bedanke ich mich bei allen weiteren Kollegen der Arbeitsgruppe, die mit hilfreichen Diskussionen und Ratschlägen zu dieser Arbeit beitrugen. Insbesondere meinen Bürokollegen Michael Kiermaier und Johannes Zwanzger sowie Elvira Rettner möchte ich aber hiermit nochmals gesondert, für die angenehme Arbeitsatmosphäre und ein allzeit offenes Ohr bei Fragen und Problemen danken.

Meine Eltern, Roswitha und Gerhard, haben mich mit Aufnahme des Studiums bis heute kontinuierlich und bedingungslos unterstützt. Meine Freundin Melanie stand mir immer liebevoll motivierend auf dem langen Weg bis zur Promotion zur Seite. Für das entgegengebrachte Verständnis möchte ich mich von ganzem Herzen bei ihnen bedanken.

Eckersdorf, im März 2014 *Thomas Feulner*

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Algorithmenverzeichnis

Symbolverzeichnis

${}_GX$	Eine Gruppenoperation von G auf einer Menge X, Seite 7
$G\backslash\backslash X$	Die Bahnen einer Gruppenoperation ${}_GX$, Seite 7
$\mathrm{Fix}_G(X)$	Die Fixpunkte der Operation von G auf X, Seite 7
$\mathcal{L}(G)$	$\{H \mid H \leq G\}$, Seite 7
$\mathcal{C}(G)$	$\{Hg \mid H \leq G,\ g \in G\}$, Seite 8
$\mathrm{CF}_G(x)$	Der kanonische Repräsentant einer Bahn Gx, Seite 8
$\mathrm{TR}_G(x)$	Ein Transporterelement zu einer Kanonisierung CF_G, Seite 8
$\mathrm{Stab}_G(x)$	Der Stabilisator von $x \in X$ zu einer Gruppenoperation ${}_GX$, Seite 9
Can_G^X	Ein Kanonisierer zu einer Gruppenoperation ${}_GX$, Seite 9
$[n]$	$\{0, \ldots, n-1\}$, Seite 9
$\mathrm{Rad}(R)$	Das Jacobson-Radikal eines Rings R, Seite 13
θ	Ein Erzeuger von $\mathrm{Rad}(R)$, Seite 14
$\mathrm{per}(x)$	Die Periode eines Elements $x \in {}_RM$, Seite 14
$\mathrm{ht}(x)$	Die Höhe eines Elements $x \in {}_RM$, Seite 14
$\mathrm{shp}(M)$	Der Umriss eines R-Linksmoduls M, Seite 15
$\mathrm{rg}(M)$	Der Rang eines R-Linksmoduls M, Seite 15
λ	Ein fest vorgegebener Umriss der betrachteten linearen Codes, Seite 16
$R^{k\times n,\lambda}$	Die Menge aller Generatormatrizen zu allen linearen Codes der Länge n vom Umriss $\lambda = (\lambda_0, \ldots, \lambda_{k-1})$, Seite 16
$T(x, G)$	Der Suchbaum zur Definition von Can_G^X, Seite 39
$I(x, Hg)$	Die Partitionierungsvorschrift im Suchbaum $T(x, G)$, Seite 39
$V(x, Hg)$	Die Verfeinerungsvorschrift im Suchbaum $T(x, G)$, Seite 40
$L(x, G)$	Die Menge aller Gruppenelemente, welche Blätter von $T(x, G)$ definieren, Seite 40
$L_0(x, G)$	$\{g \in L(x, G) \mid gx = \mathrm{CF}_G(x)\}$, Seite 41
$B(x, Hg)$	Die Bewertung des Knotens Hg im Suchbaum $T(x, Hg)$, Seite 42
$\mathcal{F}^{\mathfrak{p}}$	Die Färbung der Koordinaten bezüglich der Partition $\mathfrak{p}$, Seite 51
$\mathfrak{P}$	Eine kanonische Partition zu $[n]$, Seite 51

$S_\mathfrak{p}$	Die (kanonische) Young-Untergruppe zur Partition $\mathfrak{p}$ von $[n]$, Seite 51
$\overline{T}(x, G)$	Der Backtrackbaum, welcher auch die Zwischenschritte bei der Gewinnung von $T(x, G)$ angibt, Seite 60
$\mathfrak{P}_0$	Eine fest vorgeschriebene kanonische Partition von $[n]$ (später $S_{\mathfrak{P}_0} = \mathrm{Stab}_{S_n}(\mu)$), Seite 61
$\overline{\mathcal{C}}(G \rtimes S_{\mathfrak{P}_0})$	$\left\{H \rtimes S_\mathfrak{P}(g;\pi) \in \mathcal{C}(G \rtimes S_{\mathfrak{P}_0}) \mid H \rtimes S_\mathfrak{P} \in \overline{\mathcal{L}}(G \rtimes S_{\mathfrak{P}_0})\right\}$, Seite 61
$\overline{\mathcal{L}}(G \rtimes S_{\mathfrak{P}_0})$	$\{H \rtimes S_\mathfrak{P} \mid H \leq G, \mathfrak{P} \preceq \mathfrak{P}_0 : H \rtimes S_\mathfrak{P} \in \mathcal{L}(G \rtimes S_{\mathfrak{P}_0})\}$, Seite 61
$G^{(f,x)}$	$\mathrm{Stab}_G(\mathrm{CF}_G(\Pi_f(x)))$, Seite 65
$\overline{F}(x, H \rtimes S_\mathfrak{P}(g;\pi), i)$	Die Folge der bereits zur inneren Kanonisierung benutzten Koordinaten, Seite 65
$R[X;\sigma]$	Der Schiefpolynomring über R zu $\sigma \in \mathrm{Aut}(R)$, Seite 71
T	Eine fest gewählte Teichmüller-Menge von R, Seite 71
ξ	Ein fest gewählter Erzeuger von T^*, Seite 71
$\mathrm{coeff}^{(i)}(a)$	Der i-te Koeffizient in der θ-adischen Entwicklung von $a \in R$, Seite 72
$\mathrm{coeff}(a)$	Die Koeffizientenmatrix der (ξ, θ)-adischen Entwicklung von $a \in R$, Seite 72
$\mathrm{coeff}^{(i,j)}(a)$	$\mathrm{coeff}(a)_{i,j}$, Seite 73
$R^{(i,j)}$	$\left\{a \in \mathrm{Rad}(R)^j \mid \forall\ 0 \leq \nu < i : \mathrm{coeff}^{(\nu,j)}(a) = 0\right\}$, Seite 74
$R^{*(i,j)}$	$1 + R^{(i,j)}$, Seite 75
$\mathrm{GR}(p^m, r)$	Der Galois-Ring der Charakteristik p^m und Kardinalität p^{rm}, Seite 76
τ	Der Frobenius-Automorphismus des Koeffizientenrings S von R, Seite 76
e	Die eindeutige Zahl $e \in [r]$ mit $\theta a = \tau^e(a)\theta$, Seite 77
χ_ψ^ω	Die Bijektion von R nach S mit $\chi_\psi^\omega(\xi) = \psi$ und $\chi_\psi^\omega(\theta) = \omega$, welche durch Fortsetzung über die (ξ, θ)-adischen Entwicklung definiert wird, Seite 78
$\mathrm{Inn}(R)$	Die Gruppe der inneren Automorphismen, Seite 79
$Z(R)$	Die Gruppe der zentralen Einheiten von R, Seite 79
Aut_T	$\{\alpha \in \mathrm{Aut}(R) \mid \alpha(T) = T\}$, Seite 79
Aut_ξ	$\{\alpha \in \mathrm{Aut}(R) \mid \alpha(\xi) = \xi\}$, Seite 79
$\mathrm{Out}(R)$	$\mathrm{Aut}(R)/\mathrm{Inn}(R)$, Seite 79
$\mathrm{GL}_\lambda(R)$	Diejenige Untergruppe von $\mathrm{GL}_k(R)$, welche auf der Menge aller Generatormatrizen vom Umriss λ operiert und deren Bahnen in Bijektion mit den Codes vom Umriss λ stehen, Seite 88

$N_\lambda(R)$	Der Kern der Operation von $\mathrm{GL}_\lambda(R)$ auf $R^{k\times n,\lambda}$, Seite 88
μ	Eine monoton fallende, fest vorgegebene Folge aus $[m+1]^n$, Seite 93
$R^{k\times n,\lambda,\mu}$	Die Menge aller Generatormatrizen vom Umriss λ mit vorgegebener Periode μ_i der i-ten Spalten, Seite 93
$(R^*)^\mu$	Der Normalteiler $\bigtimes_{j\in[n]}(1+\mathrm{Rad}(R)^{\mu_j})$ von R^{*n}, Seite 94
$G^{(\lambda,\mu)}$	$\big(\big(\mathrm{GL}_\lambda(R)/N_\lambda(R)\big)\times\big((R^*)^n/(R^*)^\mu\big)\big)\rtimes \mathrm{Aut}_T$, Seite 96
$\mathfrak{p}^\Gamma$	Die feinste Partition $\mathfrak{p}$ von $[k]$ sd. die Trägermengen aller Spalten von $\Gamma\in R^{k\times n,\lambda,\mu}$ in genau einem Block $P\in\mathfrak{p}$ liegen, Seite 97
$\mathrm{Cols}_P(\Gamma)$	$\{j\in[n]\mid \mathrm{supp}(\Gamma_{*,j})\subseteq P\}$, Seite 97
$\Phi^{(i,x,\Gamma)}$	Vgl. Beweis zu 5.1.43, Seite 105

1. Einleitung

In dieser Dissertation werden Algorithmen zur Kanonisierung verschiedener Klassen von Codes entwickelt und angewandt. Unter einer *Kanonisierung* zu einer Gruppenoperation von G auf X verstehen wir hierbei eine Abbildung $\mathrm{CF}_G : X \to X$, die jedem Element $x \in X$ einen eindeutig bestimmten Repräsentanten $\mathrm{CF}_G(x)$ seiner Bahn Gx zuordnet. Wir nennen diesen den *kanonischen Repräsentanten* oder auch die *kanonische Form* von Gx beziehungsweise von x.

Der Wunsch nach einem solchen Werkzeug in der Codierungstheorie wurde bereits 1960 von D. Slepian in der Arbeit [69] geäußert:

> *„The task of analyzing group codes would be greatly simplified if a canonical form could be found for each equivalence class of Ω-matrices. That is, for a given n and k, we should like to be able to write down one generator matrix from each equivalence class. This would provide a simple means of describing each of the essentially different (n, k)-codes.“*

Dabei verwendet D. Slepian den Begriff „group code“ für einen binären, linearen Code, d.h. einen Untervektorraum des $\mathbb{F}_2^n$, und „Ω-matrices“ für Generatormatrizen[1]. Die Äquivalenz von linearen Codes über dem Alphabet $\mathbb{F}_q$ definiert man für gewöhnlich über die Gruppenoperation der linearen Hamming-Isometrien von $\mathbb{F}_q^n$ auf der Menge aller Untervektorräume.

Eng verwandt mit der Fragestellung nach einem kanonischen Repräsentanten ist auch die Berechnung des Stabilisators. So entwickelte J. Leon [51] im Jahre 1982 einen Algorithmus, welcher in der Lage ist, diese Gruppe für einen gegebenen linearen Code über einem endlichen Körper $\mathbb{F}_q$ zu berechnen. Weiterhin ist sein Verfahren auch dazu geeignet, zu zwei gegebenen linearen Codes die Frage der Äquivalenz zu entscheiden. Man findet diesen Algorithmus in den Computeralgebrasystemen GAP [28] (Paket GUAVA [15]), Magma [53] oder Sage [70] implementiert.

Offensichtlich hat der Äquivalenztest aber im Gegensatz zur Kanonisierung einen entscheidenden Nachteil: Müssen größere Familien linearer Codes auf Äquivalenz hin untersucht werden – etwa im Rahmen einer vollständigen Klassifikation –, so muss gegebenenfalls der Algorithmus für alle Paare aufgerufen werden. Dahingegen transformiert der Kanonisierungsansatz jede Instanz auf ihren kanonischen Repräsentanten und führt anschließend nur noch eine einfache Sortierung der kanonischen Repräsentanten durch.

[1] Dies sind Matrizen, deren Zeilen eine Basis des linearen Codes bilden.

Auch die von N. Sendrier [66] vorgeschlagene Modifikation an J. Leons Algorithmus beinhaltet immer noch diesen entscheidenden Nachteil. Diese Arbeit ist jedoch, wie wir später sehen werden, für komplexitätstheoretische Betrachtungen von Interesse.

Schließlich soll auch nicht unerwähnt bleiben, dass es neben den oben genannten direkten Ansätzen auch Lösungsvorschläge gibt, welche das Problem auf die Berechnung von kanonischen Repräsentanten für Graphen zurückführen, etwa [60] oder das Software-Paket „Q-Extension“ [7]: Bei sorgfältiger Modellierung sind zwei gegebene lineare Codes genau dann äquivalent, falls sie auf isomorphe Graphen abgebildet werden. Anschließend werden kanonische Repräsentanten dieser Graphen über die verfügbaren, sehr effizienten Kanonisierungsverfahren – etwa *nauty* [55] – berechnet. Diese haben aber nicht zwingend lineare Codes zum Urbild und dienen daher nur als eindeutige Zertifikate für die Äquivalenzklassen. Jedoch können unter Umständen die derart erzielten Graphen einen exponentiellen Speicherbedarf, im Vergleich zur Darstellung des Codes über eine Generatormatrix, aufweisen. Dies stellt ebenfalls eine erheblichen Nachteil dar, welchem – etwa wieder im Rahmen einer vollständigen Klassifikation – eine entscheidende Rolle zukommen kann.

In meiner Diplomarbeit [22] wurde, nach meinem Wissen erstmals, ein Algorithmus zur Bestimmung kanonischer Generatormatrizen äquivalenter linearer Codes entwickelt und in der Arbeit [24] fortentwickelt. Weiterhin wird in diesen Arbeiten auch ein allgemeinerer und natürlicherer Äquivalenzbegriff (es operiert die Gruppe der *semilinearen Isometrien* auf $\mathbb{F}_q^n$) für lineare Codes verwendet.

Seit D. Slepians über 50 Jahre altem Wunsch nach kanonischen Repräsentanten für binäre, lineare Codes hat sich die Codierungstheorie in vielfältige Richtungen entwickelt. In dieser Arbeit werden wir nun diese neueren Entwicklungen aufgreifen und geeignete Werkzeuge zur Definition effizienter Kanonisierer zur Verfügung stellen:

- Ein Zweig der Codierungstheorie schwächt die Forderung an die Teilmenge $C \subseteq \mathbb{F}_q^n$ ab[2]: Bildet C eine Untergruppe der additiven Gruppe von $\mathbb{F}_q^n$, so nennt man C auch einen *additiven Code*. Insbesondere für $q = 4$ spielen diese Codes eine wichtige Rolle für die Fehlerkorrektur auf Quantencomputern, siehe [9].

  Ist $\mathbb{F}_{q'}$ ein echter Teilkörper von $\mathbb{F}_q$ und $C \subseteq \mathbb{F}_q^n$ ein $\mathbb{F}_{q'}$-linearer Teilraum, so nennen wir C auch einen $\mathbb{F}_{q'}$*-linearen* $\mathbb{F}_q$*-Code*. Der $\mathbb{F}_{q'}$-lineare $\mathbb{F}_q$-Code C ist also immer auch ein additiver Code. Umgekehrt ist jeder additive Code gleichermaßen auch ein $\mathbb{F}_p$-linearer $\mathbb{F}_q$-Code, für den Primkörper $\mathbb{F}_p$ von $\mathbb{F}_q$.

- Ein weiterer Zweig hält an der linearen Struktur der Codes fest, lässt aber allgemeinere Alphabete (und Metriken) zu. Statt des Körpers $\mathbb{F}_q$ werden *endliche Kettenringe* R oder sogar Frobenius-Ringe betrachtet und (R-)lineare Codes als R-Linksuntermoduln von R^n definiert. Diese Verallgemeinerungen wurden motiviert durch die Arbeiten [36, 57] über die $\mathbb{Z}_4$-Linearität[3] gewisser Serien von nicht-

[2] Stellt man keine Forderung an C, so nennt man C auch einen *Blockcode*.

[3] Die Codes sind Bilder von $\mathbb{Z}_4$-linearer Codes unter der sogenannten Gray-Abbildung von $\mathbb{Z}_4$ nach $\mathbb{F}_2^2$.

linearen, binären Codes, welche bessere[4] Parameter besitzen als ihre linearen Gegenstücke. In [32, 37] findet man eine kurze Einführung in dieses Thema.

- Neueste Entwicklungen beschäftigen sich aufgrund geänderter technischer Voraussetzungen mit anderen Übertragungsmodellen. Im klassischen Shannon-Modell wird von einem Sender und einem Empfänger ausgegangen, welche über *einen direkten* Übertragungsweg miteinander kommunizieren, etwa beim Lesen einer CD, der Übertragung von Daten eines Satelliten zur Erde, usw. Die Codeworte werden hierbei sequentiell als Zeichenketten übertragen. Hier kommen die oben beschriebenen linearen bzw. additiven Codes zum Einsatz.

  In diesen geänderten Kommunikationsmodellen werden Nachrichten über ein (unbekanntes) Netzwerk von Knoten mit *mehreren Sendern, Empfängern und Zwischenknoten* übertragen. Es stehen also für ein gegebenes Paar aus Sender und Empfänger mehrere potentielle Übertragungswege zur Verfügung. Jedoch muss auch davon ausgegangen werden, dass bestimmte Teilabschnitte eines Weges von weiteren Sendevorgängen belegt werden könnten. Um solche gegenseitigen Blockaden zu umgehen und in der Gesamtheit höhere Übertragungsraten zu erzielen, werden die Informationen an den Knoten aufgebrochen und auch wieder zusammengeführt, vgl. [47]. In diesem Fall bilden nun Unterräume von $\mathbb{F}_q^n$ die zu übertragenden Codeworte, wobei zwischen den einzelnen Knoten des Netzwerks immer nur je ein Vektor aus $\mathbb{F}_q^n$ gesendet wird. Solche Codes – also Teilmengen aller Untervektorräume von $\mathbb{F}_q^n$ – nennt man *Network-Codes*.

  Als Anwendungsbeispiel dienen das Internet, Funknetzwerke, die dezentrale Speicherung von Daten auf mehreren Servern (distributed storage) oder der dezentrale Austausch von Informationen (filesharing).

- Es gibt bereits erste Untersuchungen [17], welche sich mit Network-Codes über endlichen Kettenringen beschäftigen. Dieser Übergang ist auch aus physikalischen Gesichtspunkten motiviert. Genauso werden auch additive Codes über Kettenringen betrachtet.

Über die Operation der Isometriegruppe des Umgebungsraums der untersuchten Codes – beziehungsweise einer Untergruppe (z.B. semilineare Isometrien), deren Elemente die vorgeschriebene Struktur (z.B. die Linearität) der Codes respektieren – erhalten wir stets einen natürlichen Äquivalenzbegriff auf der Menge aller Codes. Äquivalenzklassen werden also über die Bahnen dieser Gruppenoperation definiert und wir können über einen Algorithmus zur Berechnung kanonischer Repräsentanten eindeutige Vertreter der Äquivalenzklassen bestimmen.

In dieser Dissertation werden wir den Fokus auf die Berechnung kanonischer Repräsentanten R-linearer Codes für beliebige endliche Kettenringe R legen. Unser Verfahren ist eine Weiterentwicklung des Algorithmus zur Kanonisierung $\mathbb{F}_q$-linearer Codes, welcher

[4] Die Codes erreichen bei gleicher Kardinalität und Länge eine höhere Minimaldistanz.

in meiner Diplomarbeit [22] bereits entworfen wurde. Die zugrunde liegenden Ideen werden wir in einer Allgemeinheit ausführen, die es dann erlaubt, unser Vorgehen auch auf weitere Klassen von Codes bzw. auf weitere Gruppenoperationen leicht zu übertragen.

Diese Dissertation ist nun folgendermaßen aufgebaut: Zunächst werden wir in dem anschließenden Kapitel 2 wichtige Grundlagen über Gruppen und endliche Gruppenoperationen (Abschnitt 2.1), Graphen (2.2), Kettenringe R und R-lineare Codes (2.3) zusammenfassen und die Notation festlegen. In diesem Kapitel klären wir auch die Frage zur Komplexität der Berechnung kanonischer Repräsentanten beziehungsweise eines Tests auf die Äquivalenz zweier gegebener R-linearer Codes, siehe Abschnitt 2.4. Hierdurch rechtfertigen wir unser Vorgehen im weiteren Verlauf der Arbeit.

Das Kapitel 3 beschreibt dann verschiedene Ansätze zur Berechnung kanonischer Repräsentanten einer beliebigen Gruppenoperation von G auf X. Das von uns bevorzugte Verfahren beruht auf der Beschreibung eines Backtrackalgorithmus, welcher erstmals für die Kanonisierung von Graphen [55] entwickelt wurde. Verallgemeinerungen auf beliebige Gruppenoperationen werden in [35] und [42] gegeben. Der Abschnitt 3.2 vereint beide Quellen und beschreibt den Backtrackbaum aus [42] über das Homomorphieprinzip für Gruppenoperationen, siehe [35, 50]. Weiterhin wird eine Bewertung der Knoten des Backtrackbaums definiert, welche das frühzeitige Abschneiden von Teilbäumen ermöglicht.

Über die Untergruppe aller bislang bekannten Automorphismen des zu kanonisierenden Objekts $x \in X$ wird ein weiterer Test entwickelt, welcher es ebenfalls erlaubt, Teilbäume des Backtrackbaums von der Suche auszuschließen. Hierzu wird der Hilfssatz 3.3.3 aus der Arbeit [35] verschärft, um bestmögliche Resultate zu erzielen. Diese Verallgemeinerung wurde bereits in [22] angegeben, jedoch nicht im Zusammenhang einer allgemeinen Gruppenoperation von G auf X formuliert. Zum Abschluss des Kapitels wird auf Sonderfälle für die Gruppe G eingegangen.

Zur Vorbereitung der Kanonisierung R-linearer Codes wird die Struktur eines Kettenrings R in Kapitel 4 untersucht und eine Totalordnung definiert. Insbesondere wird in diesem Kapitel auch auf die Struktur der additiven Gruppe, der multiplikativen Gruppe und der Automorphismengruppe über die Angabe von Normalreihen eingegangen. Das Kapitel 5 geht dann schließlich auf lineare Codes über einem gegebenen Kettenring ein. Zunächst werden weitere wichtige Grundlagen gelegt. Anschließend formuliert man die Äquivalenz R-linearer Codes der Länge n über eine Gruppenoperation einer Gruppe $G \rtimes S_n$ auf der Menge der Generatormatrizen. Für diese Operation wird schließlich ein Kanonisierer entwickelt.

Das folgende Kapitel 6 gibt Anwendungsbeispiele für den entwickelten Kanonisierer und beschreibt mögliche Modifikationen, um diesen auch in der Kryptographie oder für Network-Codes und $\mathbb{F}_q$-lineare $\mathbb{F}_{q^r}$-Codes anzuwenden. In mehreren Beispielen wird gezeigt, dass eine effiziente Kanonisierung häufig im Rahmen einer Klassifikation benötigt wird und dort das Herzstück bildet. So konnte etwa über eine vollständige Klassifikation in 217 Fällen die Existenz eines $\mathbb{F}_q$-linearen Codes zu einem vorgeschriebenen Para-

metersatz, welcher in [31] als offen[5] geführt wird, ausgeschlossen werden. Die Beispiele belegen, dass es sich bei dem Kanonisierer um einen kompetitiven Algorithmus handelt. Über die beiliegende CD können die erzielten Verbesserungen eingesehen werden.

Das Kapitel 7 schließt die Arbeit mit der Beschreibung der entstandenen Software ab. Die entstandenen Programmpakete können über die beiliegende CD installiert werden. Der Quellcode dieser Programme ist unter den Bedingungen der GNU General Public License (Version 3) [30] freigegeben, d.h. er darf beliebig kopiert, verbreitet, modifiziert und genutzt werden.

[5] Die Minimaldistanz eines optimalen linearen Codes ist bei vorgegebener Länge und Dimension unbekannt.

2. Grundlagen

In dieser Arbeit werden die auftretenden algebraischen Strukturen, wie Gruppen, Ringe, Körper, Moduln usw., stets endlich sein. Wir werden die auftretenden Operationen, d.h. Gruppenoperationen auf Mengen bzw. die Moduloperationen, bevorzugt von links betrachten. Zumeist lassen sich die Definitionen und Sätze auf eine Operation von rechts übertragen. Gegebenenfalls werden wir auf diese auch ohne eine entsprechende Definition zurückgreifen.

Zur Unterscheidung von Links- bzw. Rechtsmoduln werden wir ${}_RM$ bzw. M_R schreiben. Im Fall von $M = R^k$ wollen wir überdies vereinbaren, dass ${}_RR^k$ den Linksmodul aller Zeilenvektoren und R_R^k den Rechtsmodul aller Spaltenvektoren bezeichne. Wir werden also, wie in der Codierungstheorie üblich, Vektoren als Zeilenvektoren auffassen.

2.1. Gruppen und Gruppenoperationen

Bis auf Weiteres sei G eine fest vorgegebene endliche Gruppe, die auf einer endlichen Menge X operiere. Wir bezeichnen X auch als *G-Menge* und werden diese Eigenschaft auch kurz mit ${}_GX$ kennzeichnen. Dieser Abschnitt soll vor allem zur Festlegung der Notationen dienen, für eine Einführung in die Theorie der Gruppenoperationen verweisen wir auf [43].

Die *Bahn* eines Elements $x \in X$ werden wir mit $Gx := \{gx \mid g \in G\}$ bezeichnen und die Menge aller Bahnen mit $G\backslash\backslash X := \{Gx \mid x \in X\}$. Ein minimales System von Repräsentanten aller Bahnen nennen wir *Transversale*. Diejenigen Elemente $x \in X$, welche unter allen Gruppenelementen wieder auf sich selbst abgebildet werden, nennen wir *Fixpunkte*. Die Menge aller Fixpunkte von X werden wir mit $\mathrm{Fix}_G(X) := \{x \in X \mid gx = x,\ \forall\, g \in G\}$ bezeichnen.

Zu einer Gruppe G sei $\mathcal{L}(G) := \{H \mid H \leq G\}$ die Menge aller Untergruppen von G. Die Gruppe G operiert durch Konjugation auf $\mathcal{L}(G)$. Ist umgekehrt $H \in \mathcal{L}(G)$ eine beliebige Untergruppe von G so können wir die Multiplikation mit Elementen aus H

- von links als eine Operation der Gruppe H von links auffassen. Eine Bahn Hg zu $g \in G$ nennen wir auch eine *Rechtsnebenklasse* von H. Die Bahnenmenge bezeichnen wir dann zur Auszeichnung dieser speziellen Situation mit $H\backslash G$.
- von rechts als eine Operation der Gruppe H von rechts auffassen. Die Bahnen nennen wir entsprechend *Linksnebenklassen* und wir bezeichnen die Bahnenmenge mit G/H.

Eine Transversale $T \subseteq G$ der (Links-)Rechtsnebenklassen von H nennen wir dann eine *(Links-)Rechtstransversale* von H in G. Ist H ein Normalteiler von G so trägt $G/H = H\backslash G$ eine von G induzierte Gruppenstruktur mit neutralem Element H.

Im Folgenden wollen wir außerdem mit

$$\mathcal{C}(G) := \{Hg \mid H \leq G,\ g \in G\} := \bigcup_{H \in \mathcal{L}(G)}^{\cdot} H\backslash G$$

die Menge aller Rechtsnebenklassen aller Untergruppen H von G bezeichnen. Zusätzlich definieren wir auf der Menge $\mathcal{C}(G)$ eine Gruppenoperation von $g_0 \in G$ über $g_0 \star Hg := Hgg_0^{-1}$.

Eine entscheidende Rolle bei der Untersuchung von Gruppenoperationen bilden Homomorphismen: Ist Y eine weitere G-Menge, so nennen wir eine Funktion $f : X \to Y$ einen *G-Homomorphismus*, falls $f(gx) = gf(x)$ für alle $x \in X$ und $g \in G$ gilt. Ist die Operation im Bildbereich trivial ($gy = y$ für alle $g \in G$ und $y \in Y$), so sprechen wir auch von einer *G-Invarianten* f.

Operiert auf Y eine Gruppe H und ist $\varphi : G \to H$ ein Gruppenhomomorphismus, welcher mit einer Abbildung $f : X \to Y$ im folgenden Sinne

$$f(gx) = \varphi(g)f(x), \forall g \in G, x \in X,$$

verträglich ist, so nennen wir das Paar (φ, f) einen *Homomorphismus von Gruppenoperationen*.

2.1.1 Bemerkung. Offensichtlich können wir über einen solchen Homomorphismus (φ, f) von Gruppenoperationen auch eine Operation von G auf dem Bild $f(X) \subseteq Y$ definieren, vermöge: $g \cdot f(x) := \varphi(g)f(x)$. Setzen wir die Operation von G auf $Y \setminus f(X)$ trivial fort, so haben wir eine Gruppenoperation von G auf Y definiert. Die Abbildung f ist also auch ein G-Homomorphismus. Homomorphismen von Gruppenoperationen haben den Vorteil, dass sich die Angabe der Operation hierdurch im Bildbereich zumeist auf natürliche Weise definieren lässt.

Nun zu der formalen Beschreibung unserer Problemstellung für allgemeine Gruppenoperationen.

2.1.2 Definition (Kanonisierung)**.** Eine G-invariante Funktion $\mathrm{CF}_G : X \to X$ nennen wir *Kanonisierung*, falls $\mathrm{CF}_G(x) \in Gx$ für alle $x \in X$ gilt. Das Element $\mathrm{CF}_G(x)$ bezeichnen wir als den *kanonischen Repräsentanten* von x bzw. der Bahn Gx.

2.1.3 Definition (Transporterelement)**.** Ist $\mathrm{CF}_G : X \to X$ eine Kanonisierung so nennen wir Gruppenelemente $g \in G$ mit $gx = \mathrm{CF}_G(x)$ *Transporterelemente* zu $x \in X$. Eine Abbildung $\mathrm{TR}_G : X \to G$, welche jedem $x \in X$ ein zugehöriges Transporterelement zuordnet heißt *Transporterabbildung*.

Transporterelemente sind bis auf Rechtsmultiplikation mit Elementen aus dem Stabilisator $\mathrm{Stab}_G(x) := \{g \in G \mid gx = x\}$ von x eindeutig bestimmt. Ist die operierende Gruppe aus dem Zusammenhang ersichtlich, so wollen wir Elemente des Stabilisators auch als *Automorphismen* von x bezeichnen. Insofern werden wir auch von der Automorphismengruppe $\mathrm{Aut}(x)$ des Objekts x sprechen. Genauso benutzen wir auch den Begriff *isomorph* für Elemente x, x' der gleichen Bahn. Da jede Gruppenoperation auch eine Äquivalenzrelation auf X definiert, werden wir die Elemente x, x' der gleichen Bahn auch als *äquivalent* bezeichnen.

2.1.4 Definition (Kanonisierer)**.** Ist $\mathrm{CF}_G : X \to X$ eine Kanonisierung mit einer Transporterabbildung $\mathrm{TR}_G : X \to G$, so wollen wir das Tripel $\mathrm{Can}_G := (\mathrm{CF}_G, \mathrm{TR}_G, \mathrm{Stab}_G)$ als *Kanonisierer* bezeichnen.

2.1.5 Bemerkung. Gegebenenfalls – etwa falls G auf mehreren Mengenoperiert – werden wir die Kanonisierer $\mathrm{Can}_G^X := (\mathrm{CF}_G^X, \mathrm{TR}_G^X, \mathrm{Stab}_G)$ über die zusätzliche Angabe der Menge X im Exponenten unterscheiden. Für die Zuordnung von $x \in X$ auf seinen Stabilisator $\mathrm{Stab}_G(x)$ besteht keine Wahlmöglichkeit. Wir verzichten daher auf die zusätzliche Angabe von X in dieser Beschreibung der Stabilisatorfunktion $\mathrm{Stab}_G : X \to \mathcal{L}(G)$.

2.1.6 Bemerkung. Durch Angabe des Transporterelements $\mathrm{TR}_G(x)$ ist der kanonische Repräsentant $\mathrm{CF}_G(x) = \mathrm{TR}_G(x)x$ von $x \in X$ bereits eindeutig bestimmt. Die redundante Information CF_G bei der Definition des Kanonisierers soll vor allem verdeutlichen, zu welcher Kanonisierungsfunktion die Transporterelemente bestimmt wurden.

Diese Definition eines Kanonisierers ist weniger aus mathematischer Sichtweise motiviert sondern gibt vielmehr unseren algorithmischen Standpunkt auf das Problem wieder. Wir wollen ein Computerprogramm entwerfen, welches einen Kanonisierer implementiert. Eingabe ist also ein Objekt $x \in X$ und wir erwarten die Rückgabe eines kanonischen Repräsentanten, eines zugehörigen Transporterelements und des Stabilisators. In diesem Sinne wollen wir auch die Berechnung des Stabilisators $\mathrm{Stab}_G(x)$ als durchgeführt ansehen, d.h. wir haben diese Untergruppe nicht nur formal sondern tatsächlich über ein berechnetes Erzeugendensystem E für weitere Untersuchungen zur Verfügung.

Über den Programmfluss steuern wir, dass tatsächlich ein Kanonisierer Can_G realisiert wird. Die mathematische Angabe der zugehörigen Kanonisierungsfunktion ist damit häufig sehr umfangreich und nur dem Programmablauf zu entnehmen.

Folgende Gruppen sind für uns von besonderem Interesse:

- Die symmetrische Gruppe $S_X := \{f : X \to X \mid f \text{ bijektiv}\}$ auf einer endlichen Menge X. Ist $X = [n] := \{0, \ldots, n-1\}$, so schreiben wir auch kurz S_n statt $S_{[n]}$.
- Die multiplikative Gruppe R^* aller Einheiten des Rings R.
- Die Gruppe aller invertierbaren $(k \times k)$-Matrizen, die allgemeine lineare Gruppe $\mathrm{GL}_k(R)$ über dem Ring R.

Über die Definition $(A, x) \mapsto xA^{-1}$ operiert die Gruppe $\mathrm{GL}_k(R)$ von links auf ${}_RR^k$. Für fest gewähltes $A \in \mathrm{GL}_k(R)$ ist hierdurch auch eindeutig eine linkslineare, invertierbare Abbildungen $f_A : {}_RR^k \to {}_RR^k, x \mapsto xA^{-1}$ gegeben und damit der natürliche Gruppenisomorphismus in die Gruppe der linkslinearen Abbildungen definiert.

Des Weiteren bezeichnen wir zu zwei Gruppen G, H und einem gegebenen Gruppenhomomorphismus $\theta : G \to \mathrm{Aut}(H)$ mit $H \rtimes_\theta G := \{(h; g) \mid h \in H, g \in G\}$ das *semidirekte Produkt* beider Gruppen bezüglich θ. Die Multiplikation ist dabei über die Vorschrift

$$(h; g)(h'; g') := (h\theta(g)(h'); gg')$$

definiert. Wir werden die Angabe des Gruppenhomomorphismus θ unterdrücken, sofern dieser aus dem Kontext eindeutig hervorgeht. Ist $\theta(g) = \mathrm{id}_H$ für alle $g \in G$, so ist das semidirekte Produkt gleich dem direkten Produkt beider Gruppen.

2.1.7 Beispiel. Es sei R ein Ring und der Gruppenhomomorphismus

$$\begin{aligned}\theta : \mathrm{Aut}(R) &\to \mathrm{Aut}(\mathrm{GL}_k(R)) \\ \alpha &\mapsto ((A_{i,j}) \mapsto (\alpha(A_{i,j})))\end{aligned}$$

definiert über die komponentenweise Anwendung des Automorphismus $\alpha \in \mathrm{Aut}(R)$ auf die Matrizen $A \in \mathrm{GL}_k(R)$. Das semidirekte Produkt $\Gamma\mathrm{L}_k(R) := \mathrm{GL}_k(R) \rtimes_\theta \mathrm{Aut}(R)$ nennen wir die *allgemeine semilineare Gruppe* vom Grad k.

2.1.8 Definition (semilineare Abbildungen)**.** Wir nennen eine Abbildung $f : {}_RR^n \to {}_RR^n$ *linkssemilinear*, falls es einen Ringautomorphismus $\alpha \in \mathrm{Aut}(R)$ gibt, so dass $f(u + v) = f(u) + f(v)$ und $f(ru) = \alpha(r)f(u)$ für alle $u, v \in R^k$ und $r \in R$ gilt.

Die Gruppe $\Gamma\mathrm{L}_k(R)$ operiert auf ${}_RR^k$ über die Definition $(A; \alpha)v := \alpha(v)A^{-1}$ und ist isomorph zur Gruppe der linkssemilinearen Abbildungen auf R^k. Im Folgenden werden wir daher die Gruppe der (semi-)linearen Abbildungen als $\mathrm{GL}_k(R)$ bzw. $\Gamma\mathrm{L}_k(R)$ ausdrücken.

Schließlich werden wir zu einer Permutation $\pi \in S_n$ und einem Ring R die Permutationsmatrix $P^{(\pi)} \in \{0_R, 1_R\}$ über

$$P^{(\pi)} := \begin{pmatrix} e_{\pi^{-1}(0)} \\ \vdots \\ e_{\pi^{-1}(n-1)} \end{pmatrix} = \begin{pmatrix} e^T_{\pi(0)} & \cdots & e^T_{\pi(n-1)} \end{pmatrix}$$

definieren. Es bezeichne hierbei e_i den i-ten Einheitsvektor. Wir können also über den Gruppenmonomorphismus $S_n \to \mathrm{GL}_n(R)$, $\pi \mapsto P^{(\pi)}$ die Gruppe S_n auch als Untergruppe von $\mathrm{GL}_n(R)$ bzw. $\Gamma\mathrm{L}_n(R)$ auffassen. Eine Permutation $\pi \in S_n$ operiert daher auf ${}_RR^n$ durch Rechtsmultiplikation mit $(P^{(\pi)})^{-1} = (P^{(\pi)})^T = P^{(\pi^{-1})}$, d.h.

$$\begin{aligned}\pi \cdot e_i := e_i P^{(\pi^{-1})} &= e_{\pi(i)} = \big((e_i)_{\pi^{-1}(0)}, \ldots, (e_i)_{\pi^{-1}(n-1)}\big) \\ \Longrightarrow \pi \cdot (v_0, \ldots, v_{n-1}) &= (v_0, \ldots, v_{n-1})P^{(\pi^{-1})} \\ &= (v_{\pi^{-1}(0)}, \ldots, v_{\pi^{-1}(n-1)}) \text{ für alle } v \in {}_RR^n.\end{aligned}$$

Ist R ein Ring und $\varphi \in R^n$, so bezeichne $\operatorname{diag}(\varphi) := D \in R^{n\times n}$ die Diagonalmatrix D mit Einträgen $D_{i,i} = \varphi_i$. Das semidirekte Produkt $(R^*)^n \rtimes_\Theta S_n$ mit $\Theta(\pi)(\varphi) := \varphi P^{(\pi^{-1})}$ nennen wir die *monomiale Gruppe* vom Grad n über R. Diese Produktbildung kann auch als das Kranzprodukt $R^* \wr_n S_n$ von R^* mit S_n gesehen werden.

Auch die monomiale Gruppe lässt sich über den Gruppenmonomorphismus

$$(R^*)^n \rtimes_\Theta S_n \to \mathrm{GL}_n(R),\ (\varphi;\pi) \mapsto \operatorname{diag}(\varphi)P^{(\pi)}$$

in die Gruppe $\mathrm{GL}_n(R)$ einbetten. Die Bilder nennen wir daher auch *monomiale Matrizen*. Diese Gruppe operiert somit auf ${}_RR^n$ durch:

$$(\varphi;\pi)v := vP^{(\pi^{-1})}\operatorname{diag}(\varphi)^{-1} = \left(v_{\pi^{-1}(0)}\varphi_0^{-1}, \ldots, v_{\pi^{-1}(n-1)}\varphi_{n-1}^{-1}\right).$$

Nimmt man zusätzlich noch die Ringautomorphismen hinzu, so erhält man die sogenannte *semimonomiale Gruppe* $((R^*)^n \rtimes_\Theta S_n) \rtimes_\theta \mathrm{Aut}(R)$ und wir können diese analog in $\Gamma\mathrm{L}_k(R)$ einbetten. Die von $(\varphi;\pi,\alpha) \in ((R^*)^n \rtimes_\Theta S_n) \rtimes_\theta \mathrm{Aut}(R)$ auf ${}_RR^n$ induzierte Abbildung

$$\begin{aligned} v \mapsto (\varphi;\pi,\alpha)v := (\varphi;\pi)\alpha(v) &= \alpha(v)P^{(\pi^{-1})}\operatorname{diag}(\varphi)^{-1} \\ &= \left(\alpha(v_{\pi^{-1}(0)})\varphi_0^{-1}, \ldots, \alpha(v_{\pi^{-1}(n-1)})\varphi_{n-1}^{-1}\right) \end{aligned}$$

nennen wir auch eine *(semi-)monomiale Transformation* von R^n.

Da die Operation der Automorphismengruppe auf Permutationsmatrizen trivial ist, können wir dieses semidirekte Produkt auch wie folgt beschreiben:

$$((R^*)^n \rtimes_\Theta S_n) \rtimes_\theta \mathrm{Aut}(R) = (R^*)^n \rtimes_\vartheta (S_n \times \mathrm{Aut}(R)) \tag{2.1}$$

mit $\vartheta((\pi,\alpha)) := \Theta(\pi) \circ \theta(\alpha) = \theta(\alpha) \circ \Theta(\pi)$ für alle $(\pi,\alpha) \in S_n \times \mathrm{Aut}(R)$.

Wir werden später sehen, dass die Gruppe (2.1) in der Codierungstheorie über endlichen Kettenringen[1] R eine zentrale Rolle spielt. Sie definiert den allgemeinsten Äquivalenzbegriff für R-lineare Codes.

Analog ist die Operation von $\Gamma\mathrm{L}_n(R)$ auf der Menge der Spaltenvektoren R_R^n gegeben durch $(A;\alpha)v^T = A\alpha(v^T)$. Sie ist isomorph zu den rechtssemilinearen Abbildungen auf R_R^n. Entsprechend leiten sich auch die Operationen der eingeführten Untergruppen ab.

2.2. Graphen

Wie wir bereits in der Einleitung kurz bemerkten, lassen sich viele Isomorphieprobleme diskreter Strukturen, wie etwa linearer Codes, auf das Graphenisomorphieproblem zurückführen. Wir wollen daher auf diese kombinatorische Struktur kurz eingehen. Umgekehrt lässt sich das Graphenisomorphieproblem aber auch mit einem Äquivalenztest

[1] Definition folgt.

für lineare Codes lösen. Dies ist vor allem aus Gründen der Komplexitätsabschätzung, die wir in Abschnitt 2.4 durchführen werden, von großem Interesse.

Des Weiteren ist dieser Exkurs auch durch die Tatsache motiviert, dass der in [55] beschriebene Kanonisierer für Graphen fundamentale Ideen, siehe Kapitel 3, für die Entwicklung unserer codierungstheoretischen Kanonisierer beinhaltet. Schließlich erfolgt die Formulierung des Lösungsalgorithmus selbst über die Definition eines Suchbaums, d.h. über spezielle Graphen.

Wir wollen zu einer beliebigen Menge V mit $\binom{V}{k}$ die Menge ihrer k-Teilmengen bezeichnen. Ein *(ungerichteter) Graph* $\Gamma = (V, E)$ ist ein Tupel mit einer Menge V von Knoten und einer Menge $E \subseteq \binom{V}{2}$ von Kanten. Wir schreiben dann auch $[2]^{\binom{V}{2}}$ für die Menge aller Graphen mit Knotenmenge V, wobei wir den Vektor $e \in [2]^{\binom{V}{2}}$ als die Menge $E := \{\{v, w\} \in \binom{V}{2} \mid e_{\{v,w\}} = 1\}$ interpretieren.

Einen *gerichteten Graphen* erhält man, wenn man die Knotenpaare geordnet betrachtet, d.h. $E \subseteq V \times V$ wählt. Für unsere Zwecke können wir immer von einer endlichen Menge V ausgehen und zumeist sind die untersuchten Graphen ungerichtet.

Wir nennen zwei Graphen $(V, E), (V', E')$ *isomorph*, falls es eine Bijektion $f : V \to V'$ gibt mit $\{v, w\} \in E \iff \{f(v), f(w)\} \in E'$. Isomorphe Graphen gehen also durch Umnummerierung der Knotenbeschriftungen auseinander hervor. Zur Untersuchung der Isomorphie von Graphen können wir stets annehmen, dass die Knotenmengen V, V' gleich sind und dass ohne Beschränkung der Allgemeinheit $V = [n] := \{0, \ldots, n-1\}$ für ein $n \in \mathbb{N}$ gilt. Andernfalls bilden wir V bzw. V' über eine beliebige Bijektion nach $[n]$ ab. Diese Beobachtung erlaubt es uns nun, die Graphenisomorphie über die Gruppenoperation der symmetrischen Gruppe S_n auf der Potenzmenge $[2]^{\binom{[n]}{2}}$ aller Zweierteilmengen von $[n]$ zu untersuchen.

Zwei Knoten $u, v \in V$ eines Graphen (V, E) heißen *benachbart* (adjazent), falls $\{u, v\} \in E$ gilt. Eine *Adjazenzmatrix* $A \in \{0, 1\}^{n \times n}$ eines Graphen $([n], E)$ beschreibt die Nachbarschaftsbeziehung durch die Definition $A_{i,j} = 1 :\iff \{i, j\} \in E$. Über eine *Inzidenzmatrix* $I \in \{0, 1\}^{m \times n}$ eines Graphen $([n], E)$ wird die Knoten-Kanten-Inklusion beschrieben; man erhält sie über eine Anordnung der Kantenmenge $(e_0, \ldots, e_{m-1})$ und Setzung $I_{i,j} = 1 :\iff j \in e_i$. Inzidenzmatrizen sind also nur bis auf Permutation der Zeilen eindeutig bestimmt.

Der nachfolgende Satz beschreibt die Isomorphie von Graphen über verschiedene Gruppenoperationen, je nachdem ob man zur Darstellung des Graphen die Kantenmenge, eine Adjazenzmatrix oder eine Inzidenzmatrix wählt:

2.2.1 Fakt. *Es seien $G_i = ([n], E_i)$, $i = 0, 1$ Graphen mit m Kanten und gegebenen Inzidenzmatrizen I_i und Adjazenzmatrizen A_i. Dann sind äquivalent:*

- *G_0 und G_1 sind isomorph.*
- *$E_1 \in S_n E_0$ (mit $\pi\{\ldots, \{u, v\}, \ldots\} := \{\ldots, \{\pi(u), \pi(v)\}, \ldots\}$)*

- $A_1 \in S_n A_0$ *(mit* $\pi A := P^{(\pi)} A P^{(\pi^{-1})}$*)*
- $I_1 \in (S_m \times S_n) I_0$ *(mit* $(\sigma, \pi) I := P^{(\sigma)} I P^{(\pi^{-1})}$*)*

Abschließend wollen wir noch *Wurzelbäume* einführen, die wir zur Definition der Kanonisierer benötigen werden. Im Graphen (V, E) nennen wir eine Folge paarweise verschiedener Knoten $v_0, \ldots, v_k \in V$ mit $\{v_i, v_{i+1}\} \in E$ für alle $i \in [k]$ einen *Pfad* von v_0 nach v_k. Die Knoten $v_0, v_k \in V$ nennen wir dann auch *verbunden*. Der Graph (V, E) ist *zusammenhängend*, falls alle Knoten paarweise verbunden sind.

Einen zusammenhängenden Graphen (V, E), bei welchem genau ein Pfad zwischen jedem beliebigen Knotenpaar existiert, nennen wir *Baum*. Ist überdies $r \in V$ ein ausgezeichneter Knoten, so nennen wir $((V, E), r)$ einen *Wurzelbaum* mit *Wurzel* r. Jedem Knoten v eines Wurzelbaums können wir über die Pfadlänge des Pfads von der Wurzel r nach v eine eindeutige natürliche Zahl $d(v)$ zuordnen. Wir nennen sie die *Tiefe* des Knotens v. Ist $\{v, w\} \in E$ eine Kante eines Wurzelbaums und $d(v) = d(w)+1$, so nennen wir w ein *Kind(-knoten)* von v und v den *Vater(-knoten)* zu w. Dementsprechend nennen wir alle Knoten w, die auf dem Pfad von der Wurzel zu $v \in V$ liegen, auch *Vorfahren* von v und umgekehrt v einen *Nachfahren* von w. Knoten ohne Nachfahren heißen *Blätter*. Der Teilgraph bestehend aus allen Nachfahren eines Knotens $v \in V$ bildet wiederum selbst einen Wurzelbaum mit Wurzel v.

Unter einer *Breitensuche* (*breadth-first-search*) auf einem Wurzelbaum $((V, E), r)$ verstehen wir eine Besuchsreihenfolge $(v_0, \ldots, v_{n-1})$ aller Knoten, so dass die entsprechende Folge der Tiefen monoton wächst. Im Gegensatz dazu dringt die sogenannte *Tiefensuche* (*depth-first-search*) zunächst bis zu einem beliebigen Blatt im Baum vor und kehrt danach rekursiv zu den Vorfahren zurück um dort alle weiteren unbesuchten Kinder und deren Nachfahren ebenfalls in Tiefensuche zu durchlaufen.

2.3. Endliche Kettenringe

Es sei R stets ein assoziativer Ring mit Eins. Die hier angegebenen Resultate über Kettenringe wurden aus [11] entnommen.

2.3.1 Definition (Kettenring)**.** Wir nennen R einen Linkskettenring, falls die Menge der Linksideale bzgl. Inklusion totalgeordnet ist. Der Idealverband bildet also eine Kette $\{0\} = I_0 \lhd I_1 \lhd \ldots \lhd I_m = R$. Rechtskettenringe seien analog definiert. Einen Ring, der sowohl Links- als auch Rechtskettenring ist, nennen wir *Kettenring*.

2.3.2 Fakt. *Ist R endlich, so ist R genau dann ein Linkskettenring, wenn R ein Rechtskettenring ist.*

Da wir im Folgenden nur noch endliche Kettenringe betrachten werden, können wir also die Unterscheidung zwischen Links- und Rechtskettenringen vernachlässigen. Das *Jacobson-Radikal* $\mathrm{Rad}(R)$ eines Rings R ist definiert als der Schnitt aller maximalen

Ideale. Für einen Kettenring ist also das Jacobson-Radikal gleich dem eindeutigen maximalen Ideal in R.

2.3.3 Fakt. *Ist R ein Kettenring und θ ein beliebiges Element aus $\mathrm{Rad}(R) \setminus \mathrm{Rad}(R)^2$, so lässt sich jedes echte Ideal $I \lhd R$ in der Form $I = \mathrm{Rad}(R)^i = R\theta^i = \theta^i R$ für eine positive Zahl i darstellen.*

Zur Vereinheitlichung definieren wir $R =: \mathrm{Rad}(R)^0$ und $a^0 := 1_R$ für alle $a \in R$. Der Nilpotenzindex von $\mathrm{Rad}(R)$, d.h. die kleinste natürliche Zahl $m \in \mathbb{N}$ mit $\mathrm{Rad}(R)^m = \{0_R\}$ bzw. $\theta^m = 0_R$, wird auch als *Kettenlänge* des Kettenrings R bezeichnet. Wir verwenden den Bezeichner m_R oder kurz m, falls der Ring R aus dem Kontext hervorgeht.

2.3.4 Fakt. *Ist R ein endlicher Kettenring, dann ist der Faktorring $R/\mathrm{Rad}(R)$ isomorph zu einem endlichen Körper $\mathbb{F}_q$.*

Im Folgenden werden wir mit $\bar{\ } : R \to \mathbb{F}_q$ die Abbildung in den Restklassenkörper bezeichnen. Des Weiteren seien die natürlichen Zahlen q_R, p_R, r_R die eindeutig bestimmten Parameter des Restklassenkörper $R/\mathrm{Rad}(R) \simeq \mathbb{F}_{q_R}$ mit $q_R = p_R^{r_R}$. Falls keine Verwechslungsgefahr besteht, werden wir den Ring R auch unterdrücken.

2.3.1. Moduln und lineare Codes

Wir werden in diesem Abschnitt lineare Codes über Kettenringen analog zu [37] einführen. Die hier aufgeführten Aussagen sind aus dieser Quelle übernommen. Für eine Einführung in die klassische Codierungstheorie verweisen wir den Leser auf [3] und [39]. Insbesondere die erstgenannte Quelle legt ihren Schwerpunkt auf die Operation der semilinearen Isometriegruppe auf der Menge aller linearen Codes.

Im Folgenden sei also nun R immer ein endlicher Kettenring mit Kettenlänge m und fest gewähltem Erzeuger θ von $\mathrm{Rad}(R)$. Weiter sei ${}_RM$ ein beliebiger R-Linksmodul.

2.3.5 Definition (Höhe, Periode)**.** Zu $x \in {}_RM$ sei $\mathrm{per}(x) \leq m$ die kleinste ganze Zahl mit $\theta^{\mathrm{per}(x)}x = 0_M$. Wir nennen sie die *Periode* des Elements x. Die Zahl $\mathrm{ht}(x) := m - \mathrm{per}(x)$ nennen wir die *Höhe* von x.

Da R selbst einen R-Linksmodul bildet, haben wir diese Begriffe auch für $a \in R$ zur Verfügung. Die Höhe $\mathrm{ht}(a)$ gibt also genau das minimale Ideal $\mathrm{Rad}(R)^{\mathrm{ht}(a)}$ an, welches a noch enthält beziehungsweise von a erzeugt wird. Es gilt also insbesondere $R\theta^{\mathrm{ht}(a)} = Ra$ und diese Zahl ist somit auch unabhängig davon, ob wir R als Links- oder Rechtsmodul betrachten. Umgekehrt gibt uns die Periode zu $x \in {}_RM$ den Annihilator $\mathrm{Rad}(R)^{\mathrm{per}(x)}$ des zyklischen Untermoduls Rx.

2.3.6 Definition (unabhängige Menge, Basis)**.** Eine Menge $\{m_0, \ldots, m_{k-1}\} \subseteq M$ heißt *unabhängig* (bzw. *linear unabhängig*), falls aus $\sum_{i=0}^{k-1} a_i m_i = 0_M$ mit $a_i \in R$ bereits $a_i m_i = 0_M$ (bzw. $a_i = 0$) für alle $i \in [k]$ folgt. Eine unabhängige Menge $B \subseteq M$ von Erzeugern von M mit $0_M \notin B$ nennen wir eine *Basis* des Moduls.

2.3.7 Fakt. *Jeder endliche R-Linksmodul*

$$M \simeq R\theta^{m-\lambda_0} \oplus \ldots \oplus R\theta^{m-\lambda_{k-1}}$$

ist eine direkte Summe zyklischer R-Moduln. Dabei ist die monoton fallende Folge $\mathrm{shp}(M) := \lambda := (\lambda_0, \ldots, \lambda_{k-1})$ *eindeutig bestimmt. Wir nennen sie den* Umriss *von M und ihre Länge den* Rang *von M, kurz:* $\mathrm{rg}(M) := k$.

Einen Modul M vom Umriss $\mathrm{shp}(M) = (m, \ldots, m)$ nennen wir *frei*. Er ist demzufolge isomorph zu ${}_RR^{\mathrm{rg}(M)}$. Mit Hilfe des vorangegangenen Satzes, der eine Verallgemeinerung des Hauptsatzes über endliche abelsche Gruppen darstellt, lässt sich leicht zeigen, dass sich der Dimensionsbegriff für Vektorräume auf Moduln über Kettenringen sinnvoll übertragen lässt.

2.3.8 Fakt. *Jede Basis B von* ${}_RM$ *hat die gleiche Mächtigkeit* $\mathrm{rg}(M)$. *Der Umriss entspricht der monoton fallenden Folge der Perioden der Elemente von B.*

Den modularen Verband aller Linksuntermoduln von ${}_RR^k$ werden wir als *projektive Links-Hjelmslev-Geometrie* $\mathrm{PHG}({}_RR^k)$ der Dimension $k-1$ bezeichnen. Die freien Untermoduln vom Rang 1, 2 bzw. $k-1$ werden auch *Punkte*, *Ebenen* bzw. *Hyperebenen* genannt. Entsprechend definiert man auch die *projektive Rechts-Hjelmslev-Geometrie* über die Untermoduln von R_R^k. Die projektiven Hjelmslev-Geometrien verallgemeinern den Begriff der desarguesschen projektiven Geometrien $\mathrm{PG}(\mathbb{F}_q^k)$ für endliche Körper und wir können den Hauptsatz in gleicher Form für projektiven Hjelmslev-Geometrie formulieren:

2.3.9 Fakt (Hauptsatz der projektiven Hjelmslev-Geometrie, [48])**.** *Für* $k \geq 3$ *werden die Verbandsautomorphismen (Kollineationen) von* $\mathrm{PHG}({}_RR^k)$ *genau von den linkssemilinearen Bijektionen* $f : {}_RR^k \to {}_RR^k$ *bzw. von Gruppenelementen* $(A; \alpha) \in \Gamma\mathrm{L}_k(R)$ *induziert.*

2.3.10 Folgerung. *Es sei* $k \geq 3$. *Eine bijektive Abbildung* $f : {}_RR^k \to {}_RR^k$ *bildet also genau dann Untermoduln auf Untermoduln ab, wenn f linkssemilinear ist. Analoge Aussagen gelten für projektive Rechts-Hjelmslev-Geometrien* $\mathrm{PHG}(R_R^k)$ *und rechtssemilineare Bijektionen.*

Für einen Linksuntermodul M von ${}_RR^k$ definieren wir den dualen Modul

$$M^\perp := \{v \in R_R^k \mid \forall\, u \in M : u \cdot v := \sum_{i=0}^{k-1} u_i v_i = 0\}.$$

Dieser ist dann offensichtlich ein Rechtsuntermodul von R_R^k. Analog definieren wir zu dem Rechtsmodul $N \leq R_R^k$ den dualen Modul

$${}^\perp N := \{u \in {}_RR^k \mid \forall\, v \in N : u \cdot v := \sum_{i=0}^{k-1} u_i v_i = 0\}.$$

2.3.11 Fakt ([37], Theorem 3.1)**.**

- *Für* $M \leq {}_R R^n$ *und* $N \leq R^n_R$ *gilt:* ${}^{\perp}\big(M^{\perp}\big) = M$ *und* $\big({}^{\perp}N\big)^{\perp} = N$.
- *Das Dualisieren definiert also zueinander inverse Verbandsantiautomorphismen* $\mathrm{PHG}(R^n_R) \to \mathrm{PHG}({}_R R^n)$ *bzw.* $\mathrm{PHG}({}_R R^n) \to \mathrm{PHG}(R^n_R)$.
- *Ist* $\mathrm{shp}(M) = (\lambda_0, \ldots, \lambda_{k-1})$*, so ist* $\mathrm{rg}(M^{\perp}) = n - k_0$ *mit* $k_0 := |\{i \in [k] \mid \lambda_i = m\}|$ *und*

$$\mathrm{shp}(M^{\perp}) = (\underbrace{m, \ldots, m}_{(n-k)\text{-mal}}, m - \lambda_{k-1}, \ldots, m - \lambda_{k_0}) \in [m+1]^{n-k_0}.$$

Zu einem Ring $(R, +, \cdot)$ sei die entgegengesetzte Ringmultiplikation $* : R \to R$ definiert durch $a * b := ba$. Man rechnet leicht nach, dass der *entgegengesetzte Ring* $R^{\mathrm{op}} := (R, +, *)$ wieder einen Ring definiert. Ist R ein Kettenring, so ist auch der entgegengesetzte Ring ein Kettenring.

Der Übergang zum entgegengesetzten Ring ist insbesondere bei der Produktbildung transponierter Matrizen zu beachten:

$$(A \cdot B)^T = B^T * A^T, \ \forall\ A \in R^{k \times n},\ B \in R^{n \times m}$$

2.3.12 Bemerkung. Wir werden aus diesem Grund möglichst auf die Transposition verzichten und machen von unserer Unterscheidung der Zeilenvektoren ${}_R R^k$ und Spaltenvektoren R^k_R Gebrauch.

Für die projektiven Hjelmslev-Geometrien über diesen Ringen gilt dann:

2.3.13 Fakt. *Es sei* R *ein Ring und* $S := R^{\mathrm{op}}$*. Dann induziert die Transpositionsabbildung* $(.)^T : R^k_R \to {}_S S^k$ *einen Verbandsisomorphismus* $\mathrm{PHG}(R^k_R) \to \mathrm{PHG}({}_S S^k)$.

Lineare Codes

2.3.14 Definition (linearer Code)**.** Ein *linearer Code* C der Länge n über einem Kettenring R ist ein R-Linksuntermodul von R^n.

Lineare Codes über endlichen Körpern treten hier als Spezialfall der Kettenringe mit Kettenlänge 1 auf.

2.3.15 Definition (Generatormatrix)**.** Wir sagen $\Gamma \in R^{k \times n}$ ist eine *Generatormatrix* eines linearen Codes $C \leq R^n$, falls die Zeilen eine Basis von C bilden und die Folge $\lambda = (\lambda_0, \ldots, \lambda_{k-1})$ der Perioden der Zeilen monoton fällt. Wir sagen auch, dass Γ eine Generatormatrix vom Umriss λ ist. Die Menge aller Generatormatrizen zum Umriss λ wollen wir mit $R^{k \times n, \lambda}$ bezeichnen.

2.3.16 Notation. Ist $\Gamma \in R^{k\times n}$ eine beliebige Matrix, so bezeichnen wir für $i \in [k]$ bzw. $j \in [n]$ mit $\Gamma_{i,*}$ bzw. $\Gamma_{*,j}$ die i-te Zeile bzw. j-te Spalte von Γ.

Ist $I = (i_0, \ldots, i_{j-1}) \in [k]^j$ ein Wort der Länge $j \in \mathbb{N}$, so bezeichnet $\Gamma_{I,*} \in R^{j\times n}$ diejenige Matrix, deren ℓ-te Zeile für $\ell \in [j]$ gleich $\Gamma_{i_\ell,*}$ ist. Ist schließlich $I = \{i_0, \ldots, i_{j-1}\} \subset [k]$ mit $i_\ell < i_{\ell+1}$ für alle $\ell \in [j-1]$, so setzen wir $\Gamma_{I,*} = \Gamma_{(i_0,\ldots,i_{j-1}),*}$. Gleiches gilt für die Spalten.

Ist Γ eine Generatormatrix vom Umriss λ, so liegen also die Einträge der i-ten Zeile $\Gamma_{i,*}$ von Γ allesamt in dem Ideal $\mathrm{Rad}(R)^{m-\lambda_i}$ und es gibt in dieser Zeile mindestens einen Eintrag $\Gamma_{i,j}$ der Höhe $m - \lambda_i$.

2.3.17 Bemerkung. Wir werden die Begriffe *Rang* und *Umriss* auch für beliebige Matrizen $\Gamma \in R^{k\times n}$ benutzen. In diesem Fall beziehen wir uns auf den von den Zeilen von Γ erzeugten R-Linksmodul.

Ist der Code C ein freier Modul und $\Gamma \in R^{k\times n}$ eine Generatormatrix, so erhalten wir die Menge aller Generatormatrizen von C durch Linksmultiplikation mit invertierbaren Matrizen, d.h. als die Bahn $\mathrm{GL}_k(R)\Gamma$. Dies ist offensichtlich nicht mehr wahr, falls C nicht frei ist. Eine Permutation $\pi \in S_k$ der Zeilen ist zum Beispiel nur dann erlaubt, falls π die Perioden der Zeilen respektiert:

2.3.18 Beispiel. Die Zeilen der folgenden Matrizen

$$\Gamma_0 = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \qquad \Gamma_1 = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} \qquad \Gamma_2 = \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix}$$

über $\mathbb{Z}_4$ erzeugen den gleichen Code. Jedoch ist nur Γ_0 eine Generatormatrix, da die Zeilen von Γ_1 keine Basis bilden und die Zeilen von Γ_2 nicht absteigend nach der Periode geordnet auftreten.

2.3.19 Notation. Sind $m, n \in \mathbb{N}$ beliebig, so soll $\mathbf{1_m}$ den Vektor bezeichnen, welcher konstant gleich 1_R auf allen Einträgen ist. Genauso benutzen wir $\mathbf{1_{m\times n}}$ für eine gleichartige $(m \times n)$-Matrix und $\mathbf{0_n}$ bzw. $\mathbf{0_{m\times n}}$ im analogen Sinne für den Nullvektor bzw. eine Nullmatrix.

Der zu C duale Modul $C^\perp$ ist ein Rechtsuntermodul von R^n_R. Um zu vermeiden, dass links- und rechtslineare Codes unterschieden werden müssen, werden wir den linearen Code $C^\perp$ gegebenenfalls auch als Linksuntermodul des entgegengesetzten Rings $(R^{\mathrm{op}})^n$ auffassen. Eine Kontrollmatrix Δ von C ist dann – wie gewöhnlich – eine Generatormatrix des dualen Codes und es gilt $\Gamma\Delta^T = \mathbf{0_{k\times(n-k_0)}}$.

2.3.2. Distanzen und Isometrien

In diesem Abschnitt werden wir nun noch beweisen, dass die semimonomiale Gruppe – wie bei den endlichen Körpern auch – einen geeigneten und den allgemeinst möglichen Äquivalenzbegriff für lineare Codes definiert.

2.3.20 Definition (Isometrie)**.** Sind $(M, d), (M', d')$ metrische Räume, so nennen wir eine Abbildung $\iota : M \to M'$ eine *Isometrie*, falls $d'(\iota(x), \iota(y)) = d(x, y)$ für alle $x, y \in M$ gilt.

2.3.21 Beispiel. Auf R^n definiert die Funktion

$$d_{\mathrm{H}} : R^n \times R^n \to \mathbb{N},\ (u, v) \mapsto \{i \in [n] \mid u_i \neq v_i\}$$

die sogenannte *Hamming-Metrik*. Zu einem Vektor $v \in R^n$ nennen wir

$$\mathrm{w}_{\mathrm{H}}(v) := d_{\mathrm{H}}(v, 0) = \Big|\underbrace{\{i \in [n] \mid v_i \neq 0\}}_{=:\mathrm{supp}(v)}\Big|$$

das Hamming-Gewicht von v. In der klassischen Codierungstheorie über endlichen Körpern bestimmt der paarweise Abstand $d_{\mathrm{H}}(c, c')$ verschiedener Codeworte $c, c' \in C$ die Fehlerkorrektureigenschaften des Codes $C \leq \mathbb{F}_q^n$.

Da eine beliebige Isometrie ι des metrischen Raums (R^n, d) aber durchaus einen linearen Code $C \leq R^n$ auch auf eine nichtlineare Teilmenge (einen sogenannten *Blockcode*) $\iota(C) \subseteq R^n$ abbilden kann, schränkt man sich bei der Wahl der Gruppenoperation und somit für den Äquivalenzbegriff auf diejenige Untergruppe der Isometriegruppe ein, die Untermoduln auf Untermoduln abbilden. Nach dem Hauptsatz der projektiven Hjelmslev-Geometrie bilden diese für $n \geq 3$ gerade die Gruppe der semilinearen Isometrien.

Für beliebige Kettenringe macht es Sinn, neben der Hamming-Metrik auch andere Metriken zu betrachten. Dies liegt an der Tatsache, dass man die Teilräume nicht selbst, sondern isometrische Bilder zum Einsatz bringen möchte. Wir definieren daher zunächst allgemeiner:

2.3.22 Definition (Gewicht)**.** Eine Funktion $w : R^n \to \mathbb{R}_0^+ := \{x \in \mathbb{R} \mid x \geq 0\}$ wollen wir *Gewicht* nennen, falls die folgenden Bedingungen erfüllt sind:

- $w(a) = 0 \iff a = \mathbf{0_n}$
- $w(a) = w(-a)$ für alle $a \in R^n$
- $w(a + b) \leq w(a) + w(b)$

2.3.23 Fakt. *Ist w ein Gewicht für R und $n \in \mathbb{N}$, so definiert die additive Fortsetzung*

$$w : R^n \to \mathbb{R}_0^+,\ x \mapsto w(x) := \sum_{i=0}^{n-1} w(x_i)$$

auf R^n eine Gewichtsfunktion und $d_w : R^n \times R^n \to \mathbb{R}_0^+, (x, y) \mapsto w(x - y)$ eine Metrik.

2.3.24 Beispiel.

- Setzt man $w_{\mathrm{H}}(a) = 1$ für alle $a \in R \setminus \{0\}$, so erhält man die oben eingeführte Hamming-Metrik.
- Definiert man $w_{\mathrm{hom}}(a) := \begin{cases} 0, & \text{falls } a = 0 \\ q_R, & \text{falls } a \in \mathrm{Rad}(R)^{m-1} \setminus \{0_R\}, \\ q_R - 1, & \text{sonst} \end{cases}$
  so erhält man das sogenannte *homogene*[2] *Gewicht*.
- Auf $\mathbb{Z}_4$ nennt man w_{hom} auch das *Lee-Gewicht*. Mit Hilfe der sogenannten *Gray-Abbildung*

$$\begin{aligned} \iota : (\mathbb{Z}_4, d_{w_{\mathrm{hom}}}) &\to (\mathbb{F}_2^2, d_{\mathrm{H}}) \\ 0 \mapsto (0,0), 1 \mapsto (1,0), 2 \mapsto (1,1), 3 &\mapsto (0,1), \end{aligned}$$

  ließ sich das oben erwähnte, viel beachtete Resultat [36] über die $\mathbb{Z}_4$-Linearität der Kerdock-, Preparata- und Goethals-Codes beweisen.
- Ist R ein beliebiger endlicher Kettenring der Kettenlänge m mit $R/\operatorname{Rad}(R) \simeq \mathbb{F}_q$, so bezeichne $q^{m-2} w_{\mathrm{hom}}$ dasjenige Gewicht, welches aus dem homogenen Gewicht über die Multiplikation mit der Konstanten q^{m-2} hervorgeht. Durch die in [33] angegebene Verallgemeinerung der Gray-Abbildung $(R, d_{q^{m-2} w_{\mathrm{hom}}}) \to (\mathbb{F}_q^{q^{m-1}}, d_{\mathrm{H}})$ ist die Untersuchung R-linearer Codes über beliebigen Kettenringen R unter Benutzung des homogenen Gewichts motiviert.

2.3.25 Definition (BTL- und BTKL-Codes)**.** Über die Gray-Abbildung können wir einen linearen Code C der Länge n über dem Kettenring R mit den linearen Codes der Länge nq^{m-1} und der gleichen Mächtigkeit $|C|$ über dem Alphabet $\mathbb{F}_q$ vergleichen. Ist die Hamming-Minimaldistanz des Gray-Bildes besser als diejenige, welche alle *bekannten* linearen Codes über $\mathbb{F}_q$ mit gleichen Parametern erreichen, so werden wir C als *BTKL-Code* bezeichnen. Ist die Minimaldistanz sogar beweisbar stets besser, so nennt man solche Codes auch *BTL-Codes*. Die Abkürzungen stehen hierbei für „**b**etter **t**han (**k**nown) **l**inear“.

2.3.26 Beispiel. Die Preparata-Codes sind BTL-Codes, siehe [8]. Nach gegenwärtigem Wissensstand sind die Kerdock-Codes BTKL-Codes.

Zu einem Gewicht $w : R \to \mathbb{R}_0^+$ sei $U_w := \{a \in R^* \mid \forall\, b \in R : w(ba^{-1}) = w(b)\}$ die Symmetriegruppe des Gewichts w. Wir sagen das Gewicht w hat die *MacWilliams-Eigenschaft*, falls sich für jeden linearen Code $C \leq {}_R R^n$, $n \in \mathbb{N}$ jede beliebige lineare Isometrie $f : (C, d_w) \to (R^n, d_w)$ zu einer U_w-monomialen Transformation von R^n fortsetzen lässt, d.h. es existiert $(\varphi; \pi) \in (U_w)^n \rtimes S_n$, so dass $f(c) = cP^{(\pi^{-1})} \operatorname{diag}(\varphi^{-1})\ \forall c \in C$ gilt.

[2] Das Durchschnittsgewicht $\frac{\sum_{a \in I} w_{\mathrm{hom}}(a)}{|I|}$ aller Ideale $\{0\} \neq I \trianglelefteq R$ ist konstant.

2.3.27 Fakt (J. MacWilliams [52]). *Das Hamming-Gewicht besitzt die MacWilliams-Eigenschaft für jeden endlichen Körper $\mathbb{F}_q$.*

Die Menge aller U_w-monomialen Transformationen von R^n bildet eine Untergruppe der monomialen Gruppe vom Grad n. Wir wollen diese Untergruppe auch die *U_w-monomiale Gruppe* vom Grad n nennen. Hat ein Gewicht $w : R \to \mathbb{R}_0^+$ die MacWilliams-Eigenschaft, so folgt sofort mit $C = R^n$, dass dann die Menge aller linearen Isometrien von R^n gleich der U_w-monomialen Gruppe vom Grad n ist.

2.3.28 Fakt ([72, Theorem 9.4]). *Alle Gewichte $w : R \to \mathbb{R}_0^+$ mit $U_w = R^*$ haben die MacWilliams-Eigenschaft.*

2.3.29 Bemerkung. J. Wood [72] definiert einen wesentlich freieren Gewichtsbegriff als wir ihn hier zulassen. Er fordert lediglich eine Funktion $w : R \to \mathbb{Q}$ mit $w(0_R) = 0$. Das obige Theorem benötigt dann als weitere Voraussetzung $w(\theta^{m-1}) \neq 0$.

Hat das Gewicht w eine kleinere Symmetriegruppe, so können die zu erfüllenden Bedingungen für die MacWilliams-Eigenschaft um ein Vielfaches schwieriger werden, siehe [72]. Da die für unsere Zwecke entscheidenden Gewichte (das Hamming-Gewicht und auch das homogene Gewicht) aber die obige Eigenschaft erfüllen, werden wir im Folgenden nur noch Gewichte mit maximaler Symmetriegruppe $U_w = R^*$ untersuchen. Für beide Gewichte bzw. Distanzen ist also die Gruppe der linearen Isometrien isomorph zu der monomialen Gruppe $(R^*)^n \rtimes S_n$. Das weitere Vorgehen ließe sich aber durchaus analog für beliebige U_w-monomiale Gruppen umsetzen. Wir verzichten hierauf zur einfacheren Verständlichkeit der weiteren Argumentationen.

2.3.30 Hilfssatz. *Ist $w : R \to \mathbb{R}_0^+$ ein Gewicht mit $U_w = R^*$, so definiert die komponentenweise Anwendung eines Ringautomorphismus $\alpha \in \mathrm{Aut}(R)$ eine Isometrie auf dem metrischen Raum (R^n, d_w), $n \in \mathbb{N}$.*

Beweis. Für alle $a \in R$ ist $\mathrm{ht}(\alpha(a)) = \mathrm{ht}(a)$ und wegen $U_w = R^*$ auch $w(\alpha(a)) = w(a)$. Damit definiert die komponentenweise Anwendung von α aber ganz offensichtlich eine Isometrie von (R^n, d_w). □

Wir haben die Formulierung der MacWilliams-Eigenschaft eines Gewichts von J. Wood übernommen. Tatsächlich zeigt J. MacWilliams aber in ihrer Arbeit [52], dass sich jede semilineare Isometrie zwischen zwei gegebenen linearen Codes über einem Körper $\mathbb{F}_q$ zu einer semimonomialen Transformation auf ganz $\mathbb{F}_q^n$ fortsetzen lässt. Der folgende Hilfssatz zeigt, dass die MacWilliams-Eigenschaft im Fall von $U_w = R^*$ auch hinreichend ist. Er verallgemeinert die Aussage aus [73, Satz 2.16], welche nur das Hamming-Gewicht und das homogene Gewicht betrachtet. Die Beweisidee ist jedoch identisch.

2.3.31 Hilfssatz. *Es sei $w : R \to \mathbb{R}_0^+$ ein Gewicht mit $U_w = R^*$. Dann lässt sich jede semilineare Isometrie $f : (C, d_w) \to (R^n, d_w)$ zu jedem $C \leq R^n$ und $n \in \mathbb{N}$ zu einer semimonomialen Transformation von R^n fortsetzen.*

Beweis. Es sei w ein Gewicht mit der MacWilliams-Eigenschaft und $f : C \to {}_R R^n$ eine beliebige semilineare Isometrie für $C \leq R^n$ mit zugehörigen Ringautomorphismus $\alpha \in \mathrm{Aut}(R)$. Dann ist aber $f \circ \alpha^{-1}$ ebenfalls eine Isometrie von C und linear. Sie lässt sich wegen der MacWilliams-Eigenschaft des Gewichts aber über eine monomiale Matrix $A = \mathrm{diag}(\varphi)P^{(\pi)}$ für ein $(\varphi; \pi) \in (R^*)^n \rtimes S_n$ darstellen. Insgesamt ergibt sich also die zu beweisende Behauptung, dass f eine semimonomiale Transformation von R^n ist. □

2.3.32 Folgerung. *Ist $w : R \to \mathbb{R}_0^+$ ein Gewicht mit $U_w = R^*$, dann ist die Gruppe aller semilinearen Isometrien von (R^n, d_w) isomorph zu $(R^*)^n \rtimes (\mathrm{Aut}(R) \times S_n)$.*

2.3.33 Bemerkung. Hat das Gewicht w eine kleinere Symmetriegruppe $U_w \neq R^*$, so muss man sich auch bei der Operation der Automorphismengruppe von R auf diejenige Untergruppe einschränken, welche das Gewicht respektiert.

2.3.34 Bemerkung. Hat ein Gewicht w mit $U_w = R^*$ die MacWilliams-Eigenschaft, so können wir diese nicht nur dazu nutzen, den Äquivalenzbegriff auf die Gruppenoperation der semimonomialen Gruppe zurückzuführen. Sie zeigt auch, dass wir die Definition der Äquivalenz von zwei linearen Codes $C_0, C_1 \leq R^n$ allgemeiner über die Existenz einer semilinearen Isometrie $\iota : C_0 \to C_1$ definieren könnten, ohne dass dies zu veränderten Äquivalenzklassen führen würde.

Wir werden im Folgenden von Gewichten w bzw. Metriken d_w ausgehen, für welche die Symmetriegruppe $U_w = R^*$ maximal ist. Wir nennen dann zwei Codes $C_0, C_1 \leq R^n$

- *(semi-)linear isometrisch*, wenn es eine (semi-)monomiale Transformation von R^n gibt, die den einen auf den anderen überführt, bzw.
- *permutationsisometrisch*, wenn es eine Koordinatenpermutation gibt, welche den einen auf den anderen überführt.

2.3.35 Definition (symmetrisiertes Gewicht)**.** Das *symmetrisierte Gewicht*[3] eines Vektors $v \in R^n$ definieren wir als $w_{\mathrm{sym}}(v) := (a_0(v), \ldots, a_m(v))$ mit $a_i := |\{j \in [n] \mid \mathrm{per}(a_j) = i\}, \ \forall\, i \in [m+1]$.

2.3.36 Folgerung. *Jedes Gewicht w mit maximaler Symmetriegruppe $U_w = R^*$ lässt sich mit Hilfe des symmetrisierten Gewichts als Vektorprodukt*

$$w(v) = w_{\mathrm{sym}}(v) \cdot (w(\theta^m), \ldots, w(\theta^0))^T$$

schreiben.

Das Gewicht w selbst wird also nicht weiter Eingang in diese Arbeit finden. Wir werden vielmehr das symmetrisierte Gewicht w_{sym} benutzen, da es eindeutig die Bahnen der Gruppe aller semilinearen Isometrien auf ${}_R R^n$ beschreibt.

[3] Dies ist kein Gewicht im Sinne von Definition 2.3.22.

2.4. Komplexität der Probleme

Für eine Einführung in die Komplexitätstheorie verweisen wir den Leser auf eines der zahlreichen Lehrbücher zur theoretischen Informatik oder zu diesem speziellen Teilgebiet, etwa [71]. Zunächst ist es für weitere Komplexitätsuntersuchungen notwendig, das zu untersuchende Problem als ein Entscheidungsproblem zu formulieren, d.h. als eine Fragestellung, die eindeutig (entscheidbar) mit ja oder nein für alle Eingaben zu beantworten ist. Zum Beispiel definieren wir das Graphenisomorphieproblem GI als die Frage: „Sind zwei beliebige gegebene Graphen isomorph?“

Unter einer Eingabe zu einem Entscheidungsproblem P wollen wir nun die Fragestellung verstehen, die es im Konkreten zu beantworten gilt; im Beispiel der Graphenisomorphie also ein Vorgabe eines Paares (G_0, G_1) aus zwei Graphen $G_0, G_1 \in \binom{V}{2}$. Der Eingabe wird nun eine Eingabelänge zugeordnet, um die benötigte Rechenzeit auch in Relation zu der Problemgröße setzen zu können. Wir gehen daher davon aus, dass sich die Eingabe I auf natürliche Weise als eine Zeichenkette über einem endlichen Alphabet beschreiben lässt. Zum Beispiel werden die Graphen durch Adjazenzmatrizen oder lineare Codes durch Generatormatrizen beschrieben. Der Eingabe I ordnen wir dann die Länge $|I|$ dieser Zeichenkette zu. Für das Graphenisomorphieproblem hat somit die Eingabe (G_0, G_1) zum Beispiel die Länge $|(G_0, G_1)| = 2 \cdot 2^{n^2}$.

Eine *Turingmaschine* ist nun ein Modell der Informatik um den Begriff der Rechenzeit eines Algorithmus zu standardisieren. Bei einer *deterministischen Turingmaschine* ist die nächste durchgeführte Aktion eindeutig durch den aktuellen Zustand der Maschine und des Zeichens an der aktuellen Position des Lesekopfs definiert. Wir sagen, eine deterministische Turingmaschine (ein deterministischer Algorithmus) löst ein Problem P in Polynomialzeit, falls ein Polynom $p \in \mathbb{R}[x]$ existiert, so dass sich für alle Eingaben I die maximal notwendige Anzahl von Schritten der Turingmaschine zur Eingabe I durch $p(|I|)$ nach oben abschätzen lässt.

Im Gegensatz dazu kann eine *nichtdeterministische Turingmaschine* in jedem Zustand für die aktuelle Eingabe stets zwischen zwei Aktionen wählen, wobei es keine Vorschrift gibt, wie diese Aktion ausgewählt wird. Eine nichtdeterministische Turingmaschine löst ein Entscheidungsproblem zur Eingabe I, falls sie bei der Korrektheit der Aussage die Antwort „ja“ auf einem zulässigen Rechenweg erreichen kann. Ist die Antwort auf die Aussage „nein“, so führen auch alle Rechenwege zur Antwort „nein“. Eine nichtdeterministische Turingmaschine arbeitet in Polynomialzeit, falls sie die Rechnung auf allen zulässigen Rechenwegen in Polynomialzeit in der Eingabelänge beendet.

Wir sagen, ein Entscheidungsproblem P_0 sei schwerer als das Entscheidungsproblem P_1, falls wir jede Eingabe I für das Problem P_1 in Polynomialzeit auf eine Eingabe $J(I)$ zu P_0 transformieren können, so dass I genau dann mit „ja“ beantwortet wird, wenn auch $J(I)$ mit „ja“ beantwortet wird. Wir sagen hierzu auch, dass wir das Entscheidungsproblem P_1 in Polynomialzeit auf das Entscheidungsproblem P_0 zurückführen.

Mit diesen Definitionen lassen sich nun die bekannten Komplexitätsklassen

P die Klasse aller Entscheidungsprobleme, welche in Polynomialzeit auf einer deterministischen Turingmaschine lösbar sind,

NP die Klasse aller Entscheidungsprobleme, die in Polynomialzeit auf einer nichtdeterministischen Turingmaschine lösbar sind,

NP-schwer die Klasse aller Entscheidungsprobleme P_0, für die jedes Problem P_1 aus der Klasse **NP** in Polynomialzeit auf P_0 zurückgeführt werden kann und

NP-vollständig die Klasse aller **NP-schweren** Entscheidungsprobleme in **NP**

definieren. Entscheidungsprobleme aus der Klasse **NP** lassen sich auch dadurch charakterisieren, dass jede „Ja"-Antwort in Polynomialzeit auf einer deterministischen Turingmaschine verifizierbar ist.

2.4.1 Beispiel. Das Graphenisomorphieproblem GI ist in **NP**, denn wir können in Polynomialzeit *überprüfen*, ob eine gegebene Permutation π der Knoten einen Isomorphismus zwischen beiden Graphen definiert.

Die Problemklasse **P** ist in **NP** enthalten und sie beinhaltet die einfacheren Fragestellungen. **NP-schwere** Probleme zeichnen sich dadurch aus, dass sie mindestens so schwer sind wie alle anderen Probleme aus **NP**.

Eine der wichtigsten Fragestellungen der Mathematik ist die Entscheidung, ob **P**=**NP** gilt. Das Clay Mathematics Institute[4] hat einen Geldpreis in Höhe von einer Millionen Dollar für die Lösung ausgeschrieben. In der offiziellen Problembeschreibung [14] findet sich auch die Einordnung des Graphenisomorphieproblems (GI):

> *„There are interesting examples of* **NP** *problems not known to be either in* **P** *or* **NP-complete**. *One example is the graph isomorphism problem: Given two undirected graphs, determine whether they are isomorphic."*

Diese Eigenschaft der Graphenisomorphie wird auch in [71] behandelt. Für lineare Codes definieren wir analog zur Graphenisomorphie folgende Entscheidungsprobleme in Abhängigkeit von dem Kettenring R und des gewählten Isometriebegriffs:

PCE_R: Sind Γ und Γ' Generatormatrizen R-linearer Codes, entscheide ob der von Γ erzeugte Code *permutationsisometrisch* zu dem von Γ' erzeugten Code ist.

LCE_R: Sind Γ und Γ' Generatormatrizen R-linearer Codes, entscheide ob der von Γ erzeugte Code *linear isometrisch* zu dem von Γ' erzeugten Code ist.

SCE_R: Sind Γ und Γ' Generatormatrizen R-linearer Codes, entscheide ob der von Γ erzeugte Code *semilinear isometrisch* zu dem von Γ' erzeugten Code ist.

[4] `http://www.claymath.org`

Wenige Arbeiten untersuchen die Komplexität der oben definierten Probleme aus der Codierungstheorie. In [62] liefern Petrank und Roth zunächst einen Beweis für die Tatsache, dass – unter einer vermutlich gültigen Annahme über die Struktur der Komplexitätsklassen – das von ihnen definierte Entscheidungsproblem „Code Equivalence“ (= Vereinigung aller $\mathrm{PCE}_{\mathbb{F}_q}$ für alle endlichen Körper $\mathbb{F}_q$) nicht in die Klasse der **NP-vollständigen** Probleme einzuordnen ist:

> *„It is believed that the polynomial-time hierarchy does not collapse, and thus we end up with the conclusion that Code Equivalence is unlikely to be* **NP-complete***.“*

Eine ähnliche Schlussfolgerung findet sich auch in [71] für das Graphenisomorphieproblem. Andererseits geben die Autoren in [62] aber auch einen Hinweis darauf, dass es sich um ein nicht allzu einfaches Problem handeln kann:

> *„Yet, we do state also a negative result, namely, that Code Equivalence is also unlikely to be too easy. We do this by relating Code Equivalence to the Graph Isomorphism problem. [...] The problem of deciding efficiently (i.e., in polynomial time) whether two graphs are isomorphic is a notoriously open question in Computer Science. The problem has been studied extensively in recent decades, but the state of the art is that there is no known efficient algorithm for determining whether two given graphs are isomorphic.“*

Da Petrank und Roth die Graphenisomorphie nur über eine Polynomialzeitreduktion auf $\mathrm{PCE}_{\mathbb{F}_2}$ zurückführen, ließe sich durchaus argumentieren, dass das Problem $\mathrm{PCE}_{\mathbb{F}_q}$ für einen anderen endlichen Körper $\mathbb{F}_q, q > 2$ möglicherweise leichter zu beantworten ist. Hierzu liefert jedoch [34] einen analoge Polynomialzeitreduktion des Graphenisomorphieproblems auf $\mathrm{PCE}_{\mathbb{F}_q}$.

Es sei im Folgenden R ein fest vorgegebener Kettenring und es bezeichne $I_m \in R^{m\times m}$ die Einheitsmatrix der Dimension $m \times m$ zu $m \in \mathbb{N}$. Wir wollen nun das Resultat von [34] nicht nur auf PCE_R sondern auch auf die Entscheidungsprobleme LCE_R und SCE_R verallgemeinern.

2.4.2 Definition. Es sei $A \in \{0,1\}^{n\times m}$ eine Inzidenzmatrix eines Graphen G mit n Punkten und m Kanten. Wir definieren zu A die Generatormatrix

$$\Gamma^{(A)} := \begin{pmatrix} I_m & I_m & \mathbf{1}_{\mathbf{m}\times\mathbf{1}} & A \end{pmatrix} \in R^{m\times(2m+n+1)}$$

eines linearen Codes $C^{(A)} \leq R^{2m+n+1}$ der Länge $2m+n+1$.

2.4.3 Hilfssatz. *$\Gamma^{(A)}$ ist bis auf Zeilenvertauschungen und -skalierungen mit Einheiten die eindeutige Generatormatrix von $C^{(A)}$, welche folgende Eigenschaften erfüllt:*

$$\forall\, v \in R^m : \mathrm{w_H}(v) = 1 \iff \mathrm{w_H}\big(v\Gamma^{(A)}\big) = 5 \tag{2.2}$$

Beweis. Zunächst zeigen wir, dass $\Gamma^{(A)}$ die Bedingung (2.2) erfüllt: Sei $v \in R^m$ beliebig. Ist $\mathrm{w_H}(v) = 1$ so ist $v\Gamma^{(A)}$ ein Vielfaches einer Zeile von $\Gamma^{(A)}$ und somit $\mathrm{w_H}\big(v\Gamma^{(A)}\big) = 2{+}1{+}2 = 5$. Ansonsten erhalten wir für das Hamming-Gewicht die folgende Abschätzung

$$\mathrm{w_H}\big(v\Gamma^{(A)}\big) \geq 2\,\mathrm{w_H}(v) + \mathrm{w_H}(vA) \geq \begin{cases} 4+2, & \text{falls } \mathrm{w_H}(v) = 2 \\ 6, & \text{falls } \mathrm{w_H}(v) \geq 3. \end{cases}$$

Ist Γ' eine weitere Generatormatrix von $C^{(A)}$ mit der Eigenschaft (2.2), so ist $\Gamma' = B\Gamma^{(A)}$ für ein $B \in \mathrm{GL}_m(R)$. Für die i-te Zeile $B_{i,*}$, $i \in [m]$, von B gilt nun aber $\mathrm{w_H}\big(B_{i,*}\Gamma^{(A)}\big) = \mathrm{w_H}\big(\Gamma'_{i,*}\big) = 5$ und somit $\mathrm{w_H}(B_{i,*}) = 1$. Damit ist B aber wegen seiner Invertierbarkeit zwingend das Produkt einer Permutationsmatrix und einer invertierbaren Diagonalmatrix. □

2.4.4 Satz. GI *besitzt eine Polynomialzeitreduktion auf* SCE_R.

Beweis. Wir zeigen, dass zwei gegebene Inzidenzmatrizen A_0 und A_1 genau dann isomorphe Graphen darstellen, wenn die Generatormatrizen $\Gamma^{(A_0)}$ und $\Gamma^{(A_1)}$ semilinear isometrische Codes erzeugen. Diese Reduktion ist offensichtlich in Polynomialzeit zu berechnen.

Zunächst gehen wir davon aus, dass die Inzidenzmatrizen A_0 und A_1 isomorphe Graphen darstellen. Es existieren also Permutationen $\pi \in S_m$ und $\sigma \in S_n$ mit $A_1 = P^{(\pi)}A_0P^{(\sigma^{-1})}$. Dann ist aber

$$P^{(\pi)}\Gamma^{(A_0)}\begin{pmatrix} P^{(\pi^{-1})} & & \\ & P^{(\pi^{-1})} & \\ & & 1 \\ & & & P^{(\sigma^{-1})} \end{pmatrix} = \Gamma^{(A_1)}$$

und damit erzeugen $\Gamma^{(A_0)}$ und $\Gamma^{(A_1)}$ permutationsisometrische Codes.

Für die Rückrichtung nehmen wir an, dass die Codes $C^{(A_0)}$ und $C^{(A_1)}$ semilinear isometrisch sind. Dann existiert eine Matrix $B \in \mathrm{GL}_m(R)$ und ein Gruppenelement

$$(\varphi; \alpha, \sigma) \in (R^*)^{2m+n+1} \rtimes (\mathrm{Aut}(R) \times S_{2m+n+1})$$

mit

$$\Gamma^{(A_1)} = B\big((\varphi;\alpha,\sigma)\Gamma^{(A_0)}\big) = B\alpha\big(\Gamma^{(A_0)}\big)P^{(\sigma^{-1})}\,\mathrm{diag}(\varphi)^{-1} = B\Big(\Gamma^{(A_0)}P^{(\sigma^{-1})}\,\mathrm{diag}(\varphi)^{-1}\Big).$$

Die Matrix $\Gamma^{(A_0)}P^{(\sigma^{-1})}\,\mathrm{diag}(\varphi)^{-1}$ ist eine Generatormatrix zu $C^{(A_1)}$ mit der Eigenschaft (2.2). Daraus folgt nun mit dem Hilfssatz, dass $B = P^{(\pi)}\,\mathrm{diag}(\psi)$ das Produkt einer Permutationsmatrix $P^{(\pi)} \in R^{m\times m}$ und einer invertierbaren Diagonalmatrix $\mathrm{diag}(\psi) \in R^{m\times m}$ ist.

Der Einsvektor $\big(\Gamma^{(A_1)}\big)_{*,2m} = \mathbf{1_{m\times 1}}$ sorgt nun dafür, dass $\mathrm{diag}(\psi)$ konstant auf der Diagonalen ist, also $\mathrm{diag}(\psi) = rI_m$ für ein $r \in R^*$ gilt. Damit erhalten wir

$$\Gamma^{(A_1)} = r \cdot P^{(\pi)}\Gamma^{(A_0)}P^{(\sigma^{-1})}\,\mathrm{diag}(\varphi)^{-1}.$$

Da $P^{(\pi)}\Gamma^{(A_0)}P^{(\sigma^{-1})}$ und $\Gamma^{(A_1)}$ nur Einträge aus $\{0_R, 1_R\}$ besitzen, können wir ohne Beschränkung der Allgemeinheit[5] $\mathrm{diag}(\varphi) = rI_m$ annehmen. Somit haben wir eine Gleichung der Gestalt

$$\Gamma^{(A_1)} = P^{(\pi)}\Gamma^{(A_0)}P^{(\sigma^{-1})}$$

erreicht. Die R-linearen Codes $C^{(A_0)}$ und $C^{(A_1)}$ sind also auch permutationsisometrisch.

Unter Berücksichtigung der Automorphismen[6] von $C^{(A_0)}$ können wir weiter annehmen, dass σ die Koordinaten $\{0,\ldots,m-1\}, \{m,\ldots,2m-1\}, \{2m\}$ und $\{2m+1,\ldots,2m+n\}$ mengenweise fix lässt. Unter dieser Voraussetzung ist dann

$$\Gamma^{(A_1)} = P^{(\pi)}\Gamma^{(A_0)}\begin{pmatrix} P^{(\pi^{-1})} & & & \\ & P^{(\pi^{-1})} & & \\ & & 1 & \\ & & & P^{(\rho^{-1})} \end{pmatrix}$$

für eine Permutation $\rho \in S_n$. Es ist also $A_1 = P^{(\pi)}A_0P^{(\rho^{-1})}$ und damit sind A_0 und A_1 Inzidenzmatrizen von isomorphen Graphen. □

2.4.5 Folgerung. *Für einen beliebigen endlichen Kettenring R besitzt* GI *eine Polynomialzeitreduktion sowohl auf* PCE_R *als auch auf* LCE_R.

Beweis. Eine zentrale Aussage des Beweises zum vorausgegangenen Satz ist, dass die von $\Gamma^{(A_0)}$ und $\Gamma^{(A_1)}$ erzeugten R-linearen Codes $C^{(A_0)}$ und $C^{(A_1)}$ genau dann semilinear isometrisch sind, wenn sie auch permutationsisometrisch bzw. linear isometrisch sind.

A_0 und A_1 sind also genau dann Inzidenzmatrizen von isomorphen Graphen, wenn $C^{(A_0)}$ und $C^{(A_1)}$ permutationsisometrisch bzw. linear isometrisch sind. □

Damit haben wir gezeigt, dass die Isomorphieprobleme für lineare Codes über endlichen Kettenringen mindestens genauso schwer sind wie das Graphenisomorphieproblem.

Die oben eingeführten Komplexitätsklassen machen Aussagen über die worst-case-Laufzeiten der Probleme. Ein Problem liegt nicht in **P**, sobald es zu jedem Polynom $p \in \mathbb{R}[x]$ eine nicht leere Teilmenge der erlaubten Eingaben gibt, welche für das exponentielle Laufzeitverhalten verantwortlich ist. Es ist also weiterhin durchaus möglich, dass für fast alle Eingaben die Anzahl der Rechenschritte durch das Polynom p nach oben beschränkt ist. Daher möchten wir im Folgenden auch kurz auf die durchschnittliche Komplexität der Probleme eingehen.

Mit Hilfe des Support-Splitting-Algorithmus [66] wird in [61] bewiesen, dass für einen beliebigen Körper $\mathbb{F}_q$ die Probleme $\mathrm{PCE}_{\mathbb{F}_q}$ für fast alle Eingaben in Polynomialzeit gelöst werden können. Für die Körper $\mathbb{F}_3$ und $\mathbb{F}_4$ wird in [67] ein ähnliches Resultat für die Problemstellungen $\mathrm{LCE}_{\mathbb{F}_3}$, $\mathrm{LCE}_{\mathbb{F}_4}$ und $\mathrm{SCE}_{\mathbb{F}_4}$ erreicht. Gleichzeitig geben die Autoren N. Sendrier und D. Simos aber auch eine Vermutung über die Körper $\mathbb{F}_q, q \geq 5$ ab:

[5] Nullspalten müssen berücksichtigt werden, d.h. isolierte Knoten des Graphen. Diese können aber mit beliebigen Einheiten multipliziert werden, ohne den Code $C^{(A_0)}$ zu ändern.

[6] Es können weitere Einheitsvektoren oder Einsvektoren in der Matrix A_0 enthalten sein.

2.4.6 Vermutung (aus [67])**.** Zu gegebenem $q \geq 5$ sind die Probleme $\mathrm{LCE}_{\mathbb{F}_q}$ und $\mathrm{SCE}_{\mathbb{F}_q}$ für fast alle Eingaben schwer[7].

Aus der Reduktion auf das Graphenisomorphieproblem schließen wir, dass es vermutlich zu jeder deterministischen Turingmaschine zur Lösung von SCE_R und zu jedem Polynom $p \in \mathbb{R}[X]$ ein Paar von Generatormatrizen (Γ_0, Γ_1) mit Eingabelänge n gibt, so dass die Maschine mehr als $p(n)$ Rechenschritte zur Beantwortung benötigt. Falls sich die Vermutung 2.4.6 für die Körper $\mathbb{F}_q, q \geq 5$ bzw. auch für beliebige Kettenringe R bewahrheitet, ist sogar davon auszugehen, dass fast alle Eingaben (Γ_0, Γ_1) dieses Verhalten zeigen. Insbesondere vermuten wir auch, dass sich gerade die für die Codierungstheorie interessanten Codes in dieser Hinsicht ungünstig verhalten werden.

Wir können also nicht davon ausgehen, einen Kanonisierer zu entwickeln, welcher in der Lage ist, die kanonische Form einer beliebigen Generatormatrix in Polynomialzeit in der Länge der Eingabe zu berechnen. Die Vermutung 2.4.6 rechtfertigt unser weiteres Vorgehen, einen Algorithmus zu entwickeln, welcher stets für alle Eingaben mit exponentieller Laufzeit arbeitet.

Da die Komplexitätstheorie uns also kein Maß für die Bewertung des Algorithmus an die Hand gibt, können wir uns nur mit den wenigen konkurrierenden Systemen für lineare Codes über endlichen Körpern vergleichen. Für beliebige Kettenringe können wir die Güte des Algorithmus nur über seine praktische Anwendbarkeit auf gewisse interessante Probleminstanzen verifizieren, siehe Kapitel 6.

[7] Fast alle Eingaben werden von einem Lösungsalgorithmus mit exponentiellen Aufwand bearbeitet.

3. Kanonisierungsalgorithmen

Zunächst sei G eine beliebige endliche Gruppe, welche auf einer endlichen Menge X operiere. Die Operation sei ohne Beschränkung der Allgemeinheit treu, d.h. der Kern $N := \cap_{x \in X} \mathrm{Stab}_G(x)$ der Gruppenoperation ist trivial. Liegt diese Situation nicht vor, so ist N ein Normalteiler in G und wir können stattdessen auch die induzierte[1] Gruppenoperation von G/N auf X untersuchen. Außerdem wollen wir auf X immer eine gegebene Totalordnung voraussetzen.

Zur Kanonisierung des vorliegenden kombinatorischen Objekts $x \in X$ unter der Operation von G werden wir die Prinzipien des Verfeinerns und Individualisierens von Partitionen (partition refinement, individualization) entwickeln. Diese algorithmische Grundidee bildet die Basis zur Definition eines Suchbaums, mit dessen Hilfe der kanonische Repräsentant und die Automorphismengruppe von x bestimmt werden können. Laut einer Aussage in [56] trat diese Strategie erstmals in [63] im Zusammenhang mit der Isomorphieerkennung bei Graphen auf. Auch alle gegenwärtigen (wettbewerbsfähigen) Algorithmen zur Berechnung eines kanonischen Repräsentanten eines Graphen basieren immer noch auf dieser Grundlage. In [56] wird hierzu eine detaillierte Laufzeitanalyse der wichtigsten Implementierungen vorgenommen.

Die in Abschnitt 3.2 beschriebene Formulierung ist zunächst eine Verallgemeinerung des Graphenkanonisierers [55]. Sie kombiniert die zeitgleich erschienenen Diskussionen aus [35] und [42] und ergänzt diese mit eigenen Ideen. Wir gehen wie folgt vor:
Zunächst beschreiben wir in Abschnitt 3.1 einige grundlegende Ideen, die wir bei der Entwicklung eines Kanonisierers einbringen können. Dies folgt im Wesentlichen der Beschreibung aus [35]. Der Autor R. Gugisch baut aus diesen Einzelkomponenten den Kanonisierer aus Abschnitt 3.2 für die Operation ${}_GX$ „von unten“ auf. Nachteil dieser Herangehensweise ist aber, dass der Korrektheitsbeweis für den Gesamtalgorithmus mit allen zusätzlichen Modifikationen schwer zu führen ist. Daher übernehmen wir diesen Teil aus [42]. In dieser Beschreibung, siehe Abschnitt 3.2, wird größerer Wert auf das Gesamtbild gelegt, ohne zu sehr im Detail zu versinken. Mit diesem Blick „von oben“ wird die Korrektheit des Vorgehens sofort offensichtlich.

Im Anschluss modifizieren wir diese Beschreibung aus [42] über mehrere Schritte hinweg derart, dass die Korrektheit weiterhin leicht ersichtlich bleibt und die Formulierung auch zu einem praxistauglichen Kanonisierer führt. Zum Beispiel beschreiben wir, wie man die Automorphismen des zu kanonisierenden Objekts $x \in X$ zur Verbesserung des Laufzeitverhaltens einbringen kann.

[1] $gNx := gx$.

Anschließend behandeln wir den Spezialfall $G = S_n$ bzw. $G = S_{\mathfrak{P}}$ für eine kanonische Young-Untergruppe[2] von S_n. Hier werden wir im wesentlichen Datentypen untersuchen und den Zusammenhang zu der Graphenkanonisierung herstellen. Im letzten Abschnitt passen wir den so gewonnenen Algorithmus an unsere codierungstheoretischen Gegebenheiten an, d.h. wir werden die Operation eines semidirekten Produkts $G \rtimes S_n$ auf einer Menge X^n näher untersuchen.

3.1. Grundbausteine der Kanonisierung

In diesem Abschnitt sollen mögliche Herangehensweisen an die Entwicklung eines Kanonisierers beschrieben werden. Zunächst wollen wir aber anmerken, dass wir über den Bahnenalgorithmus – wie etwa in [50] beschrieben – immer die Möglichkeit haben, einen solchen Algorithmus zu definieren. Wir berechnen die Bahn Gx des Elements $x \in X$ und setzen den kanonischen Repräsentanten gleich dem minimalen Element der Bahn Gx. Offensichtlich bildet die Bahnenlänge den entscheidenden Faktor bei der Laufzeitabschätzung zu dieser Herangehensweise.

Wir wollen also nun untersuchen, mit welchen Hilfsmitteln wir dieses naive Vorgehen verbessern können.

3.1.1. Kanonisierung mittels Homomorphieprinzip

Als sehr wichtiges Hilfsmittel wird sich das nachfolgende Homomorphieprinzip für Gruppenoperationen erweisen. Wir werden es nur für einen G-Homomorphismus formulieren, weisen aber auch auf die Gültigkeit für Homomorphismen von Gruppenoperationen hin. Diese Verallgemeinerung können wir leicht aus Bemerkung 2.1.1 herleiten.

3.1.1 Fakt (Homomorphieprinzip, [50])**.** *Es sei $f : X \to Y$ ein G-Homomorphismus. Liegen $y, y' \in f(X)$ in derselben Bahn unter der Operation von G, so schneiden die Urbilder dieselben Bahnen von G auf X. Andernfalls sind die getroffenen Bahnen disjunkt. Zwei Elemente x, x' liegen genau dann in derselben Bahn von G auf X, wenn*

- *es ein Gruppenelement $g_1 \in G$ gibt mit $g_1 f(x) = f(x')$ und*
- *ein Gruppenelement $g_2 \in \mathrm{Stab}_G(f(x))$ mit $g_2 x = g_1^{-1} x'$.*

3.1.2 Folgerung. *Ist $T_{G\backslash\backslash Y}$ eine Transversale von ${}_GY$ und $T_{\mathrm{Stab}_G(y)\backslash\backslash f^{-1}(y)}$ eine Transversale des Urbilds eines jeden $y \in T_{G\backslash\backslash Y}$ unter dem Stabilisator $\mathrm{Stab}_G(y)$, so ist*

$$T_{G\backslash\backslash X} := \bigcup_{y \in T_{G\backslash\backslash Y}} T_{\mathrm{Stab}_G(y)\backslash\backslash f^{-1}(y)}$$

eine Transversale der Operation ${}_GX$.

[2] Definition folgt.

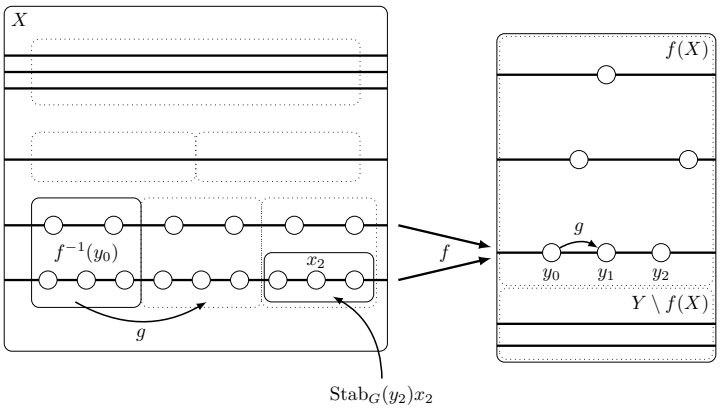


Abbildung 3.1.: Homomorphieprinzip

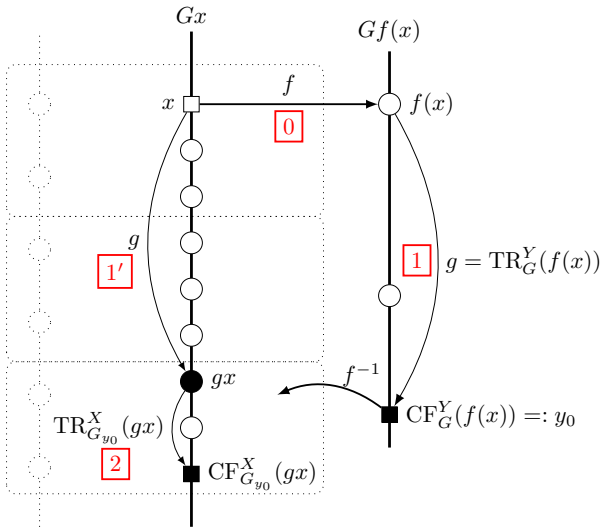


Abbildung 3.2.: Kanonisieren mittels Homomorphieprinzip; Aufspalten

Abbildung 3.1 zeigt die wichtigsten Eigenschaften eines G-Homomorphismus f. In dieser Darstellung deuten die Horizontalen die Bahnen von G auf X bzw. auf Y an.

Aus dem Homomorphieprinzip können wir nun einen Kanonisierer Can_G^X für die Kanonisierung auf ${}_GX$ definieren:

3.1.3 Folgerung (Kanonisierer mittels Homomorphieprinzip; Aufspalten)**.**
Es sei $f : X \to Y$ ein G-Homomorphismus und Can_G^Y ein Kanonisierer für die Operation von G auf Y. Weiter sei für jedes $y \in T_{G\backslash\backslash Y} := \mathrm{Can}_G^Y(Y)$ ein Kanonisierer

$$\mathrm{Can}^{f^{-1}(y)}_{\mathrm{Stab}_G(y)} = \left(\mathrm{CF}^{f^{-1}(y)}_{\mathrm{Stab}_G(y)}, \mathrm{TR}^{f^{-1}(y)}_{\mathrm{Stab}_G(y)}, \mathrm{Stab}_{\mathrm{Stab}_G(y)}\right)$$

zu der Operation des Stabilisators $\mathrm{Stab}_G(y)$ auf dem Urbild $f^{-1}(y)$ von y gegeben. Dann

definiert der folgende Algorithmus einen Kanonisierer Can_G^X *für die Gruppenoperation von* G *auf* X*: Zu einer Eingabe* $x \in X$

1. *berechne* $(y_0, g, \mathrm{Stab}_G(f(x))) := \mathrm{Can}_G^Y(f(x))$,
2. *setze*
   - $G_{y_0} := \mathrm{Stab}_G(y_0) = g\,\mathrm{Stab}_G(f(x))g^{-1}$,
   - $C_G(x) := \mathrm{CF}_{G_{y_0}}^{f^{-1}(y_0)}(gx)$,
   - $T_G(x) := \mathrm{TR}_{G_{y_0}}^{f^{-1}(y_0)}(gx)g$,
   - $S_G(x) := g^{-1}\,\mathrm{Stab}_{G_{y_0}}(gx)g$
3. *und definiere* $\mathrm{Can}_G^X(x) := (C_G(x), T_G(x), S_G(x))$.

Abbildung 3.2 zeigt eine Darstellung zu dem Vorgehen.

Beweis. Wir beweisen zunächst, dass die Funktion $C_G : X \to X, x \mapsto C_G(x)$ eine Kanonisierung realisiert. Hierzu sei $x \in X$ und $g_0 \in G$ beliebig vorgegeben. Mit $(y_0, g, \mathrm{Stab}_G(f(x))) := \mathrm{Can}_G^Y(f(x))$ sei, wie oben, das Resultat der Kanonisierung im Bildbereich bezeichnet.

Man rechnet leicht nach, dass das Gruppenelement $h := \mathrm{TR}_G^Y(f(g_0x))g_0g^{-1}$ den kanonischen Repräsentanten y_0 fix lässt, also $h \in G_{y_0} := \mathrm{Stab}_G(y_0)$ gilt. Hieraus folgt

$$\begin{aligned} C_G(g_0x) &= \mathrm{CF}_{G_{y_0}}^{f^{-1}(y_0)}(\mathrm{TR}_G^Y(f(g_0x))g_0x) = \mathrm{CF}_{G_{y_0}}^{f^{-1}(y_0)}(hgg_0^{-1}g_0x) \\ &= \mathrm{CF}_{G_{y_0}}^{f^{-1}(y_0)}(hgx) = \mathrm{CF}_{G_{y_0}}^{f^{-1}(y_0)}(gx) = C_G(x) \end{aligned}$$

und somit der Beweis für die G-Invarianz der Funktion C_G. Da $C_G(x)$ auch für jedes $x \in X$ in der Bahn Gx liegt, definiert C_G eine Kanonisierung.

Die Aussage, dass die Funktion T_G eine Transporterabbildung zur Kanonisierung C_G definiert, zeigt man analog. Für den Stabilisator folgert man aus $\mathrm{Stab}_G(gx) \subseteq \mathrm{Stab}_G(f(gx))$ die Gleichheit der Mengen

$$\mathrm{Stab}_G(gx) = \mathrm{Stab}_{\mathrm{Stab}_G(f(gx))}(gx) = \mathrm{Stab}_{G_{y_0}}(gx).$$

Konjugation mit g^{-1} führt zur angegebenen Definition von $S_G(x)$. □

3.1.4 Bemerkung. R. Laue gibt in der Arbeit [50] einen Algorithmus zur Berechnung der Transversale von $T_{G\backslash\backslash X}$ über das Homomorphieprinzip an. Im Wesentlichen fasst obiger Kanonisierer die lokal notwendigen Schritte zusammen. In [50, Theorem 1.10] beweist er, dass dieses Verfahren unter Standardbedingungen zu einer Logarithmierung des Aufwands führt.

Dies gilt auch für den Kanonisierer: Gehen wir vereinfacht zunächst davon aus, dass die Laufzeit der genutzten Kanonisierer – die etwa über Einsatz des Bahnenalgorithmus gewonnen wurden – linear (mit gleicher Konstante) von der Bahnenlänge abhängig sind. Dann hat der direkte Einsatz auf ${}_GX$ einen zu

$$|Gx| = \frac{|G|}{|\operatorname{Stab}_G(x)|} = \frac{|G|}{|\operatorname{Stab}_G(f(x))|} \cdot \frac{|\operatorname{Stab}_G(f(x))|}{|\operatorname{Stab}_G(x)|}$$

proportionalen Aufwand. Wendet man jedoch das Homomorphieprinzip an, so erhält man eine Abschätzung für die Laufzeit in der Größenordnung von

$$|\operatorname{Stab}_G(f(gx))gx| + |Gf(x)| = \frac{|G|}{|\operatorname{Stab}_G(f(gx))|} + \frac{|\operatorname{Stab}_G(f(x))|}{|\operatorname{Stab}_G(x)|},$$

welche im Fall von $|\operatorname{Stab}_G(x)| < |\operatorname{Stab}_G(f(x))| = |\operatorname{Stab}_G(f(gx))| < |G|$ somit durchaus mit einer logarithmischen Aufwandsreduktion gleichzusetzen ist.

In der Praxis sind natürlich komplexere Einflussgrößen zu berücksichtigen, zum Beispiel der Aufwand für die Berechnung von $f(x)$, oder auch ob für das Bild ein wesentlich effizienterer Kanonisierer als für das Urbild zur Verfügung steht.

3.1.5 Beispiel. Wir geben einige Beispiele für das Aufspalten von Bahnen und möglicher Kanonisierer:

- Ist $X = Y \times Z$ mit komponentenweiser Operation der Gruppe G, so definiert die Projektion $\Pi_Y : Y \times Z \to Y, \quad (y, z) \mapsto y$ auf die erste Komponente einen G-Homomorphismus. Die Anwendung des Homomorphieprinzips entspricht nun dem Vorgehen, zunächst für (y, z) einen kanonischen Repräsentanten $y_0 = gy$ der ersten Komponente zu bestimmen. Anschließend wird die Gruppenoperation auf der modifizierten zweiten Komponente gz auf den Stabilisator $\operatorname{Stab}_G(y_0)$ eingeschränkt.

- Dieses Vorgehen lässt sich leicht auf n-fache kartesische Produkte $\times_{i\in[n]} X^{(i)}$ übertragen. Man bildet sukzessiv die Elemente x_i, $i \in [n]$ des Vektors $(x_0, \ldots, x_{n-1})$ auf ihren kanonischen Repräsentanten ab. Anschließend wird das berechnete Transporterelement auf x angewandt, bevor man zur nächsten Komponente x_{i+1} übergeht. Natürlich setzt dieses Vorgehen voraus, dass man für die entsprechend auftretenden Stabilisatoren Kanonisierer auf $X^{(i)}$ zur Verfügung hat.

- Damit ist sofort klar, wie man bei Folgen $f_0, \ldots, f_{n-1}$ von G-Homomorphismen bzw. bei Vorliegen eines G-Homomorphismus der Gestalt $f = (f_0, \ldots, f_{n-1}) : X \to Y := \times_{i=0}^{n} Y^{(i)}$ einen effizienten Kanonisierer für ${}_GX$ gewinnt. Man berechnet $\operatorname{Can}_G^Y(f(x))$ sukzessiv über die Einzelschritte im Bildbereich analog zu den vorausgehenden Beobachtungen für kartesische Produkte. Dieses Vorgehen ist als surjektive Auflösung bekannt, siehe etwa [43, 9.4.2 Surjective Resolution].

R. Laue [50] beschreibt, wie man das Homomorphieprinzip, auch in umgekehrter Richtung, zur Definition einer Transversalen im Bildbereich nutzen kann: Ist $f : X \to Y$ ein surjektiver G-Homomorphismus, so kann man aus einer gegebenen Transversale von G auf X einen Algorithmus zur Berechnung einer Transversale von G auf Y angeben. Er bezeichnet diesen als die Verschmelzung von Bahnen.

Ebenso kann man ausgehend von einem Kanonisierer Can_G^X auf X über einen surjektiven G-Homomorphismus einen Kanonisierer Can_G^Y für die Operation auf Y entwerfen:

3.1.6 Hilfssatz (Kanonisierer mittels Homomorphieprinzip; Verschmelzung)**.**
Es sei $f : X \to Y$ ein surjektiver G-Homomorphismus und Can_G^X ein Kanonisierer zur Operation von G auf X. Weiter sei die Transversale $\mathrm{Can}_G^X(X)$ durch $<$ totalgeordnet. Dann definiert der folgende Algorithmus einen Kanonisierer Can_G^Y für die Gruppenoperation von G auf Y: Zu einer Eingabe $y \in Y$

1. *bestimme ein $x_0 \in f^{-1}(y)$ mit $\mathrm{CF}_G^X(x_0) = \min\{\mathrm{CF}_G^X(x) \mid x \in f^{-1}(y)\}$ und die Menge $X_y := \big\{x \in f^{-1}(y) \mid \mathrm{CF}_G^X(x) = \mathrm{CF}_G^X(x_0)\big\}$,*
2. *setze*
   - $C_G(y) := f\big(\mathrm{CF}_G^X(x_0)\big)$
   - $T_G(y) := \mathrm{TR}_G^X(x_0)$
   - $S_G(y) := \big\langle \mathrm{Stab}_G(x_0) \cup \{\mathrm{TR}_G^X(x_0)^{-1}\,\mathrm{TR}_G^X(x) \mid x \in X_y\}\big\rangle$
3. *und definiere $\mathrm{Can}_G^Y(y) := (C_G(y), T_G(y), S_G(y))$.*

Abbildung 3.3 zeigt eine Visualisierung des so gewonnenen Kanonisierers.

Beweis. Es sei $y \in Y$ beliebig und $y' := gy$ ein Element der gleichen Bahn. Da für alle $\overline{x} \in \mathrm{Can}_G^X(X)$ die Äquivalenz $f^{-1}(y) \cap G\overline{x} \neq \emptyset \iff f^{-1}(y') \cap G\overline{x} \neq \emptyset$ gilt, sind auch die Mengen $\{\mathrm{CF}_G^X(x) \mid x \in f^{-1}(y)\}$ und $\{\mathrm{CF}_G^X(x) \mid x \in f^{-1}(y')\}$ identisch. Damit ist auch die Wahl des kanonischen Repräsentanten offensichtlich G-invariant und $T_G(y)$ ein Transporterelement zu dieser Kanonisierung.

Es bleibt die Behauptung zum Stabilisator: Die Erzeuger von $S_G(y)$ sind offensichtlich Elemente des Stabilisators $\mathrm{Stab}_G(y)$, es bleibt also nur die Inklusion $\mathrm{Stab}_G(y) \subseteq S_G(y)$ zu zeigen. Hierzu wählen wir ein $g \in \mathrm{Stab}_G(y)$ beliebig. Es ist $gx_0 \in X_y$ und somit auch $\mathrm{TR}_G^X(x_0)^{-1}\,\mathrm{TR}_G^X(gx_0)$ im Erzeugendensystem. Außerdem gilt $\mathrm{TR}_G^X(x_0)^{-1}\,\mathrm{TR}_G^X(gx_0)g \in \mathrm{Stab}_G(x_0)$ und damit die Aussage. □

3.1.7 Bemerkung. Ist in der vorausgegangen Folgerung der Stabilisator $\mathrm{Stab}_G(y)$ bereits zu Beginn bekannt, so kann man sich in Schritt 1. auf eine Transversale T der Bahnen von $\mathrm{Stab}_G(y)$ auf $f^{-1}(x)$ zurückziehen, d.h. man bestimmt ein $x_0 \in T$ mit $\mathrm{CF}_G^X(x_0) = \min\{\mathrm{CF}_G^X(x) \mid x \in T\}$.

Schließlich wollen wir auch Beispiele für die Anwendung der Verschmelzung angeben:

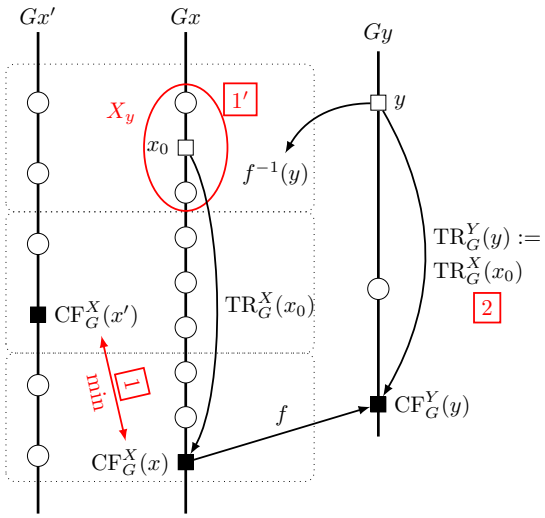


Abbildung 3.3.: Kanonisieren mittels Homomorphieprinzip; Verschmelzen

3.1.8 Beispiel (Verschmelzung)**.**

- Ist $N \trianglelefteq G$ ein Normalteiler in G, so operiert die Gruppe G auch auf $N\backslash\backslash X$ via $g \cdot Nx := Ngx$.

  Haben wir also einen Kanonisierer für die Operation von G auf X zur Verfügung, so liefert uns der G-Homomorphismus $f : X \to N\backslash\backslash X, x \mapsto Nx$ einen Kanonisierer für G auf $N\backslash\backslash X$. Hier beobachten wir auch, dass die Bahnen von G auf X und G auf $N\backslash\backslash X$ in Bijektion stehen. Damit ist aber $\{\mathrm{CF}_G^X(x') \mid x' \in f^{-1}(Nx)\}$ einelementig und der Spezialfall $X_{Nx} = f^{-1}(Nx) = Nx$ liegt vor.

- Es operiert neben G auch die Gruppe G/N auf $N\backslash\backslash X$ und der Homomorphismus $(g \mapsto gN, f)$ von Gruppenoperationen definiert gerade obige Operation von G auf $N\backslash\backslash X$.

- Es sei R ein Kettenring mit Kettenlänge m und $0 < k \leq n$. Es operiert die Gruppe

  $$(\mathrm{GL}_k(R) \times (R^*)^n) \rtimes (\mathrm{Aut}(R) \times S_n)$$

  auf der Menge $R^{k\times n,(m,\dots,m)}$ aller Generatormatrizen der freien linearen Codes vom Rang k und es ist $\mathrm{GL}_k(R)$ ein Normalteiler der Gruppe. Die Bahnenmenge $\mathrm{GL}_k(R)\backslash\backslash R^{k\times n,(m,\dots,m)}$ repräsentiert aber gerade die Menge aller freien linearen Codes vom Rang k der Länge n. Die Faktorgruppe $(R^*)^n \rtimes (\mathrm{Aut}(R) \times S_n)$ gibt uns den Isomorphiebegriff für diese Objekte.

- Es operiere G auf einer Menge X. Dann operiert $G \times S_n$ auf natürliche Weise auch auf X^n. Nun ist S_n aber ein Normalteiler dieser Gruppe und wir erhalten aus einem Kanonisierer $\mathrm{Can}^X_{G\times S_n}$ einen Kanonisierer $\mathrm{Can}^{S_n\backslash\backslash X^n}_G$ für die Operation der

Gruppe $G \simeq (G \times S_n)/S_n$ auf der Menge $S_n \backslash\backslash X^n$. Eine Bahn $S_n(x_0, \ldots, x_{n-1}) \in S_n \backslash\backslash X^n$ können wir aber eineindeutig mit dem Inhalt des Vektors beschreiben, d.h. über eine Aufzählung $\{\{x_0, \ldots, x_{n-1}\}\}$ aller Einträge in ihrer Vielfachheit, die keinen Wert auf die Reihenfolge legt. Eine solche Struktur bezeichnen wir als *Multimenge* und wir notieren diese mit doppelt geschweiften Klammern, um sie von gewöhnlichen Mengen zu unterscheiden.

Somit haben wir einen alternativen Ansatz zur Kanonisierung von n-elementigen (Multi-)Mengen unter der Operation von G gegeben. Da die Darstellung einer (Multi-)Menge ohnehin innerhalb des Computers über eine geeignete Anordnung der Elemente realisiert werden muss, macht es überdies Sinn, diese zusätzliche algebraische Struktur über die Gruppenoperation mit S_n in dem Algorithmenentwurf zu berücksichtigen.

Den Kanonisierer $\mathrm{Can}^X_{G \times S_n}$ aus dem letzten Beispiel kann man wiederum über das Aufspalten zu dem Homomorphismus $((g, \pi) \mapsto g, x \mapsto Gx)$ gewinnen. Wir bezeichnen mit $\mathcal{M}_n(X)$ die Menge aller n-elementigen Multimengen einer Menge X. Es ergibt sich folgendes Bild von Homomorphismen von Gruppenoperationen und die Möglichkeit zur Definition von Kanonisierern:

$$\begin{array}{ccccc} {}_{S_n}(G \backslash\backslash X^n) & \xleftarrow[Gx \hookleftarrow x]{\pi \hookleftarrow (g,\pi)} & {}_{G \times S_n}X^n & \xrightarrow[x \mapsto S_n x]{(g,\pi) \mapsto g} & {}_G(S_n \backslash\backslash X^n) \simeq {}_G(\mathcal{M}_n(X)) \\ \mathrm{Can}^{G \backslash\backslash X^n}_{S_n} & \xrightarrow{\text{Aufspalten}} & \mathrm{Can}^{X^n}_{G \times S_n} & \xrightarrow{\text{Verschmelzen}} & \mathrm{Can}^{S_n \backslash\backslash X^n}_{G} \end{array}$$

Dieses Beispiel dient uns zur Motivation des weiteren Vorgehens, jedoch sei darauf hingewiesen, dass in diesem Spezialfall die Bahnenmengen $G \backslash\backslash (S_n \backslash\backslash X^n)$, $(G \times S_n) \backslash\backslash X^n$ und $S_n \backslash\backslash (G \backslash\backslash X^n)$ ohnehin in Bijektion stehen, d.h. keine Aufspaltungen und Verschmelzungen im eigentlichen Sinne stattfinden. Es handelt sich hierbei um äquivalente Formulierungen des gleichen Problems.

Schließlich möchten wir noch darauf hinweisen, dass wir tatsächlich die Kanonisierung zu Gruppenoperationen von Gruppen $G \rtimes S_n$ auf Mengen X^n zum Ziel haben werden. Da $G \trianglelefteq G \rtimes S_n$ weiterhin Normalteiler ist, ist die Gewinnung des Kanonisierers $\mathrm{Can}^{X^n}_{G \rtimes S_n}$ aus $\mathrm{Can}^{G \backslash\backslash X^n}_{S_n}$ über das Aufspalten weiterhin möglich.

Der Übergang zu einer Operation von G auf $S_n \backslash\backslash X^n$ ist aber, wegen der fehlenden Normalteilereigenschaft von S_n, nicht ohne weitere Voraussetzungen möglich. In den von uns zu untersuchenden Situationen (z.B. semilineare Isometrie von linearen Codes) ist die Gruppe G jedoch selbst ein semidirektes Produkt $H^n \rtimes G'$ mit folgenden Annahmen:

- Die symmetrische Gruppe operiert auf $H^n \rtimes G'$ nur über eine Permutation der Komponente H^n, d.h. $(H^n \rtimes G') \rtimes S_n = H^n \rtimes (G' \times S_n)$.

- Die Gruppe G' operiert simultan auf allen Komponenten von X^n und die Gruppe H^n komponentenweise.

Dann können wir zunächst den Normalteiler $H^n \rtimes S_n$ heraus teilen und erhalten damit eine Operation von G' auf $(H^n \rtimes S_n)\backslash\backslash X^n$. Diese Menge können wir aber als $S_n\backslash\backslash(H^n\backslash\backslash X^n) = \mathcal{M}_n(H\backslash\backslash X)$ interpretieren.

3.1.9 Beispiel. Die Gruppe

$$(\mathrm{GL}_k(R) \times (R^*)^n) \rtimes (\mathrm{Aut}(R) \times S_n)$$

operiert auf der Menge $R^{k\times n}$ aller $k \times n$-Matrizen. Wir setzen $H := R^*$ und $G' := \Gamma\mathrm{L}_k(R) = (\mathrm{GL}_k(R) \rtimes \mathrm{Aut}(R))$. Damit erhalten wir also eine Operation der Gruppe $\Gamma\mathrm{L}_k(R)$ auf $\mathcal{M}_n(R^*\backslash\backslash R_R^k)$, via

$$(A, \alpha) \cdot \{\{R^* \star x_0, \ldots, R^* \star x_{n-1}\}\} := \{\{R^* \star A\alpha(x_0), \ldots, R^* \star A\alpha(x_{n-1})\}\}.$$

Für ein $v \in R_R^k$ können wir die Bahn $R^* \star v = \{v\varphi^{-1} \mid \varphi \in R^*\}$ aber gerade mit dem zyklischen Rechtsmodul vR identifizieren.

3.1.10 Folgerung. *Zwei Generatormatrizen $\Gamma, \Gamma' \in R^{k\times n,(m,\ldots,m)}$ erzeugen genau dann semilinear isometrische Codes, wenn die Multimengen*

$$\{\{\Gamma_{*,i}R \mid i \in [n]\}\} \quad \text{und} \quad \{\{\Gamma'_{*,i}R \mid i \in [n]\}\}$$

der von den Spalten erzeugten zyklischen R-Rechtsmoduln unter der Gruppenoperation von $\Gamma\mathrm{L}_k(R)$ isomorph sind.

Damit haben wir ein weiteres Theorem aus [52] auf R-lineare Codes erweitert und den wohlbekannten Zusammenhang zwischen dem Äquivalenzbegriff der Codierungstheorie und dem der projektiven Geometrie hergestellt. Dieser Zusammenhang wird etwa in [37, 45, 73] und vielen weiteren Arbeiten zur Konstruktion von guten linearen Codes aus Punktkonfigurationen der projektiven Rechts-Hjelmslev-Geometrie ausgenutzt, siehe auch Kapitel 6.1.

Gleichartig verhält es sich, wenn wir in der projektiven Geometrie von den eindimensionalen Unterräumen (=Punkten) zu r-dimensionalen Unterräumen übergehen. Dies wird uns eine Bijektion der Äquivalenzklassen von Network-Codes über $\mathbb{F}_q$ und den Äquivalenzklassen von $\mathbb{F}_q$-linearen Codes über dem Alphabet $\mathbb{F}_{q^r}$ liefern, siehe Kapitel 6.3.

3.1.2. Kanonisierung über Untergruppen

Nicht immer ist es möglich, einen leicht zu berechnenden G-Homomorphismus f anzugeben, für den die Kanonisierung im Bildbereich effizient zu implementieren ist. Zumeist liegt für diesen dann die Situation vor, dass $\mathrm{Stab}_G(f(x))$ gleich G ist und somit keine Information gewonnen werden kann. Wir wollen nun zeigen, wie sich das Auftreten dieser Situation umgehen lässt.

Zunächst nehmen wir an, es sei eine Untergruppe U von G gegeben, für die wir bereits einen effizienten Kanonisierer Can_U entwickelt haben. Des Weiteren benötigen wir für unser Vorgehen eine Transversale T der Rechtsnebenklassen $U\backslash G$. Dann können wir den Kanonisierer $\mathrm{Can}_G = (\mathrm{CF}_G, \mathrm{TR}_G, \mathrm{Stab}_G)$ wie folgt definieren:

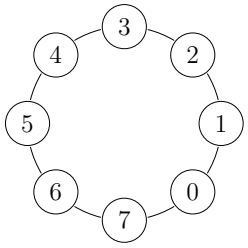
- Zu einem gegebenen $x \in X$ berechne $\mathrm{CF}_G(x) := \min_{t\in T} \mathrm{CF}_U(tx)$ und speichere in T_0 diejenigen Transversalenelemente, welche zum Minimum führen;
- setze $\mathrm{TR}_G(x) := \mathrm{TR}_U(t_0x)t_0$ für ein $t_0 \in T_0$;
- erzeuge den Stabilisator $\mathrm{Stab}_G(x)$ über

$$t_0^{-1}\,\mathrm{Stab}_U(t_0x)t_0 \quad \text{und} \quad \{\mathrm{TR}_G(x)^{-1}\,\mathrm{TR}_U(tx)t \mid t \in T_0\}$$

Wir bezeichnen dieses Vorgehen als das *Heben* eines Kanonisierers für die Operation mit U zu einem Kanonisierer für G. Im Wesentlichen zerlegen wir das Problem also in $|U\backslash G| = \frac{|G|}{|U|}$ Teilprobleme in der Hoffnung, dass diese einfacher zu lösen sind.

Wir können dieses Vorgehen zum Beispiel dann einsetzen, wenn wir keinen geeigneten G-Homomorphismus für die Anwendung des Homomorphieprinzips angeben können, jedoch sehr wohl einen U-Homomorphismus zur Verfügung haben.

3.1.11 Beispiel. Es sei der folgende Graph $\Gamma = ([8], E)$ gegeben:



Für diesen Graph ist offensichtlich der Zykel $(0, 1, 2, \ldots, 7) \in S_8$ ein Automorphismus. Somit werden wir die Knoten durch Anwendung eines beliebigen S_8-Homomorphismus nicht unterscheiden können.

Zerlegen wir jedoch das Problem mit Hilfe der Untergruppe $U = \mathrm{Stab}_{S_8}(0)$, so brechen wir die Symmetrie und wir können einen effizienteren Kanonisierer für U über den U-Homomorphismus

$$f : [2]^{\binom{[8]}{2}} \to \mathbb{N}^8, \qquad E \mapsto (\text{Länge eines kürzesten Pfades in } E \text{ von } i \text{ nach } 0)_{i\in[8]}$$

entwerfen. Wir erhalten dann $f(\Gamma) = (0, 1, 2, 3, 4, 3, 2, 1)$. Bei der Anwendung des Homomorphieprinzips sortieren wir diesen Vektor lexikographisch aufsteigend, etwa über die Anwendung der Permutation $\pi = (2, 3, 6, 4, 7)$. Anschließend wenden wir diese Permutation auch auf Γ an und schränken uns im weiteren Verlauf auf den Stabilisator

$$U' = \mathrm{Stab}_U((0, 1, 1, 2, 2, 3, 3, 4)) = \langle (1, 2), (3, 4), (5, 6)\rangle$$

des kanonischen Repräsentanten $f(\pi\Gamma) = (0, 1, 1, 2, 2, 3, 3, 4)$ ein.

Es ist leicht einzusehen, dass die kanonischen Repräsentanten $\mathrm{CF}_U(t\Gamma)$ für alle Rechtstransversalenelemente $t \in T$ von $U\backslash G$ in diesem Beispiel identisch sind. Über dieses Vorgehen werden wir also maximal $|T| \cdot \mathrm{Stab}_U(f(tx)) = 8 \cdot 8 = 64$ Permutationen untersuchen.

Ein weiterer solcher Schritt mit der Untergruppe $\mathrm{Stab}_{U'}(1)$ in der Kanonisierung für U reduziert in diesem Beispiel die Anzahl der zu betrachtenden Permutationen sogar auf die Mächtigkeit 16 der Automorphismengruppe.

3.2. Partitionen und Verfeinerungen

In diesem Abschnitt wollen wir nun aufzeigen, wie wir die vorangegangenen Ideen zur Definition eines effizienten Kanonisierers zusammenfügen. Wir werden dazu einen Backtrack-Algorithmus zum systematischen Durchlauf der Gruppe G formulieren. Das Vorgehen lässt sich für ein $x \in X$ am besten über die Definition eines zugehörigen Suchbaums $T(x, G)$ analog zu [42] beschreiben. Wieder sei mit $\mathcal{L}(G)$ die Menge aller Untergruppen von G bezeichnet und mit $\mathcal{C}(G)$ die Menge aller Rechtsnebenklassen aller Untergruppen H von G.

Die Knoten des Suchbaums $T(x, G)$ werden von einer Teilmenge der Rechtsnebenklassen $Hg \in \mathcal{C}(G)$ gebildet. Die Untergruppe H repräsentiert dabei genau diejenigen Gruppenelemente, welche wir in dieser Phase des Algorithmus noch zur Anwendung bringen wollen. Das Gruppenelement g wurde an dieser Stelle bereits auf x angewandt. Der Baum selbst beziehungsweise der Ablauf des Algorithmus wird induktiv über die folgenden Basisoperationen definiert:

Partitionierung Die Nebenklasse Hg mit $|H| > 1$ wird zerlegt in eine disjunkte Menge von Nebenklassen $\{H'h_0g, \ldots, H'h_{u-1}g\}$ einer echten Untergruppe $H' < H$. Die Menge $\{h_0, \ldots, h_{u-1}\}$ sei dabei eine beliebige Rechtstransversale von H' in H. Dieses Vorgehen haben wir oben als das Heben eines Kanonisierers für die Operation mit der Untergruppe H' zu einem Kanonisierer für die Operation mit H beschrieben.

Offensichtlich lässt sich diese Operation bereits durch Angabe der Untergruppe H' eindeutig beschreiben. Um gleiche Resultate bei isomorphen Eingaben garantieren zu können, setzen wir voraus, dass diese Operation über die Bereitstellung einer G-Invarianten

$$I : X \times \mathcal{C}(G) \to \mathcal{L}(G),\ (x, Hg) \mapsto H' \text{ mit } H' < H \tag{3.1}$$

eindeutig bestimmt ist. Dabei ist die Operation von G auf dem Definitionsbereich $X \times \mathcal{C}(G)$ über die Definition

$$g_0 \star (x, Hg) := (g_0x, Hgg_0^{-1}) \text{ für alle } g_0 \in G, (x, Hg) \in X \times \mathcal{C}(G)$$

gegeben.

Verfeinerung Unter einer Verfeinerung verstehen wir das Ersetzen einer Nebenklasse Hg durch eine Teilmenge $H'hg$ mit $H' \leq H$ und $h \in H$. Auch diese Vorgehen kann mit Hilfe eines G-Homomorphismus

$$V : X \times \mathcal{C}(G) \to \mathcal{C}(G), (x, Hg) \mapsto (H'hg) \text{ mit } H' \leq H \text{ und } h \in H \tag{3.2}$$

beschrieben werden. Wie wir später zeigen werden, erhalten wir Verfeinerungen über die Anwendung des Homomorphieprinzips.

Es lässt sich leicht verifizieren, dass die Bedingungen, welche in [42] an diese Basisoperationen gestellt werden, äquivalent zu unserer Forderung der G-Invarianz bzw. G-Homomorphie der Funktionen I und V sind. Nun können wir den eigentlichen Aufbau des Suchbaums $T(x, G)$ beschreiben:

3.2.1 Definition. Es sei $x \in X$ beliebig. Wir definieren einen Wurzelbaum $T(x, G)$ induktiv wie folgt:

1. Die Wurzel des Baums $T(x, G)$ wird von der Verfeinerung $V(x, G)$ gebildet.
2. Für einen Knoten Hg mit $|H| > 1$ sei $H' := I(x, G)$ und $\{h_0, \ldots, h_{u-1}\}$ eine Rechtstransversale von H' in H. Die Kinder $\{V(x, H'h_0g), \ldots, V(x, H'h_{u-1}g)\}$ von Hg werden dann über die Verfeinerung der Partitionierung definiert.

3.2.2 Fakt ([42], Theorem 5.30)**.** *Für $x \in X$ und $g_0 \in G$ induziert die Gruppenoperation von G auf $\mathcal{C}(G)$ einen Isomorphismus der Suchbäume $T(x, G)$ und $T(g_0x, G)$, d.h.*

- *die Wurzel von $T(g_0x, G)$ ist $g_0 \star V(x, G) = V(x, G)g_0^{-1}$,*
- *$Hg \in \mathcal{C}(G)$ ist genau dann ein Knoten von $T(x, G)$, wenn Hgg_0^{-1} ein Knoten in $T(g_0x, G)$ ist und*
- *$\{H'hg, Hg\}$ ist genau dann eine Kante in $T(x, G)$, wenn $\{H'hgg_0^{-1}, Hgg_0^{-1}\}$ eine Kante in $T(g_0x, G)$ ist.*

Eine eingängige Visualisierung dieses Theorems zeigt Abbildung 3.4.

Aufgrund unserer Definition werden die Blattknoten von $T(x, G)$ von einelementigen Mengen $\{g\}$ mit $g \in G$ gebildet. Wir wollen die Menge dieser Gruppenelemente mit

$$L(x, G) := \{g \in G \mid \{g\} \text{ Blatt in } T(x, G)\} \subseteq G$$

bezeichnen.

3.2.3 Folgerung. *Die Abbildung $x \mapsto \{gx \mid g \in L(x, G)\}$ ist G-invariant. Insbesondere definiert also die Zuordnung $\mathrm{CF}_G(x) := \min\{gx \mid g \in L(x, G)\}$ einen kanonischen Repräsentanten für $x \in X$.*

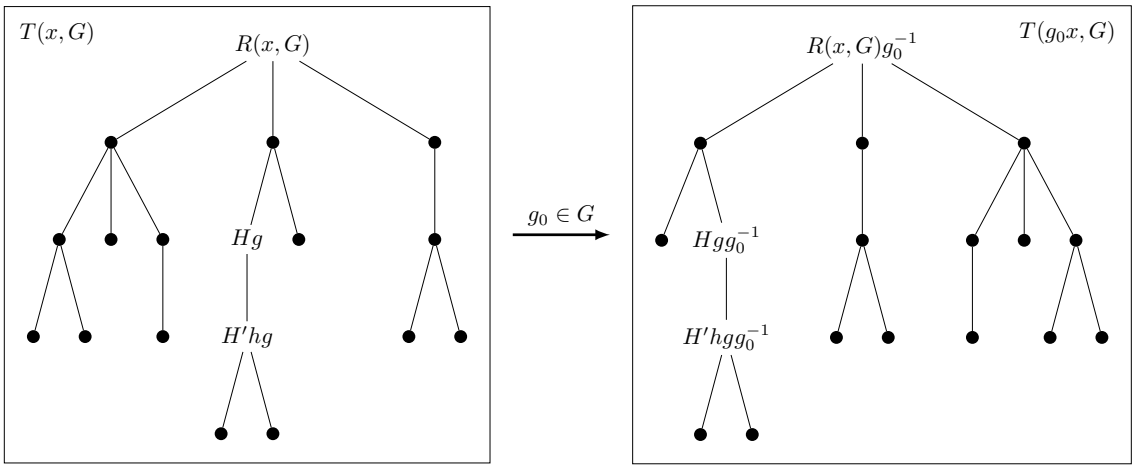


Abbildung 3.4.: Illustration von Fakt 3.2.2, siehe auch [42, Figure 5.3]

Beweis. Ist $g_0 \in G$ beliebig, so ist nach Fakt 3.2.2 die Menge der Blattknoten von $T(g_0x, G)$ bestimmt durch Rechtsmultiplikation der Blätter von $T(x, G)$ mit g_0^{-1}. Es ist also $L(g_0x, G) = L(x, G)g_0^{-1}$ und somit

$$\{gx \mid g \in L(x,G)\} = \{gg_0^{-1}g_0x \mid g \in L(x,G)\} = \{g'g_0x \mid g' \in L(g_0x,G)\}.$$

Damit haben wir die G-Invarianz gezeigt und somit auch, dass die Funktion $\mathrm{CF}_G(x) := \min\{gx \mid g \in L(x,G)\}$ eine Kanonisierung definiert. □

Wir werden im Folgenden zu $x \in X$ mit

$$L_0(x,G) := \{g \in L(x,G) \mid gx = \mathrm{CF}_G(x)\}$$

die Menge aller Transporterelemente bezeichnen. Entscheidend für die Laufzeit des Algorithmus ist nun die Frage, wie wir möglichst viele Äste des Suchbaums $T(x, G)$ abschneiden können, ohne wesentliche Informationen zu verlieren.

Zum einen können wir bekannte Automorphismen $A \leq \mathrm{Stab}_G(x)$ des Objekts x ausnutzen, da es offensichtlich genügt, eine Transversale von $L_0(x,G)/A$ zu untersuchen. Eine Diskussion dieser Methode findet sich im Abschnitt 3.2.2. Zunächst wollen wir uns aber der Frage:

„Ist $L_0(x,G) \cap Hg = \emptyset$ für einen Knoten Hg von $T(x,G)$?“

zuwenden. Offensichtlich können wir – für eine beliebig gewählte Totalordnung $(X, \leq)$ – dies nur dann entscheiden, falls wir $\mathrm{CF}_G(x)$ und Hgx bereits kennen.

Nun erfolgte aber die Wahl der Totalordnung auf X zur Definition des Minimums in Folgerung 3.2.3 willkürlich. Wir werden also versuchen diese Wahl unserem Problem anzupassen, d.h. wir wollen eine Totalordnung wählen, welche einen effizienten Test auf $L_0(x,G) \cap Hg = \emptyset$ ermöglicht. Dazu beobachten wir zunächst, dass wir die Verfeinerung $V : X \times \mathcal{C}(G) \to \mathcal{C}(G)$ auch als Anwendung des Homomorphieprinzips sehen können.

3.2.4 Satz. *Für alle $H \in \mathcal{L}(G)$ sei ein H-Homomorphismus $f_H : X \mapsto Y$ gegeben sowie ein Kanonisierer $y \mapsto (\mathrm{CF}_H(y), \mathrm{TR}_H(y), \mathrm{Stab}_H(y))$. Dann ist*

$$V : X \times \mathcal{C}(G) \to \mathcal{C}(G), \quad (x, Hg) \mapsto \mathrm{Stab}_H(\mathrm{CF}_H(f_H(gx)))\,\mathrm{TR}_H(f_H(gx))g \tag{3.3}$$

ein G-Homomorphismus und erfüllt somit die Bedingungen an eine Verfeinerung.

Beweis. Zunächst zeigen wir die Wohldefiniertheit der Abbildungsvorschrift: Sind $g, g' \in G$ verschiedene Nebenklassenrepräsentanten von Hg, etwa $\overline{h}g = g'$ für ein $\overline{h} \in H$, so gilt

$$\underbrace{\mathrm{TR}_H(f_H(gx))}_{=:h} f_H(gx) = \mathrm{CF}_H(f_H(gx)) = \mathrm{CF}_H(f_H(g'x)) = \underbrace{\mathrm{TR}_H(f_H(g'x))}_{=:h'} f_H(g'x),$$

denn $f_H(gx)$ und $f_H(g'x) = \overline{h} f_H(gx)$ liegen in der gleichen H-Bahn. Wir folgern hieraus, dass $\overline{H} := \mathrm{Stab}_H(\mathrm{CF}_H(f_H(gx)))$ unabhängig von dem Nebenklassenrepräsentanten g ist. Es bleibt zu zeigen, dass $\overline{H}h'g' = \overline{H}hg$ ist. Wir zeigen hierzu $h'\overline{h}h^{-1} \in \overline{H}$:

$$\begin{aligned} h'\overline{h}h^{-1}\,\mathrm{CF}_H(f_H(gx)) &= h'\overline{h}h^{-1}hf_H(gx) = h'f_H(\overline{h}gx) = h'f_H(g'x) \\ &= \mathrm{CF}_H(f_H(g'x)) = \mathrm{CF}_H(f_H(gx)) \end{aligned}$$

Die Behauptung über die G-Homomorphie der Abbildung V beweist man folgendermaßen: Es seien $g_0 \in G$ und $(x, Hg) \in \mathcal{C}(G)$ beliebig. Dann gilt:

$$\begin{aligned} V(g_0 \star (x, Hg)) &= V(g_0 x, Hgg_0^{-1}) \\ &= \mathrm{Stab}_H\big(\mathrm{CF}_H(f_H(gg_0^{-1}g_0x))\big)\,\mathrm{TR}_H(f_H(gg_0^{-1}g_0x))gg_0^{-1} \\ &= V(x, Hg)g_0^{-1} = g_0 \star V(x, Hg) \end{aligned}$$

□

Im Folgenden setzen wir voraus, dass die Verfeinerung V über eine feste Wahl einer Familie $(f_H)_{H \in \mathcal{L}(G)}$ von H-Homomorphismen definiert sei. Dabei sei ohne Beschränkung der Allgemeinheit die Menge Y totalgeordnet[3].

Über die H-Homomorphismen können wir dann induktiv eine Bewertung $B(x, Hg)$ auf den Knoten des Suchbaums $T(x, G)$ einführen:

1. Die Wurzel $Hg = V(x, G)$ sei mit $B(x, Hg) = (f_G(gx))$ bewertet.
2. Ist $H'hg$ ein Knoten der Tiefe $i > 0$ im Baum $T(x, Hg)$ mit Vater Hg und Bewertung $B(x, Hg) = (y_0, \ldots, y_{i-1})$, so definieren wir
$$B(x, H'hg) := (y_0, \ldots, y_{i-1}, f_{I(x,Hg)}(hgx)).$$
3. Ist $\{g\}$ ein Blattknoten, so modifizieren wir dessen aktuelle Bewertung durch das zusätzliche Anfügen von gx.

[3]Dies können wir immer erreichen, da wir $Y = \bigcup_{H \in \mathcal{L}(G)} f_H(X)$ als endliche Vereinigung endlicher Mengen wählen können.

3.2.5 Hilfssatz. *Ist Hg ein Knoten in $T(x, G)$ mit Bewertung $B(x, Hg)$ und $g_0 \in G$ beliebig, so trägt der Knoten Hgg_0^{-1} in $T(g_0x, G)$ eine identische Bewertung $B(g_0x, Hgg_0^{-1}) = B(x, Hg)$.*

Beweis. Ist $g_0 \in G$ beliebig und $H'hgg_0^{-1}$ ein Knoten in $T(g_0x, G)$, dann ist über eine Induktion über die Tiefe der Knoten leicht zu beweisen, dass die Bewertungsvektoren $B(g_0x, H'hgg_0^{-1})$ und $B(x, H'hg)$ übereinstimmen:

Im Induktionsschritt schließt man aus der G-Invarianz von I zunächst, dass $\overline{H} := I(x, Hg) = I(g_0x, Hgg_0^{-1})$ gelten muss. Damit ist aber offensichtlich die Gleichheit von $f_{\overline{H}}(hgg_0^{-1}g_0x) = f_{\overline{H}}(hgx)$ stets gegeben. Ist $H'hg = \{hg\}$ überdies ein Blatt, so ist auch dort die angefügte Bewertung $hgg_0^{-1}g_0x = hgx$ identisch. □

3.2.6 Folgerung. *Ist $g_0 \in \mathrm{Stab}_G(x)$ und $\{g\}$ ein Blatt von $T(x, G)$, so ist $\{gg_0\}$ ebenfalls ein Blatt von $T(x, G)$ mit Bewertung $B(x, \{gg_0\}) = B(x, \{g\})$.*

Beweis. Es definiert $g_0^{-1} \in \mathrm{Stab}_G(x)$ einen Automorphismus von $T(x, G)$. Nach dem vorangegangenen Hilfssatz ist somit $\{gg_0\}$ ebenfalls ein Blatt von $T(x, G) = T(g_0^{-1}x, G)$ mit einer identischen Bewertung. □

Durch die Blätter $\{g\}$ von $T(x, G)$ erhalten wir über den lexikographischen Vergleich der Bewertung $B(x, \{g\})$ eine wohldefinierte Totalordnung auf der Teilmenge $\{gx \mid g \in L(x, G)\}$ der Bahn Gx. Wir können diese zur Minimumbildung und damit zur eindeutigen Festlegung eines kanonischen Repräsentanten heranziehen. Diese Ordnung auf X hat nun den Vorteil, dass wir bereits beim Durchlauf des Suchbaums nicht optimale Äste erkennen und abschneiden können:

3.2.7 Folgerung. *Es seien Hg und $H'g'$ Knoten der gleichen Tiefe i des Suchbaums $T(x, G)$. Dann gilt*

$$B(x, Hg) < B(x, H'g') \Longrightarrow H'g' \cap L_0(x, G) = \emptyset.$$

Zusammenfassend erhalten wir also den folgenden Kanonisierer Can_G für die Gruppenoperation von G zu einer Eingabe $x \in X$:

- Bilde einen Suchbaum $T(x, G)$ über eine fest gewählte Partitionierungsvorschrift I und Verfeinerung V unter eventueller Berücksichtigung des Abschneidekriteriums aus Folgerung 3.2.7. (Wir geben hierzu auch eine ausführliche, weiterführende Diskussion in Abschnitt 3.2.3).

- Bestimme die Menge

  $$L_0(x, G) := \{g \in L(x, G) \mid B(x, \{g\}) \leq B(x, \{\overline{g}\}) \ \forall\ \overline{g} \in L(x, G)\}$$

  aller Blätter mit minimaler Bewertung.

- Wähle ein Element $\mathrm{TR}_G(x) \in L_0(x, G)$ beliebig.

- Setze $\mathrm{CF}_G(x) := \mathrm{TR}_G(x)x$.
- Berechne $S_G(x) := \{\mathrm{TR}_G(x)^{-1}g \mid g \in L_0(x, G)\}$.

3.2.8 Satz. *Die oben erklärte Funktion*

$$\begin{aligned}\mathrm{Can}_G : X &\to X \times G \times \mathcal{L}(G)\\ x &\mapsto (\mathrm{CF}_G(x), \mathrm{TR}_G(x), S_G(x))\end{aligned}$$

definiert eine Kanonisierung für die Gruppenoperation von G auf X.

Beweis. Die Korrektheit der Aussage zu dem kanonischen Repräsentanten und dem gewählten Transporterelement folgt sofort aus der vorangegangenen Diskussion. Die Menge $S_G(x)$ ist offensichtlich eine Teilmenge von $\mathrm{Stab}_G(x)$. Ist umgekehrt $g_0 \in \mathrm{Stab}_G(x)$ beliebig, so gilt

$$L_0(x, G)g_0^{-1} = L_0(g_0x, G) = L_0(x, G).$$

Damit existiert ein Gruppenelement $g \in L_0(x, G)$ mit $\mathrm{TR}_G(x) = gg_0^{-1}$ und es ist $g_0 = \mathrm{TR}_G(x)^{-1}g \in S_G(x)$. □

3.2.1. Zur Kanonizität unter isomorphen Gruppenoperationen

Wir wollen nun besprechen, inwieweit der von uns beschriebene Algorithmus zur Kanonisierung von der Darstellung der Gruppenoperation abhängt. Wir betrachten also eine weitere Gruppenoperation von G' auf X', welche zu der ursprünglichen Operation von G auf X isomorph ist, d.h. es gibt einen Gruppenisomorphismus $\Psi : G' \to G$ und eine Bijektion $\Phi : X' \to X$, welche mit den Gruppenoperationen verträglich sind: $\Phi(g'x') = \Psi(g')\Phi(x')$ für alle $g' \in G'$ und $x' \in X'$.

Als Motivation dient uns das nachfolgende Beispiel, welches wir in Abschnitt 5.2.3 dann wieder aufgreifen werden.

3.2.9 Beispiel. Wir wollen annehmen, dass wir lineare Codes über isomorphen Kettenringen R und R' betrachten, die etwa durch den Ringisomorphismus $\alpha : R' \to R$ aufeinander übergeführt werden können. Dann entspricht die Abbildung Φ der komponentenweise Anwendung von α auf die Vektoren in R^n. Zusammen mit dem Gruppenisomorphismus

$$\begin{aligned}\Psi : (R'^*)^n \rtimes (\mathrm{Aut}(R') \times S_n) &\to (R^*)^n \rtimes (\mathrm{Aut}(R) \times S_n)\\ (\varphi; \beta, \pi) &\mapsto (\alpha(\varphi); \alpha\beta\alpha^{-1}, \pi)\end{aligned}$$

erhalten wir dann einen Isomorphismus von Gruppenoperationen.

Wir gehen nun zunächst zu der ursprünglichen Definition des Suchbaums zurück und nehmen an, dass die Verfeinerungsfunktion für die isomorphe Operation wie folgt definiert sei:

$$V' : X' \times \mathcal{C}(G') \to \mathcal{C}(G'), (x', H'g') \mapsto \Psi^{-1}(V(\Phi(x'), \Psi(H'g'))). \tag{3.4}$$

Außerdem sei die Partitionierung über

$$I' : X' \times \mathcal{C}(G') \to \mathcal{L}(G'),\ (x', H'g') \mapsto \Psi^{-1}(I(\Phi(x'), \Psi(H'g'))) \tag{3.5}$$

gegeben.

3.2.10 Hilfssatz. *Die Abbildung V' ist ein G'-Homomorphismus. Die Abbildung I' ist eine G'-Invariante.*

Beweis. Wir zeigen die Aussage exemplarisch für V'. Es sei $g'_0 \in G'$, $x' \in X'$ und $H'g' \in \mathcal{C}(G')$ beliebig. Dann gilt:

$$\begin{aligned} V'(g'_0 \star (x', H'g')) &= V'(g'_0 x', H'g'{g'_0}^{-1}) = \Psi^{-1}\Big(V(\Phi(g'_0 x'), \Psi(H'g'{g'_0}^{-1}))\Big) \\ &= \Psi^{-1}\big(V(\Psi(g'_0)\Phi(x'), \Psi(H'g')\Psi(g'_0)^{-1})\big) \\ &= \Psi^{-1}(\Psi(g'_0) \star V(\Phi(x'), \Psi(H'g'))) \\ &= g'_0 \star \Psi^{-1}(V(\Phi(x'), \Psi(H'g'))) = g'_0 \star V'(x', H'g') \end{aligned}$$ □

Ist nun $x' \in X'$ beliebig, so wollen wir mit $T'(x', G')$ den über V' und I' definierten Suchbaum zur Kanonisierung von $x' \in X'$ bezeichnen. Dabei wollen wir zunächst das Abschneiden von Teilbäumen über die Bewertungsfunktion noch unberücksichtigt lassen.

3.2.11 Hilfssatz. *Die Suchbäume $T'(x', G')$ und $T(\Phi(x'), G)$ sind isomorph, d.h.*

- *die Wurzel von $T(x', G')$ ist $\Psi^{-1}(V(\Phi(x'), G))$,*
- *$\Psi^{-1}(Hg)$ ist genau dann ein Knoten von $T(x', G')$, wenn $Hg \in \mathcal{C}(G)$ ein Knoten von $T(\Phi(x'), G)$ ist und*
- *$\{\Psi^{-1}(H'hg), \Psi^{-1}(Hg)\}$ ist genau dann eine Kante in $T(x', G')$, wenn $\{H'hg, Hg\}$ eine Kante in $T(\Phi(x'), G)$ ist.*

Beweis. Wir führen eine Induktion nach der Tiefe des Knotens Hg im Baum $T(\Phi(x'), G)$ durch. Ist der Knoten Hg die Wurzel von $T(\Phi(x'), G)$, so gilt

$$\Psi^{-1}(Hg) = \Psi^{-1}(V(\Phi(x'), G)) = V'(x', G').$$

Damit ist der Induktionsanfang bereits gezeigt.

Ist nun Hg ein beliebiger Knoten und $H \neq \{1_G\}$, so werden die Kinder von Hg durch einen Partitionierungsschritt und die nachfolgende Verfeinerung gewonnen. Ist T eine

beliebige Rechtstransversale von H nach $I(\Phi(x'), Hg)$, so ist $\Psi^{-1}(T)$ eine Rechtstransversale von $\Psi^{-1}(H)$ nach $I'(x', \Psi^{-1}(Hg)) = \Psi^{-1}(I(\Phi(x'), Hg))$.

Wir wählen nun $t \in T$ beliebig. Es gilt

$$\begin{aligned} & V'\big(x', I'(x', \Psi^{-1}(Hg))\Psi^{-1}(t)\Psi^{-1}(g)\big) = V'\big(x', \Psi^{-1}(I(\Phi(x'), Hg))\Psi^{-1}(tg)\big) \\ &= \Psi^{-1}\big(V\big(\Phi(x'), \Psi\big(\Psi^{-1}(I(\Phi(x'), Hg)tg)\big)\big)\big) \\ &= \Psi^{-1}(V(\Phi(x'), I(\Phi(x'), Hg)tg)) \end{aligned}$$

Auf der linken Seite dieser Gleichung betrachten wir einen Sohn von $\Psi^{-1}(Hg)$ im Baum $T'(x', G')$. Auf der rechten Seite steht ein Sohn $V(\Phi(x'), I(\Phi(x'), Hg)tg)$ von Hg in $T(\Phi(x'), G)$, auf welchen wir Ψ^{-1} anwenden. Damit sind die beiden weiteren Behauptungen bewiesen. □

Wie oben sei nun die Verfeinerung V für die Operation von G auf X über die Wahl einer Familie $(f_H)_{H \in \mathcal{L}(G)}$ von H-Homomorphismen definiert.

3.2.12 Hilfssatz. *Ist $H' \in \mathcal{L}(G')$ beliebig, so definiert $\widetilde{f}_{H'} := f_{\Psi(H')} \circ \Phi$ zusammen mit Ψ einen Homomorphismus von Gruppenoperationen. Damit können wir $\widetilde{f}_{H'}$ auf natürliche Weise[4] auch als H'-Homomorphismus interpretieren. Die von der Familie $(\widetilde{f}_{H'})_{H' \in \mathcal{L}(G')}$ gemäß Satz 3.2.4 induzierte Verfeinerung V' erfüllt dann die Bedingung* (3.4).

Beweis. Man rechnet leicht nach, dass $\widetilde{f}_{H'}$ die behaupteten Eigenschaften hat. Wir beweisen nur die Aussage zu der Verfeinerung V'. Hierzu sei $x' \in X'$ und $H'g' \in \mathcal{C}(G')$ beliebig, dann gilt:

$$\begin{aligned} & \Psi(V'(x', H'g')) = \Psi\Big(\mathrm{Stab}_{H'}\Big(\mathrm{CF}_{H'}(\widetilde{f}_{H'}(g'x'))\Big) \cdot \mathrm{TR}_{H'}(\widetilde{f}_{H'}(g'x')) \cdot g'\Big) \\ &= \mathrm{Stab}_{\Psi(H')}\big(\mathrm{CF}_{\Psi(H')}(f_{\Psi(H')}(\Phi(g'x')))\big) \cdot \mathrm{TR}_{\Psi(H')}(f_{\Psi(H')}(\Phi(g'x'))) \cdot \Psi(g') \\ &= \mathrm{Stab}_{\Psi(H')}\big(\mathrm{CF}_{\Psi(H')}(f_{\Psi(H')}(\Psi(g')\Phi(x')))\big) \cdot \mathrm{TR}_{\Psi(H')}(f_{\Psi(H')}(\Psi(g')\Phi(x'))) \cdot \Psi(g') \\ &= V(\Phi(x'), \Psi(H'g')) \end{aligned}$$

□

Als direkte Konsequenz aus dem vorausgegangenen Hilfssatz überlegt man sich nun leicht, dass die Bewertungen $B(x', H'g')$ und $B(\Phi(x), \Psi(H'g'))$ der Nichtblattknoten $H'g'$ und $\Psi(H'g')$ identisch sind. Dies führt nun zu dem folgenden Satz:

3.2.13 Satz. *Ist Φ ordnungserhaltend, so gilt für die Kanonisierungen CF_G^X und $\mathrm{CF}_{G'}^{X'}$, welche wir gemäß Satz 3.2.8 definieren:*

$$\Phi\Big(\mathrm{CF}_{G'}^{X'}(x')\Big) = \mathrm{CF}_G^X(\Phi(x')) \text{ für alle } x' \in X'.$$

[4] $g'y := \Psi(g')y$ für alle $y \in Y$, $g' \in G'$.

Beweis. Für ein Blatt $\{g'\}$ in $T(x', G')$ wird gemäß Vereinbarung der Bewertungsvektor noch um einen weiteren Eintrag $g'x'$ verlängert. Für das Bild $\{\Psi(g')\}$ des Knotens $\{g'\}$ wird entsprechend die Bewertung um $\Psi(g')\Phi(x') = \Phi(g'x')$ ergänzt.

Ist Φ aber ordnungserhaltend, so gilt für $g', \overline{g}' \in L(x', G')$

$$B(x', \{g'\}) \leq B(x', \{\overline{g}'\}) \iff B(\Phi(x'), \{\Psi(g')\}) \leq B(\Phi(x'), \{\Psi(\overline{g}')\}),$$

denn die Vektoren $B(x', \{g'\})$ und $B(\Phi(x'), \{\Psi(g')\})$ stimmen mit Ausnahme des letzten Eintrags überein. Den letzten Eintrag von $B(\Phi(x'), \{\Psi(g')\})$ erhalten wir aber durch Anwendung der ordnungserhaltenden Abbildung Φ. Somit ist $\Psi(L_0(x', G')) = L_0(\Phi(x'), G)$ und für jedes beliebige $g' \in L_0(x', G')$ gilt dann

$$\Phi\Big(\mathrm{CF}_{G'}^{X'}(x')\Big) = \Phi(g'x') = \Psi(g')\Phi(x') = \mathrm{CF}_{G}^{X}(\Phi(x')). \qquad \square$$

3.2.2. Ausnutzen bekannter Automorphismen

Neben der Bewertungsfunktion können wir auch bereits bekannte Automorphismen von x zum Abschneiden von Teilbäumen des Suchbaums $T(x, G)$ heranziehen. Wir nehmen hierzu an, dass eine Untergruppe $A \leq \mathrm{Stab}_G(x)$ gegeben sei. Weiter sei T_A eine fest gewählte Linkstransversale von G/A. Die Menge $L_0(x, G) \cap T_A$ ist nicht leer. Aus ihr lässt sich bereits die gesamte Information zur Kanonisierung von x analog zu Satz 3.2.8 gewinnen. Dabei wird der Stabilisator $\mathrm{Stab}_G(x)$ – wie sich leicht zeigen lässt – von A und $S'_G(x) := \{\mathrm{TR}_G(x)^{-1} g \mid g \in L_0(x, G) \cap T_A\}$ erzeugt. Damit können wir also jeden Teilbaum von $T(x, G)$ mit Wurzel Hg abschneiden, sobald wir feststellen können, dass $Hg \cap T_A = \emptyset$ gilt.

Aus diesem Grund bietet es sich auch an, den Suchbaum mittels einer Tiefensuche zu durchlaufen: In diesem Fall können wir stets die Gruppe A zu einer Obergruppe $A' := \langle A, g_0^{-1} g_1 \rangle$ vergrößern (und damit den Test verschärfen), sobald zwei Blattknoten $\{g_0\}$ und $\{g_1\}$ mit $g_0 x = g_1 x$ erreicht wurden. Wichtig ist nur, dass bei der Wahl der Transversalen $T_{A'}$ die Inklusionsbedingung $T_{A'} \subseteq T_A$ berücksichtigt wird. Nur so lässt sich garantieren, dass alle Knoten $L_0(x, G) \cap T_{A'}$ des Suchbaums besucht wurden. Weitere Details zu der Besuchsreihenfolge der Knoten des Baums $T(x, G)$ werden wir dann auch noch in Abschnitt 3.2.3 geben.

Eine hinreichende Bedingung für $Hg \cap T_A = \emptyset$ findet sich in [35, Hilfssatz 3.3.3]. Wir werden diese und die notwendigen Definitionen nun aufgreifen und verschärfen. Später wird sich zeigen, dass diese Verschärfung sogar zu einer hinreichenden und notwendigen Bedingung führen wird. Diese Beobachtung ist bereits im Rahmen meiner Arbeit [24, Lemma 5.9] erschienen. Dort wurde aber bereits zu Beginn davon ausgegangen, dass die operierende Gruppe G gleich der symmetrischen Gruppe S_n ist. Außerdem wurde nur sehr kurz umschrieben, in welcher Form das Lemma während des Backtracking zur Anwendung gebracht werden soll.

Um die Gruppe A der bekannten Automorphismen zu verwalten, führen wir zunächst *vollständige Labelled Branchings* nach M. Jerrum [40] ein. Hierzu wählen wir eine (kleinere) endliche Hilfsmenge [5] Z der Kardinalität $n := |Z|$, auf welcher G treu operiert. Über eine fest gewählte Anordnung $z_0 < \ldots < z_{n-1}$ der Elemente in $Z = \{z_0, \ldots, z_{n-1}\}$ definieren wir den Gruppenmonomorphismus

$$\begin{aligned}\Phi_{(z_0,\ldots,z_{n-1})} : G &\to S_n \\ g &\mapsto \overline{g} \quad \text{mit } \forall\, i,j \in [n] :\ \overline{g}(i) = j :\Longleftrightarrow g z_i = z_j,\end{aligned}$$

welcher eine Einbettung von G in die symmetrische Gruppe S_n ermöglicht. Den Vektor $(z_0, \ldots, z_{n-1})$ wollen wir als eine *Basis* der Gruppenoperation bezeichnen. Mit den Notationen $\overline{g} := \Phi_{(z_0,\ldots,z_{n-1})}(g)$ für $g \in G$ und $\overline{H} := \{\overline{g} \mid g \in H\}$ für jedes $H \leq G$ wollen wir uns auf diese fest gewählte Einbettung beziehen.

3.2.14 Beispiel.

- Die symmetrische Gruppe S_n operiert auf der Menge $[2]^{\binom{[n]}{2}}$ aller Graphen auf n Punkten. Gleichzeitig operiert sie auf natürliche Weise auch auf der Menge $Z = [n]$. Hierbei stellt die natürliche Anordnung $0 < 1 < \ldots < n-1$ nur eine der $n!$ Möglichkeiten zur Basiswahl dar.

- Die Gruppe $\mathrm{GL}_k(\mathbb{F}_q)$ operiert auf der Menge der $k \times n$ Matrizen über $\mathbb{F}_q$. Sie operiert aber auch auf natürliche Weise treu auf der kleineren Menge $Z = \mathbb{F}_q^k$ aller Spaltenvektoren.

Ist G eine Untergruppe von S_n, so werden wir im Folgenden mit

$$G^{(i)} := \mathrm{Stab}_G((0, \ldots, i-1)) = \mathrm{Stab}_G(0) \cap \ldots \cap \mathrm{Stab}_G(i-1)$$

den punktweisen Stabilisator der ersten $0 \leq i \leq n$ Elemente von $[n]$ bezeichnen. Jedem $\pi \in S_n$ können wir somit zwei Indizes $\mathrm{typ}(\pi) := (i, j)$ zuordnen:

3.2.15 Definition. Für $\mathrm{id}_n \neq \pi \in S_n$ sei das Paar $\mathrm{typ}(\pi) := (i, j) \in [n] \times [n]$ eindeutig bestimmt durch $\pi \in S_n^{(i)}$ und $j = \pi(i) \neq i$. Für die Identität setzen wir $\mathrm{typ}(\mathrm{id}_n) := (n-1, n-1)$.

Für ein $g \in G \leq S_n$ gibt die erste Komponente des Typs $\mathrm{typ}(g) = (i, j)$ genau den maximalen Index $i \in [n]$ zurück, für den noch $g \in G^{(i)}$ gilt. Die zweite Komponente unterscheidet die Element $g \in G^{(i)}$ nach ihrer Zugehörigkeit zu einer Linksnebenklasse von $G^{(i)}/G^{(i+1)}$.

[5] Die Hilfsmenge muss nicht notwendig in X liegen.

3.2.16 Definition (Labelled Branching)**.** Wir nennen ein Paar (B, σ) ein *Labelled Branching* zu $G \leq S_n$, falls

- $B = ([n], E)$ ein Branching ist, d.h. B ist ein gerichteter Graph ohne Schleifen[6], bei welchem in jedem Knoten höchstens eine Kante einmündet, und
- $\sigma : E \to G$ eine Kantenbeschriftung (= Label) definiert, für die gilt:
  - Ist $(i, j) \in E$, so ist $\sigma_{ij} := \sigma((i, j))$ vom Typ (i, j). Insbesondere ist also $i < j$.
  - Die Menge $\{\sigma_{ij} \mid (i, j) \in E\}$ aller Kantenbeschriftungen erzeugt G.

Wir setzen die Abbildung σ durch Produktbildung auf die Menge aller Pfade in B fort: Für einen Pfad $j_0, j_1, \ldots, j_k$ definieren wir $\sigma_{j_0,j_k} := \sigma_{j_{k-1},j_k} \cdots \sigma_{j_0,j_1}$. Diese Vorschrift ist wohldefiniert, da höchstens ein Pfad mit Startpunkt j_0 und Endpunkt j_k existieren kann[7]. Weiterhin definiert $\sigma_{j_0,j_k} \in G$ eine Gruppenelement vom Typ (j_0, j_k).

3.2.17 Definition (vollständiges Labelled Branching)**.** Das Labelled Branching (B, σ) heißt *vollständig*, falls zu jedem $i \in [n]$ die Menge

$$T^{(i)} := \{\sigma_{ij} \mid \text{ es gibt in } B \text{ einen Pfad von } i \text{ nach } j \in [n]\}$$

eine Linkstransversale von $G^{(i)}/G^{(i+1)}$ bildet.

Verschiedene bekannte Ansätze zur Berechnung eines vollständigen Labelled Branchings werden in [35] diskutiert. Die Kantenbeschriftungen $\sigma(E)$ bilden ein sogenanntes starkes Erzeugendensystem[8] der Stabilisatorkette

$$\{1_G\} = G^{(n-1)} \leq G^{(n-2)} \leq \ldots \leq G^{(1)} \leq G.$$

Diese Datenstruktur zur Verwaltung von G erlaubt es uns also, ein kurzes Erzeugendensystem mit höchstens $n-1$ Erzeugern von G anzugeben. Außerdem ermöglicht sie einen effizienten Test auf Enthalten sein in G und eine einfache Berechnung der Bahnen der Untergruppen $G^{(i)}$.

Die symmetrische Gruppe S_n ist totalgeordnet über den lexikographischen Vergleich der Listenschreibweisen

$$\pi \leq \sigma :\Longleftrightarrow [\pi(0), \ldots, \pi(n-1)] \leq [\sigma(0), \ldots, \sigma(n-1)],\ \forall\, \pi, \sigma \in S_n.$$

Der nachfolgende Satz ermöglicht es uns nun, über ein vollständiges Labelled Branching von $G \leq S_n$, die Elemente der Linkstransversalen $\{\pi \in S_n \mid \pi \leq \pi g, \forall\, g \in G\}$ aller minimalen Elemente der Linksnebenklassen S_n/G zu beschreiben.

[6] Eine Kante $(i, j) \in E$ nennt man Schleife, falls $i = j$ gilt.
[7] B ist ein sogenannter Wald, d.h. eine disjunkte Vereinigung von (gerichteten) Wurzelbäumen.
[8] Für alle $i \in [n]$ ist $\langle G^{(i)} \cap \sigma(E) \rangle = G^{(i)}$.

3.2.18 Fakt (Jerrum [40]). *Sei $G \leq S_n$ und (B, σ) ein vollständiges Labelled Branching zu G. Ein $\pi \in S_n$ ist genau dann minimal in der Linksnebenklasse πG, wenn für alle Pfade von i nach j in B gilt: $\pi(i) \leq \pi(j)$.*

3.2.19 Bemerkung. Die Permutationen $\pi \in S_n$, welche obige Pfadbedingung erfüllen, nennt man auch *topologische Anordnungen* zu B.

3.2.20 Folgerung. *Ist $A \leq G \leq S_n$ und (B, σ) ein vollständiges Labelled Branching zu A, so erhält man eine Transversale T_A der Linksnebenklassen G/A durch*

$$T_A := \{\pi \in G \mid \pi \leq \pi a, \forall\, a \in A\} = \{\pi \in G \mid \pi \text{ ist topologische Anordnung zu } B\}.$$

Für die Kanonisierung haben wir damit eine Linkstransversale $T_A := \{t \in G \mid \overline{t} \in T_{\overline{A}}\}$ von G/A über die Linkstransversale $T_{\overline{A}}$ von $\overline{G}/\overline{A}$ definiert. Mit deren Hilfe können wir nun leicht einen Test für die Bedingung $Hg \cap T_A = \emptyset$ angeben:

3.2.21 Fakt ([35], Hilfssatz 3.3.3). *Es seien $\pi \in S_n$ und $i \in [n]$ beliebig. Dann gibt es in der Rechtsnebenklasse $S_n^{(i)}\pi$ genau dann topologische Anordnungen zu einem vollständigen Labelled Branching $(([n], E), \sigma)$, wenn für alle Kanten $(j, k) \in E$ mit $\pi(k) < i$ gilt: $\pi(j) < \pi(k)$.*

Leider hängt die Güte dieses Tests bei der Anwendung im Suchbaum $T(x, G)$ stark von der getroffenen Wahl der (zunächst beliebigen) Basis, d.h. der Anordnung der Elemente in Z ab. Ist zum Beispiel für alle Nichtblattknoten Hg in $T(x, G)$ der Punkt z_0 kein Fixpunkt für die Gruppenoperation von H auf Z, so können wir den Test auch nicht zur Anwendung bringen. Wir bemerken aber, dass wir – über eine geeignete Implementierung des Backtracking – das Auftreten dieser Situation durch die folgenden Überlegungen verhindern können:

- In der ersten Variante schreibt man vor, dass die Partitionierung einer Nebenklasse Hg stets zu dem Stabilisator $H' := \mathrm{Stab}_H(z_i)$ des kleinsten Nichtfixpunktes $z_i \in Z \setminus \mathrm{Fix}_H(Z)$ zu erfolgen hat. Dieses Vorgehen hat zum einen den Nachteil, dass immer noch nicht alle Fixpunkte von H genutzt werden. Zum anderen kann es passieren, dass wir hierdurch unnötig große Verzweigungszahlen aufgezwungen bekommen, da der Index von H' in H groß werden kann. Dies versucht man (je nach der tatsächlichen Eingabemenge X) normalerweise aber gerade durch die Wahl einer möglichst großen Untergruppe $I(x, Hg) < H$ in der Partitionierung zu vermeiden.

- Die zweite Variante bildet ein sogenannter Basiswechsel, d.h. man wechselt zur Laufzeit des Backtrackalgorithmus die Anordnung der Menge Z bzw. die gewählte Einbettung $\Phi_{(z_0,\ldots,z_{n-1})}$ von G in S_n. Damit kann man stets erreichen, dass die Fixpunktmenge $\mathrm{Fix}_{\overline{H}}([n])$ zu dem aktuell betrachteten Knoten Hg eine Teilmenge $[i] \subseteq [n]$ bildet. Damit hat man die stärkstmögliche Einschränkung nach Hilfssatz 3.2.21 erreicht.

Ein Basiswechsel ist ganz offensichtlich zulässig, solange noch keine Knoten aus $L_0(x, G)$ in einem abgeschnittenen Teilbaum auftrat. Umgekehrt wird er gegebenenfalls notwendig, wenn bei der Tiefensuche ein Knoten Hg mit kleinerer Bewertung erreicht wird, da die Untergruppe $H \leq G$ möglicherweise bislang noch nicht als Stabilisator im Backtracking auftrat. Die Fixpunkte $\mathrm{Fix}_H(Z)$ können also wieder für beliebige Elemente aus Z angenommen werden.

Nachteilig an diesem Vorgehen ist der eventuell zu erwartende hohe Rechenaufwand für mehrfache Basiswechsel, welcher den zeitlichen Nutzen im Backtracking durchaus übersteigen kann.

Im Folgenden wollen wir nun Hilfssatz 3.2.21 verschärfen und damit eine dritte Variante entwickeln. Dies wird dazu führen, dass in den von uns untersuchten Gruppenoperationen auf obige Überlegungen verzichtet werden kann. Wir führen zunächst den Begriff einer Partition der Menge $[n]$ ein:

3.2.22 Definition (geordnete Partition)**.** Eine *geordnete Partition* der Menge $[n]$ sei eine Folge $\mathfrak{p} = (P_0, \ldots, P_{\ell-1})$ disjunkter, nicht leerer Teilmengen von $[n]$, deren Vereinigung ganz $[n]$ bildet. Indem wir den Elementen $i \in P_j$ die „Farbe“ j zuordnen, werden wir in diesem Zusammenhang auch von einer Färbung der Koordinaten sprechen. Im Folgenden sei mit $\mathcal{F}^{\mathfrak{p}} : [n] \to [\ell]$ die so definierte surjektive Funktion bezeichnet.

Falls die Reihenfolge der Blöcke keine Rolle spielt, so sprechen wir von *ungeordneten* Partitionen.

3.2.23 Definition. Ist $\mathfrak{p} = (P_0, \ldots, P_{\ell-1})$ eine Partition von $[n]$, so nennen wir ℓ die *Länge* von $\mathfrak{p}$ und schreiben hierfür kurz $|\mathfrak{p}|$. Die Teilmengen P_i werden wir *Blöcke* von $\mathfrak{p}$ nennen. Eine Partition der Länge n (d.h. alle Blöcke der Partition sind einelementig) nennen wir *diskret*.

3.2.24 Definition (kanonische Partition)**.** Eine Partition $\mathfrak{P}$ von $[n]$ mit $\mathcal{F}^{\mathfrak{P}}(i) \leq \mathcal{F}^{\mathfrak{P}}(j)$ für alle $0 \leq i < j \leq n-1$ nennen wir eine *kanonische Partition*.

Wir werden für kanonische Partitionen zur Verdeutlichung immer Großbuchstaben verwenden. Obige Definition ist äquivalent zu der Forderung, dass die Blöcke $P_i \in \mathfrak{P}$ der Partition $\mathfrak{P}$ geordnete Intervalle sind, d.h. $P_i = [n_{i+1}] \setminus [n_i]$ für eine aufsteigende Folge natürlicher Zahlen $0 = n_0 < n_1 < \ldots < n_{|\mathfrak{P}|} = n$.

3.2.25 Definition ((kanonische) Young-Untergruppe)**.** Es sei $\mathfrak{p}$ eine (kanonische) Partition von $[n]$. Die Untergruppe $S_{\mathfrak{p}} := \bigcap_{P \in \mathfrak{p}} \mathrm{Stab}_{S_n}(P)$ von S_n nennen wir *(kanonische) Young-Untergruppe* von S_n.

Damit können wir nun die angekündigte Verschärfung des Tests auf $Hg \cap T_A = \emptyset$ angeben:

3.2.26 Hilfssatz. *Es sei $\pi \in S_n$ und $\mathfrak{P} := (P_0, \ldots, P_{\ell-1})$ eine kanonische Partition zu $0 = n_0 < n_1 < \ldots < n_\ell = n$. Dann gibt es in der Rechtsnebenklasse $S_\mathfrak{P}\pi$ genau dann topologische Anordnungen zu einem vollständigen Labelled Branching $(([n], E), \sigma)$ von $A \leq S_n$, wenn aus $(i, j) \in E$ bereits $\mathcal{F}^\mathfrak{P}(\pi(i)) \leq \mathcal{F}^\mathfrak{P}(\pi(j))$ folgt.*

Beweis. Zu $i \in [n]$ bezeichne $i_0 := \mathcal{F}^\mathfrak{P}(\pi(i))$ die Farbe des Bilds $\pi(i)$.
Wir nehmen zunächst an, dass $\pi \in S_n$ die Implikation nicht erfüllt, es also eine Kante $(i, j) \in E$ gibt mit $i_0 = \mathcal{F}^\mathfrak{P}(\pi(i)) > \mathcal{F}^\mathfrak{P}(\pi(j)) = j_0$. Für eine beliebige Permutation $\sigma \in S_\mathfrak{P}$ liegen die Bilder $\sigma\pi(i) \in P_{i_0}$ bzw. $\sigma\pi(j) \in P_{j_0}$ aber in den gleichen Mengen wie $\pi(i)$ bzw. $\pi(j)$. Damit ist aber wegen $\sigma\pi(j) \leq n_{j_0+1} - 1 < n_{i_0} \leq \sigma\pi(i)$ die Permutation $\sigma\pi$ ebenfalls keine topologische Anordnung zu $([n], E)$.

Für die Rückrichtung geben wir eine Permutation $\sigma \in S_\mathfrak{P}$ an, so dass $\sigma\pi$ eine topologische Anordnung zu $([n], E)$ ist. Wir wählen hierzu $\sigma \in S_\mathfrak{P}$ derart, dass in jedem Block $P_{\ell'} = [n_{\ell'+1}] \setminus [n_{\ell'}]$, $\ell' \in [\ell]$ die Urbilder von $\sigma\pi$ lexikographisch angeordnet sind:

$$\pi^{-1}\sigma^{-1}(n_{\ell'}) < \ldots < \pi^{-1}\sigma^{-1}(n_{\ell'+1} - 1).$$

Eine solche Permutation existiert, da wir – mit Hilfe der Gruppenoperation von S_n auf $[n]^n$ – mit der Gruppe $S_\mathfrak{P}$ den Vektor

$$(\pi^{-1}(0), \ldots, \pi^{-1}(n-1)) = (0, \ldots, n-1) \cdot P^{(\pi^{-1})}$$

blockweise lexikographisch sortieren können. Die Einschränkung der Abbildung $(\sigma\pi)^{-1}$ auf einen Block $P \in \mathfrak{P}$ ist also ordnungserhaltend. Für eine beliebige Kante $(i, j) \in E$ gilt nun entweder

- $i_0 < j_0$ und damit ohnehin $\sigma\pi(i) \leq n_{i_0+1} - 1 < n_{j_0} \leq \sigma\pi(j)$,
- oder es ist $i_0 = j_0$ und damit nach Konstruktion $\sigma\pi(i) < \sigma\pi(j)$, da für die Urbilder $i, j \in (\sigma\pi)^{-1}(P_{i_0})$ die Relation $i < j$ gilt. □

Anhand der Bahnen der Gruppe $\overline{H}$ auf $[n]$ für einen Knoten Hg in $T(x, G)$ können wir eine minimale kanonische Young-Untergruppe $\overline{H} \leq S_\mathfrak{P}$ für diesen Test bestimmen. Wieder lässt sich durch einen Basiswechsel erreichen, dass die Bahnen von $\overline{H}$ Intervalle bilden. Somit steht die vollständige Information aus dem Hilfssatz zur Verfügung. Wir werden aber im Abschnitt 3.2.4 aufzeigen, dass dies in den von uns betrachteten Fällen bei geeigneter Wahl der Homomorphismen f_H und der Partitionierungsvorschrift I nicht notwendig ist, da alle im Backtracking auftretenden Gruppen $H \leq S_n$ stets kanonische Young-Untergruppen sind.

3.2.3. Implementierungsdetails

Der Algorithmus aus Satz 3.2.8 macht noch keine Aussage zum Aufbau des Suchbaums $T(x, G)$. Dies möchten wir – da nun alle erforderlichen Hilfsmittel zur Verfügung stehen – an dieser Stelle nachholen.

- Im Rahmen einer Breitensuche kann man über die Bewertungsfunktion B leicht Teilbäume identifizieren, die nicht zu Elementen $g \in L_0(x, G)$ führen. Da wir Automorphismen aber nur über besuchte Blattknoten auffinden werden, bedeutet dies auch, dass wir diese erst im letzten Schritt erreichen werden. Somit kann das Abschneiden nach Hilfssatz 3.2.26 nur sehr eingeschränkt genutzt werden.

- Umgekehrt können wir im Rahmen einer Tiefensuche den aktuellen Knoten $H'g'$ der Tiefe i – wegen der nicht vollständigen Information über die auftretenden Bewertungen – nur dann abschneiden, falls ein anderer, *bereits besuchter* Knoten Hg der gleichen Tiefe i mit kleinerer Bewertung $B(x, Hg) < B(x, H'g')$ existiert. Man wird also auch Teilbäume untersuchen, deren Blätter nicht in der Menge $L_0(x, G)$ liegen. Das zuletzt besuchte Blatt liefert während der Laufzeit immer nur einen Kandidaten für das Transporterelement.

  Jedoch definieren gleich bewertete Blätter immer einen Automorphismus von x, unabhängig davon ob sie den kanonischen Repräsentanten definieren. Eine stetig wachsende Untergruppe $A \leq \mathrm{Stab}_G(x)$ des Stabilisators von x steht also bereits frühzeitig zur Verfügung, um das Abschneidekriterium aus Hilfssatz 3.2.26 anzuwenden. Außerdem muss im Rahmen einer Tiefensuche immer nur die lokale Information abgespeichert werden, d.h. im wesentlichen die Zustände der Vorfahren zum aktuellen Knoten Hg.

  Es ist leicht ersichtlich, dass das Laufzeitverhalten eines solchen Algorithmus auch entscheidend von der Tatsache abhängig ist, wann zum ersten mal ein Blatt aus der Menge $L_0(x, G)$ erreicht wird. Daher kann es bei verschiedenen isomorphen Eingaben x und $x' = g_0 x$ zu erheblichen Laufzeitunterschieden kommen.

- Eine viel versprechende Kombination beider Ansätze im Bereich der Graphenkanonisierung wird gegenwärtig in [56] diskutiert. Der Baum wird in Breitensuche durchlaufen. Dabei wird jedem auftretenden Knoten Hg ein zufällig gewählter Blattknoten $\{g'\}$ mit $g' \in Hg$ zugeordnet. Die Autoren nennen diese Zuordnung einen *experimentellen Pfad* zu einem Blattknoten. Nach dem Abschneiden mittels der Bewertungsfunktion werden zusätzlich noch die Bahnenelemente $g'x$ zu den experimentellen Pfaden gebildet und verglichen. Bei Gleichheit kann der so gefundene Automorphismus zu den Erzeugern der Gruppe A hinzugefügt werden.

- Schließlich möchten wir noch auf das sogenannte Iterative Deepening hinweisen, welches den zu erwartenden hohen Speicherbedarf einer Breitensuche mit dem Einsatz eines höheren Rechenaufwands versucht zu vermeiden. R. Gugisch [35, Seite 87] schreibt hierzu:

  > *„Als Lösung bietet sich an, den Baum iterativ mehrmals zu durchlaufen, und in jedem Durchlauf eine Ebene weiter vorzudringen. In Durchlauf i werden also nur die Knoten bis zur Ebene i besucht und das Optimum [...] bestimmt. So kann man bei den darauf folgenden Durchläufen alle auf Ebene i nicht optimalen Knoten überspringen.*

Dieses als Iterative Deepening bekannte Vorgehen benötigt erstaunlicherweise kaum mehr theoretischen Aufwand als eine Breitensuche.“

Bei R. Gugisch werden die Bahnenelemente gx der Nebenklassenvertreter g aller Knoten Hg der Tiefe i direkt miteinander verglichen[9]. Damit reduziert sich der Rechenaufwand – aber auch die Erfolgsaussichten zur Generierung von Automorphismen – natürlich erheblich im Vergleich zu dem Verfahren experimentelle Pfade analog zu [56] zu bestimmen. Eine Kombination beider Ansätze stellt somit eine weitere sehr interessante Strategie dar.

Im Weiteren werden wir den Suchbaum immer über eine Tiefensuche aufbauen, da sich eine klassische Breitensuche wegen des benötigten Speicherbedarfs unpraktikabel erweist. Ebenso erschien uns das Iterative Deepening als nicht geeignet, da sich der Aufwand zur Bestimmung der Kinder in unserem Fall wesentlich rechenintensiver als bei [35] gestaltet und außerdem wie bereits besprochen, die Frage nach der Bestimmung der Automorphismen unzureichend beantwortet wurde. Eine Untersuchung der Strategie nach [56] konnte leider aus zeitlichen Gründen nicht mehr umgesetzt werden.

3.2.27 Bemerkung. Möchte man nur die Automorphismengruppe von x bestimmen, so genügt es, die Bewertung des zuerst erreichten Blattknotens $\{g_0\}$ heranzuziehen. Man schneidet dann im Backtrackdurchlauf einen Teilbaum mit Wurzelknoten Hg auf Tiefe $i > 0$ ab, falls $B(x, \{g_0\})_i \neq B(x, Hg)_i$ gilt.

3.2.28 Bemerkung. Will man zwei Elemente $x, x' \in X$ ausschließlich auf Isomorphie testen, so kann man die Bewertung eines beliebigen Blattknotens $\{g_0\}$ in $T(x, G)$ als Abschneidekriterium für die Erzeugung von $T(x', G)$ heranziehen. Man bricht das Backtracking an den Knoten Hg von $T(x', G)$ auf Tiefe $i \geq 0$ ab, für welche die Bewertungen $B(x', Hg)_i$ nicht mit den entsprechenden Werten $B(x, \{g_0\})_i$ übereinstimmen. Erreicht man ein Blatt $\{g'\}$, so gilt $g'x' = g_0x$ und man kann den Algorithmus mit positiver Antwort sofort beenden. Dringt der Algorithmus nicht bis zu den Blättern von $T(x', G)$ vor, so liegen die Elemente x, x' in verschiedenen Bahnen $Gx \neq Gx'$.

3.2.4. Spezialfall: Die Kanonisierung von Graphen

Wir wollen nun als Beispiel die Kanonisierung eines Graphen $\Gamma = ([n], E)$ unter der Operation der symmetrischen Gruppe S_n untersuchen. Insbesondere wollen wir hierbei aufzeigen, dass sich die in der Literatur [41], [55], [56], u.v.m. entworfenen Algorithmen zur Operation der symmetrischen Gruppe S_n als Spezialfall des hier dargestellten Algorithmus beschreiben lassen.

[9] Wir wollen jedoch darauf hinweisen, dass die an dieser Stelle gemachte Aussage [35, Seite 87]

„$g_{\text{opt}}x \sim_{i-1} g_i$ *mit* $a := g_{\text{opt}}^{-1} g_i \in G_{(x_i,\ldots,x_{n-1})}$, *so ist* $a \in G_x$“

im Allgemeinen inkorrekt ist.

Zunächst stellen wir den Zusammenhang zwischen geordneten Partitionen und Nebenklassen her, da die oben genannten Implementierungen einen Suchbaum aufbauen, dessen Knoten mit geordneten Partitionen beschrieben werden. Zu einer geordnete Partition $\mathfrak{p} = (p_0, \ldots, p_{\ell-1})$ von $[n]$ sei nun immer $\mathfrak{P} = (P_0, \ldots, P_{\ell-1})$ diejenige kanonische Partition von $[n]$ für die $|p_i| = |P_i|,\ \forall\, i \in [\ell]$ gilt. Die geordnete Partition $\mathfrak{p}$ repräsentiert in den oben genannten Algorithmen die Menge aller Permutationen

$$S_n^{\mathfrak{p}} := \{\pi \in S_n \mid \pi(p_i) = P_i, \forall\, i \in [\ell]\}.$$

Sie lässt sich auch über die Nebenklasse $S_n^{\mathfrak{p}} = S_{\mathfrak{P}}\pi \in \mathcal{C}(S_n)$ mit $\pi \in S_n^{\mathfrak{p}}$ ausdrücken. Somit ist also gezeigt, dass die geordnete Partition $\mathfrak{p}$ nur eine weitere Datenstruktur zur Verwaltung der Nebenklassen $S_{\mathfrak{P}}\pi$ darstellt.

Auf der Menge aller Partitionen von $[n]$ lässt sich wie folgt eine Halbordnung definieren:

$$\mathfrak{p} \preceq \mathfrak{q} :\Longleftrightarrow (\forall\, i, j \in [n] : \mathcal{F}^{\mathfrak{q}}(i) < \mathcal{F}^{\mathfrak{q}}(j) \Longrightarrow \mathcal{F}^{\mathfrak{p}}(i) < \mathcal{F}^{\mathfrak{p}}(j))$$

Sind $\mathfrak{p}$ und $\mathfrak{q}$ Partitionen von $[n]$ und $\mathfrak{p} \prec \mathfrak{q}$, dann sagen wir $\mathfrak{p}$ ist *feiner* als $\mathfrak{q}$ oder $\mathfrak{p}$ ist eine *Verfeinerung* von $\mathfrak{q}$. Eine Verfeinerung $\mathfrak{p} \prec \mathfrak{q}$ impliziert, dass jeder Block von $\mathfrak{p}$ in einem Block von $\mathfrak{q}$ enthalten sein muss. Diese Beobachtung ist aber noch nicht hinreichend, es muss zusätzlich auch noch die Anordnung der Blöcke in $\mathfrak{q}$ berücksichtigt werden. Dies erklärt auch die Wahl des Begriffs „Verfeinerung“ für die in Gleichung (3.2) definierte Basisoperation.

Die Partitionierung einer Nebenklasse $S_n^{\mathfrak{p}} = S_{\mathfrak{P}}\pi$ wird in diesen Algorithmen stets über eine sogenannte *Individualisierung* erreicht. Dazu wählt man einen Block $p_i \in \mathfrak{p}$ mit $|p_i| > 1$ und definiert das Resultat eines Partitionierungsschritts über die Menge

$$\{\,(p_0, \ldots, p_{i-1}, \{p\}, p_i \setminus \{p\}, p_{i+1}, \ldots, p_{\ell-1}) \mid p \in p_i\},$$

welche alle Möglichkeiten darstellt einen Punkt $p \in p_i$ von dem Block p_i abzutrennen.

Der gewählte Index $i \in [\ell]$ muss, um die Bedingungen an eine Partitionierung zu erfüllen, wieder für isomorphe Eingaben $(\Gamma, S_n^{\mathfrak{p}}), (\pi_0\Gamma, S_n^{\mathfrak{p}}\pi_0^{-1})$ identisch sein. Man wählt zum Beispiel immer den kleinsten (bzgl. Kardinalität) nicht trivialen Block p_i mit niedrigstem Index i. In der Sprache aus Gleichung (3.1) führt dies dann zur Definition der Untergruppe $I(\Gamma, S_{\mathfrak{P}}\pi) := \mathrm{Stab}_{S_{\mathfrak{P}}}(\min(P_i)) < S_{\mathfrak{P}}$.

Wir wollen noch etwas näher auf die Verfeinerung eingehen. Im Rahmen der Graphenkanonisierung wird diese zum Beispiel über das Zählen der Nachbarn nach der Knotenfarbe erreicht, also durch Anwendung der $S_{\mathfrak{P}}$-Homomorphismen

$$\begin{aligned} N_{\mathfrak{P}} : [2]^{\binom{[n]}{2}} &\to \left(\mathbb{Z}^{|\mathfrak{P}|}\right)^n \\ E &\mapsto \left(\left(|\{j \in P_k \mid \{i, j\} \in E\}|\right)_{k \in [|\mathfrak{P}|]}\right)_{i \in [n]}. \end{aligned}$$

Hier zeigt sich, dass es unter Umständen notwendig ist, das Homomorphieprinzip iteriert anzuwenden. Man betrachte dazu folgendes Beispiel:

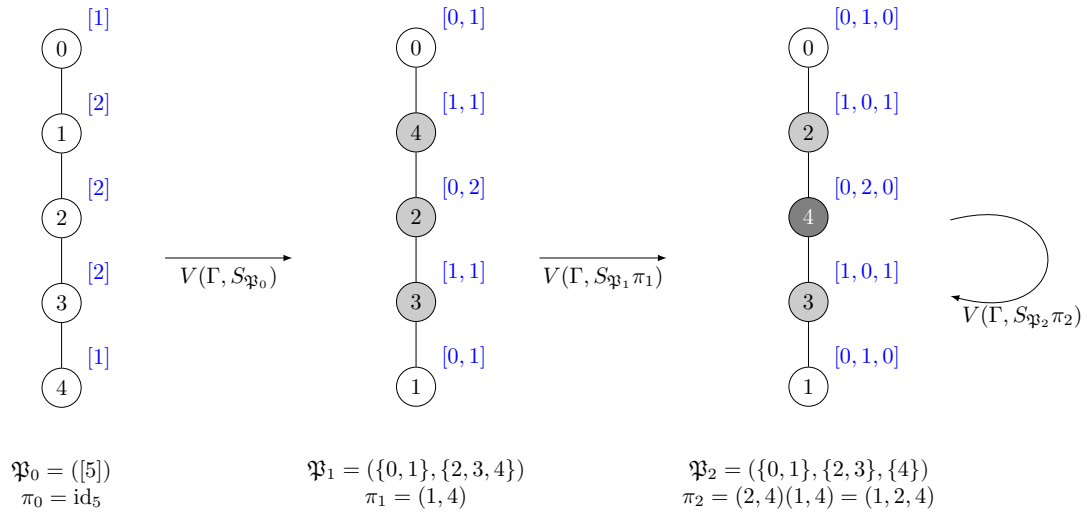
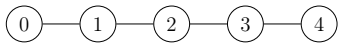


Abbildung 3.5.: Iterierte Verfeinerung

3.2.29 Beispiel. Es soll der folgende Graph $\Gamma =$  kanonisiert werden.

- Die Wurzel des Suchbaums $T(\Gamma, S_5)$ wird über die Verfeinerung $V(\Gamma, S_5)$ definiert und diese wiederum mittels Homomorphieprinzip aus dem S_5-Homomorphismus $N_{([5])}$. Das Bild $N_{([5])}(\Gamma) = ((1),(2),(2),(2),(1))$ ordnet jedem Knoten die Anzahl seiner Nachbarn zu, siehe auch Abbildung 3.5 (links).

  Die Kanonisierung des Vektors $N_{([5])}(\Gamma) = ((1),(2),(2),(2),(1))$ unter der Operation der symmetrischen Gruppe S_5 erfolgt nun durch lexikographisches Sortieren, etwa über die Permutation $(1,4)$. Im Anschluss werden nur noch Permutationen aus dem Stabilisator

  $$\mathrm{Stab}_{S_5}\Big(\underbrace{N_{([5])}((1,4)\Gamma)}_{=((1),(1),(2),(2),(2))}\Big) = S_{(\{0,1\},\{2,3,4\})}$$

  auf $(1,4)\Gamma$ angewandt. Wir erhalten somit als Wurzel des Backtrackbaums $T(\Gamma, S_5)$ die Nebenklasse $V(\Gamma, S_5) = S_{(\{0,1\},\{2,3,4\})}(1,4)$.

- An dieser Stelle würde man nun mit der Partitionierung von $S_{\mathfrak{P}_1}(1,4)$ fortfahren. Man beobachtet aber, siehe Abbildung 3.5 (Mitte), dass es durch die vorhergehende Verfeinerung möglich ist, im Graphen $(1,4)\Gamma$ den Knoten (2) von (3) und (4) zu separieren. Dies erfolgt durch eine erneute Anwendung des Homomorphieprinzips zum Homomorphismus $N_{(\{0,1\},\{2,3,4\})}$.

- Erst durch diesen Schritt erreicht man eine Färbung der Knoten bzw. eine Nebenklasse $S_{\mathfrak{P}_2}(1,2,4)$, welche unter Anwendung der Verfeinerung V invariant bleibt.

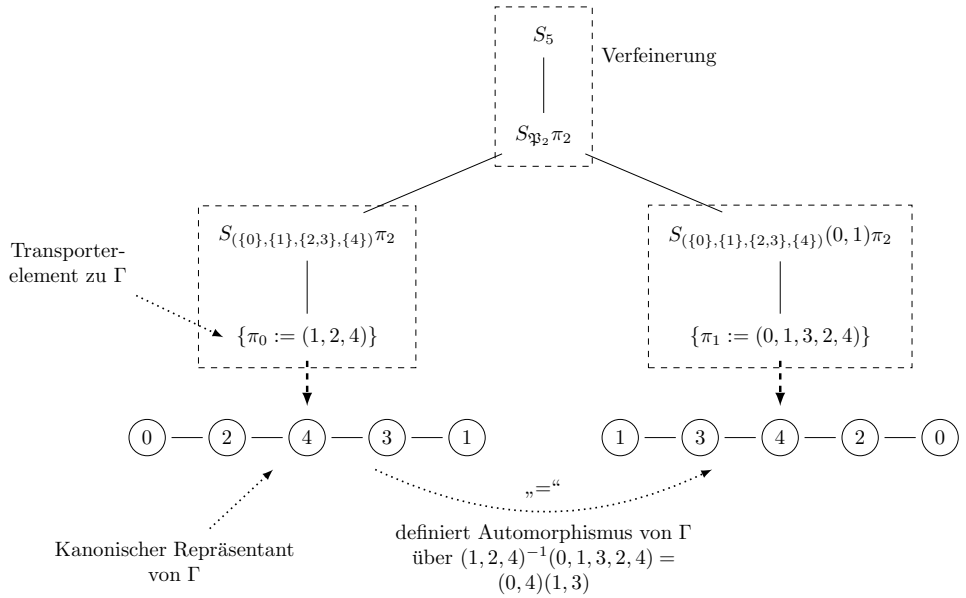


Abbildung 3.6.: Suchbaum zu Beispiel 3.2.29

In der Graphenkanonisierung wird eine solche Partitionierung als balanciert (engl. „*equitable*“) bezeichnet.

Abbildung 3.6 zeigt nun den gesamten Suchbaum $T(\Gamma, S_5)$. Wir haben zur Verdeutlichung die Zwischenschritte der Verfeinerung ebenfalls notiert. Die gestrichelten Bereiche zeigen Urbild und Bild einer jeden (iterierten) Verfeinerung. Nur die Bilder treten tatsächliche im Suchbaum $T(\Gamma, S_5)$ als Knoten auf.

3.2.30 Bemerkung. Sollen bereits gefärbte Graphen als die zu kanonisierenden Objekte untersucht werden, so transformiert man das Problem zur Kanonisierung eines gegebenen Graphen $\Gamma = ([n], E)$, welcher über eine geordnete Partition $\mathfrak{p}$ gefärbt ist, auf eine Kanonisierung von $\pi\Gamma$ unter $S_{\mathfrak{P}}$ mit $S_n^{\mathfrak{p}} = S_{\mathfrak{P}}\pi$. Dann treten im Backtracking ebenfalls nur kanonische Young-Untergruppen auf.

3.2.5. Iterierte Verfeinerung

Das Beispiel 3.2.29 deutet an, dass die tatsächlich genutzte Verfeinerungsfunktion V auch über eine mehrfache Anwendung einer weiteren Verfeinerung $\overline{V} : X \times \mathcal{C}(G) \to \mathcal{C}(G)$ gewonnen werden kann. Wir definieren hierzu:

$V_0(x, Hg) := \overline{V}(x, Hg)$

und für beliebiges $n \geq 1$

$V_n(x, Hg) := \overline{V}(x, V_{n-1}(x, Hg))$.

3.2.31 Hilfssatz. *Es sei $n \geq 1$ beliebig. Dann definiert die Funktion V_n ebenfalls eine Verfeinerung. Außerdem existiert eine natürliche Zahl $N \geq 1$, so dass für alle $n' \geq N$ die Gleichheit $V_{n'} = V_N$ gilt.*

Beweis. Es seien $g_0 \in G$, $x \in X$ und $Hg \in \mathcal{C}(G)$ beliebig. Dann ergibt sich die erste Aussage induktiv aus der Gleichung

$$\begin{aligned} g_0 * V_n(x, Hg) &= g_0 * \overline{V}(x, V_{n-1}(x, Hg)) = \overline{V}(g_0 x, g * V_{n-1}(x, Hg)) \\ &= \overline{V}\big(g_0 x, V_{n-1}(g_0 x, Hgg_0^{-1})\big) = V_n(g_0 x, Hgg_0^{-1}) = V_n(g_0 * (x, Hg)), \end{aligned}$$

sowie der Tatsache, dass $V_n(x, Hg) = \overline{V}(x, V_{n-1}(x, Hg)) \subseteq V_{n-1}(x, Hg) \subseteq Hg$ gilt. Da die Menge aller Rechtsnebenklassen endlich ist, muss die absteigende Kette

$$V_0(x, Hg) \supseteq V_1(x, Hg) \supseteq \dots$$

von Rechtsnebenklassen ab einer natürlichen Zahl $N(x, Hg)$ stationär werden, d.h.

$$V_0(x, Hg) \supsetneq V_1(x, Hg) \supsetneq \dots \supsetneq V_{N(x,Hg)}(x, Hg) = V_{N(x,Hg)+1}(x, Hg) = \dots \ .$$

Da auch der Definitionsbereich $X \times \mathcal{C}(G)$ endlich ist, erfüllt

$$N := \max_{(x', H'g') \in X \times \mathcal{C}(G)} N(x', H'g') \in \mathbb{N}$$

die Aussage aus dem zweiten Punkt. □

3.2.32 Bemerkung. Wir nennen dieses Prinzip zur Gewinnung weiterer Verfeinerungen daher auch *iterierte Verfeinerung*. Der Begriff einer balancierten Partition zu einem gefärbten Graphen aus der Graphenkanonisierung beschreibt also genau das Erreichen der Zahl $N(x, Hg)$.

Für die Korrektheit des in Satz 3.2.8 entworfenen Kanonisierungsalgorithmus ist es unerheblich ob $V_n(x, Hg) = V_{n-1}(x, Hg)$ gilt, beziehungsweise ob die Verfeinerungen noch zu echten Untergruppen geführt haben. Wir gehen daher zur Vereinfachung im Folgenden von einer fest gewählten, eingabeunabhängigen Zahl $v \in \mathbb{N}$ aus und betrachten die Verfeinerung V_v.

Wir nehmen weiter an, dass die ursprüngliche Verfeinerung $\overline{V}$ über die Anwendung des Homomorphieprinzips aus einer Familie $(\overline{f}_H)_{H \in \mathcal{L}(G)}$ von H-Homomorphismen $\overline{f}_H$ gewonnen wurde, siehe Satz 3.2.4. Das Ziel dieses Abschnittes ist es, nun zu beweisen, dass sich auch V_v über eine Familie $(f_H)_{H \in \mathcal{L}(G)}$ gewinnen lässt. Dies ist notwendig um weiterhin die Bewertungen $B(x, Hg)$ auf den Knoten Hg des Backtrackbaums $T(x, G)$ definieren zu können.

Wir wollen diese Aussage induktiv beweisen. Daher nehmen wir zunächst an, dass für alle Untergruppen H von G zwei H-Homomorphismen $f_H^{(0)} : X \to Y$ und $f_H^{(1)} : X \to Z$ gegeben seien.

3.2.33 Hilfssatz. *Für alle $H \in \mathcal{L}(G)$ definiert die Funktion*

$$\begin{aligned} f_H : X &\to Y \times Z \\ x &\mapsto \left(f_H^{(0)}(x),\ \left(\mathrm{TR}_H^Y\left(f_H^{(0)}(x)\right)\right)^{-1} \cdot f^{(1)}_{\mathrm{Stab}_H\left(\mathrm{CF}_H^Y(f_H^{(0)}(x))\right)}\left(\mathrm{TR}_H^Y\left(f_H^{(0)}(x)\right)\cdot x\right)\right) \end{aligned}$$

einen H-Homomorphismus.

Beweis. Es seien $h \in H \in \mathcal{L}(G)$ und $x \in X$ beliebig. Zunächst beweist man leicht über die Kanonisierung von $f_H^{(0)}(x)$ und $f_H^{(0)}(hx)$, dass es ein $h' \in H' := \mathrm{Stab}_H\left(\mathrm{CF}_H^Y(f_H^{(0)}(x))\right)$ gibt mit

$$\underbrace{\mathrm{TR}_H^Y\left(f_H^{(0)}(hx)\right)}_{=:h_1} h = h' \underbrace{\mathrm{TR}_H^Y\left(f_H^{(0)}(x)\right)}_{=:h_0}.$$

Damit zeigen wir:

$$\begin{aligned} h^{-1} \cdot f_H(hx) &= h^{-1} \cdot \left(f_H^{(0)}(hx),\ h_1^{-1} \cdot f_{H'}^{(1)}(h_1 \cdot hx)\right) \\ &= \left(f_H^{(0)}(x),\ h^{-1}h_1^{-1} \cdot f_{H'}^{(1)}(h_1 hx)\right) = \left(f_H^{(0)}(x),\ h_0^{-1}h'^{-1} \cdot f_{H'}^{(1)}(h'h_0 x)\right) \\ &= \left(f_H^{(0)}(x),\ h_0^{-1} \cdot f_{H'}^{(1)}(h_0 \cdot x)\right) = f_H(x) \end{aligned}$$

□

Für die Untergruppen $H \in \mathcal{L}(G)$ seien Can_H^Y bzw. Can_H^Z Kanonisierer für die Operationen von H auf den Mengen Y und Z. Mit diesen gewinnen wir die Verfeinerungen $V^{(0)}$ und $V^{(1)}$ gemäß Satz 3.2.4.

Außerdem können wir nach Beispiel 3.1.5 einen Kanonisierer $\mathrm{Can}_H^{X\times Y}$ für das direkte Produkt $Y \times Z$ definieren, indem wir zunächst die erste Komponente auf ihren kanonischen Repräsentanten transformieren und anschließend nur noch mit dem Stabilisator auf der zweiten Komponenten operieren. Es definiert also auch die Familie $(f_H)_{H\in\mathcal{L}(G)}$ zusammen mit den Kanonisierern $\left(\mathrm{Can}_H^{X\times Y}\right)_{H\in\mathcal{L}(G)}$ nach Satz 3.2.4 eine Verfeinerung V. Wir zeigen nun, dass diese als iterierte Verfeinerung gesehen werden kann:

3.2.34 Hilfssatz. *Für alle $(x, Hg) \in X \times \mathcal{C}(G)$ gilt $V(x, Hg) = V^{(1)}\big(x, V^{(0)}(x, Hg)\big)$.*

Beweis. Wir zeigen für ein beliebiges Paar $(x, Hg) \in X \times \mathcal{C}(G)$, dass

$$V^{(1)}\big(x, V^{(0)}(x, Hg)\big) = \mathrm{Stab}_H\big(\mathrm{CF}_H^{X\times Y}(f_H(gx))\big)\,\mathrm{TR}_H^{X\times Y}(f_H(gx))g$$

gilt. Hierzu sei $H' = \mathrm{Stab}_H\left(\mathrm{CF}_H^Y\left(f_H^{(0)}(gx)\right)\right)$ und $h' := \mathrm{TR}_H^Y\left(f_H^{(0)}(gx)\right)$. Es ist also $V^{(0)}(x, Hg) = H'h'g$ und wir erhalten:

$$\begin{aligned} V^{(1)}(x, H'h'g) &= \mathrm{Stab}_{H'}\left(\mathrm{CF}_{H'}^Z\left(f_{H'}^{(1)}(h'gx)\right)\right) \cdot \mathrm{TR}_{H'}^Z\left(f_{H'}^{(1)}(h'gx)\right) \cdot h'g \\ &= \mathrm{Stab}_H\big(\mathrm{CF}_H^{X\times Y}(f_H(gx))\big) \cdot \mathrm{TR}_H^{X\times Y}(f_H(gx)) \cdot g = V(x, Hg) \end{aligned}$$

□

Nun sei $\overline{V}$ die aus $\left(\overline{f}_H, \overline{\mathrm{Can}}_H\right)_{H\in\mathcal{L}(G)}$ gewonnene Verfeinerung, welche wir iteriert – wie zu Beginn des Abschnitts angedeutet – anwenden wollen. Obiger Hilfssatz erlaubt es uns nun induktiv zu jeder Verfeinerung V_i, $i \in [v+1]$ eine geeignete Familie $(f_H^{(i)})_{H\in\mathcal{L}(G)}$ von H-Homomorphismen $f_H^{(i)}$ und zugehörigen Kanonisierern $\mathrm{Can}_H^{(i)}$ anzugeben. Zum Induktionsstart können wir offensichtlich

$$\left(f_H^{(0)}, \mathrm{Can}_H^{(0)}\right)_{H\in\mathcal{L}(G)} := \left(\overline{f}_H, \overline{\mathrm{Can}}_H\right)_{H\in\mathcal{L}(G)}$$

setzen. Aus der Induktionsvoraussetzung folgt, dass es zu der Verfeinerung V_{i-1} eine Familie von H-Homomorphismen samt Kanonisierung im Bildbereich gibt, welche V_{i-1} definiert. Damit sind die Voraussetzungen für den Hilfssatz 3.2.34 erfüllt (mit $V^{(1)} = \overline{V}$ und $V^{(0)} = V_{i-1}$) und wir schließen, dass sich V_i ebenso definieren lässt.

Dieser Beweis zeigt noch mehr. Es ist also möglich die Verfeinerung V für das Backtracking über eine Folge beliebiger Verfeinerungen $(\overline{V}_0, \ldots, \overline{V}_{v-1})$ zu definieren. Wir schreiben aus diesem Grund eine Folge von Verfeinerungen $(\overline{V}_0, \ldots, \overline{V}_{v-1})$ bzw. für jedes $i \in [v]$ eine Familie $(f_H^{(i)})_{H\in\mathcal{L}(G)}$ samt Kanonisierern im Bildbereich zur Gewinnung der iterativen Verfeinerung V vor.

3.2.35 Bemerkung. Man kann sich auch überlegen, dass die Wahl der Verfeinerung $\overline{V}_i, i \in [v]$ hierbei durchaus von den Resultaten der Verfeinerungen auf den Vorfahren abhängig gemacht werden kann. Wir gehen hierauf aber nicht mehr näher ein.

Um im Folgenden leichter auf die Eingaben und Ausgaben für den iterierten Aufruf der Verfeinerungen zugreifen zu können, werden wir mit $\overline{T}(x, G)$ denjenigen Wurzelbaum bezeichnen, der auch alle Zwischenschritte des Backtracking über eine iterierte Verfeinerung wiedergibt. Die Knotenmenge ist hierbei über eine Teilmenge von $\mathcal{C}(G) \times [v+1]$ gegeben. Die zusätzliche Bereitstellung des Iterationszählers für die Knotenbeschriftungen verhindert, dass Schleifen in $\overline{T}(x, G)$ auftreten. Den Wurzelbaum definieren wir wieder induktiv:

- Die Wurzel von $\overline{T}(x, G)$ sei $(G, 0)$.
- Ist (Hg, i) ein Knoten in $\overline{T}(x, G)$ mit $i < v$ so definiert $(\overline{V}_i(x, Hg), i+1)$ das einzige Kind von (Hg, i).
- Für einen Knoten (Hg, v) mit $|H| > 1$ sei $H' := I(x, Hg)$ und $\{h_0, \ldots, h_{u-1}\}$ eine Rechtstransversale von H' in H. Die Kinder von (Hg, v) werden dann von der Menge $\{H'h_0g, \ldots, H'h_{u-1}g\}$ gebildet.

Die Bewertungsfunktion verallgemeinern wir wie oben: Die Knoten der Gestalt $(Hg, 0)$ mit $Hg \neq G$ wurden durch eine Partitionierung erzeugt, sie übernehmen daher ihre Bewertung vom Vater. Für alle anderen Knoten (Hg, i), $i \geq 1$ verlängern wir den Bewertungsvektor des Vaters $(H'g, i-1)$ um den Eintrag $\mathrm{CF}_{H'}(f_{H'}^{(i-1)}(gx))$. Wir können somit alle zuvor bewiesenen Aussagen für $T(x, G)$ sofort auf $\overline{T}(x, G)$ übertragen, insbesondere Fakt 3.2.2 und auch das Abschneiden von Teilbäumen über bekannte Automorphismen und nicht optimaler Bewertungen.

3.3. Gruppen der Gestalt $G \rtimes_\varphi S_{\mathfrak{P}_0}$

In diesem Abschnitt sei $\mathfrak{P}_0$ eine fest gewählte kanonische Partition von $[n]$. Wir wollen nun unser Vorgehen für Gruppenoperationen der Gestalt $G \rtimes_\varphi S_{\mathfrak{P}_0}$ mit einem Gruppenhomomorphismus $\varphi : S_{\mathfrak{P}_0} \to \mathrm{Aut}(G)$ auf einer Menge X näher beschreiben. Da wir uns φ fest vorgeben, werden wir auf die weitere Angabe von φ innerhalb des semidirekten Produkts zu Gunsten der Übersichtlichkeit verzichten.

Wie wir bereits in Abschnitt 3.1.1 gesehen haben, können wir einen Kanonisierer zu dieser Operation auch aus einem Kanonisierer $\mathrm{Can}_{S_{\mathfrak{P}_0}}$ zu der Operation von $S_{\mathfrak{P}_0}$ auf $G\backslash\backslash X$ über das Homomorphieprinzip gewinnen. Wir werden auf diese Entsprechung später eingehen, zunächst beschreiben wir das Backtracking aber über den Suchbaum $\overline{T}(x, G \rtimes S_{\mathfrak{P}_0})$. Als Beispiel und Motivation dient uns die Gruppenoperation von

$$((\mathrm{GL}_k(R) \times (R^*)^n) \rtimes \mathrm{Aut}(R)) \rtimes S_n$$

auf der Menge $R^{k\times n,(m,\dots,m)}$ aller Generatormatrizen von freien Codes vom Rang k.

3.3.1 Bemerkung. Unser Vorgehen ist maßgeblich durch diese Gruppenoperation motiviert. Im Allgemeinen haben wir mit diesem Ansatz zur Definition des Kanonisierers für derartige Gruppenoperationen, d.h. für Gruppenoperationen, die sich auch als Operation von G auf $S_n\backslash\backslash Y^n$ mit $X = Y^n$ beschreiben lassen, bislang ausnahmslos positive Erfahrungen gemacht. Gegebenenfalls sind für Gruppenoperationen aus andersartigen Problemstellungen aber auch grundverschiedene Ansätze zur Gewinnung eines effizienten Kanonisierers nötig.

Über die Wahl der Verfeinerung und der Partitionierung werden wir überdies steuern, dass die zulässigen Knoten des Wurzelbaums $\overline{T}(x, G \rtimes S_{\mathfrak{P}_0})$ bzw. die auftretenden Rechtsnebenklassen, auf die Menge

$$\overline{\mathcal{C}}(G \rtimes S_{\mathfrak{P}_0}) := \left\{H \rtimes S_{\mathfrak{P}}(g;\pi) \mid H \rtimes S_{\mathfrak{P}} \in \overline{\mathcal{L}}(G \rtimes S_{\mathfrak{P}_0}) \text{ und } (g;\pi) \in G \rtimes S_{\mathfrak{P}_0}\right\}$$

zu Untergruppen der Gestalt

$$\overline{\mathcal{L}}(G \rtimes S_{\mathfrak{P}_0}) := \{H \rtimes S_{\mathfrak{P}} \mid H \leq G, \mathfrak{P} \preceq \mathfrak{P}_0 : H \rtimes S_{\mathfrak{P}} \in \mathcal{L}(G \rtimes S_{\mathfrak{P}_0})\} \tag{3.6}$$

beschränkt ist. Dementsprechend beschränken wir auch Definitions- und Bildbereich der Verfeinerungen bzw. Partitionierungsvorschrift bereits im Vorfeld auf:

$$\overline{V}_i : X \times \overline{\mathcal{C}}(G \rtimes S_{\mathfrak{P}_0}) \to \overline{\mathcal{C}}(G \rtimes S_{\mathfrak{P}_0}),\ i \in [v]$$

und

$$I : X \times \overline{\mathcal{C}}(G \rtimes S_{\mathfrak{P}_0}) \to \overline{\mathcal{L}}(G \rtimes S_{\mathfrak{P}_0}).$$

Im Folgenden wollen wir zu $i \in [v]$ mit $\left(f^{(i)}_{H\rtimes S_{\mathfrak{P}}}\right)_{H\rtimes S_{\mathfrak{P}} \in \overline{\mathcal{L}}(G\rtimes S_{\mathfrak{P}_0})}$ diejenige Familie von $H \rtimes S_{\mathfrak{P}}$-Homomorphismen bezeichnen, welche zur Definition der Verfeinerung $\overline{V}_i$ vorgeschrieben wurde.

3.3.2 Bemerkung. In der Praxis ist die Menge der tatsächlich auftretenden Eingaben $(x, H \rtimes S_{\mathfrak{P}}(g;\pi))$ noch wesentlich stärker auf Teilmengen von $X \times \overline{\mathcal{C}}(G \rtimes S_{\mathfrak{P}_0})$ beschränkt: Es kann zum Beispiel $(H \rtimes S_{\mathfrak{P}}(g;\pi), i)$ nur dann als Knoten von $\overline{T}(x, G \rtimes S_{\mathfrak{P}_0})$ auftreten, wenn $f^{(0)}_{G \rtimes S_{\mathfrak{P}_0}}((g;\pi)x)$ einen kanonischen Repräsentanten unter der Operation mit $G \rtimes S_{\mathfrak{P}_0}$ definiert und $H \rtimes S_{\mathfrak{P}} \leq \mathrm{Stab}_{G \rtimes S_{\mathfrak{P}_0}}\left(f^{(0)}_{G \rtimes S_{\mathfrak{P}_0}}((g;\pi)x)\right)$ ist. Diese Beobachtung lässt sich leicht induktiv auf die weiteren, zum Einsatz gebrachten Homomorphismen fortsetzen.

Die Verfeinerung V soll nun aus einer iterierten Anwendung von zwei Verfeinerungen $V^{(\mathrm{im})}$ und $V^{(a)}$, die abwechselnd zur Anwendung kommen, aufgebaut werden. Diese Verfeinerungen werden wiederum aus den Homomorphismen $f^{(\mathrm{im})}_{H \rtimes S_{\mathfrak{P}}}$ bzw. $f^{(a)}_{H \rtimes S_{\mathfrak{P}}}$ gewonnen. Die iterierte Verfeinerung beginnt mit der Anwendung von $V^{(\mathrm{im})}$ und endet auch mit dieser, d.h. $v \geq 3$ sei ungerade und für $i \in [v]$ und $H \rtimes S_{\mathfrak{P}} \in \overline{\mathcal{L}}(G \rtimes S_{\mathfrak{P}_0})$ sei stets

$$\overline{V}_i := \begin{cases} V^{(\mathrm{im})}, & \text{falls } i \text{ gerade} \\ V^{(a)}, & \text{falls } i \text{ ungerade} \end{cases} \quad \text{bzw.} \quad f^{(i)}_{H \rtimes S_{\mathfrak{P}}} := \begin{cases} f^{(\mathrm{im})}_{H \rtimes S_{\mathfrak{P}}}, & \text{falls } i \text{ gerade} \\ f^{(a)}_{H \rtimes S_{\mathfrak{P}}}, & \text{falls } i \text{ ungerade.} \end{cases}$$

Mit der Verfeinerung $V^{(a)}$ – wir nennen sie auch die *äußere Verfeinerung* – wollen wir eine Einschränkung der zulässigen Permutationen erzielen. Sie lässt bei jedem Aufruf $V^{(a)}(x, (H \rtimes S_{\mathfrak{P}})(g;\pi))$ für ein beliebiges $x \in X$ und $(H \rtimes S_{\mathfrak{P}})(g;\pi) \in \overline{\mathcal{C}}(G \rtimes S_{\mathfrak{P}_0})$ die Untergruppen H unverändert. Wir gewinnen sie über eine Familie von $(H \rtimes S_{\mathfrak{P}})$-Homomorphismen $f^{(a)}_{H \rtimes S_{\mathfrak{P}}} : X \to Z^n$, Z eine beliebige geordnete Menge, wobei die kanonische Young-Untergruppe $S_{\mathfrak{P}} \leq S_{\mathfrak{P}_0}$ auf natürliche Weise auf Z^n operiert und $f^{(a)}_{H \rtimes S_{\mathfrak{P}}}$ unter der Untergruppe H invariant ist. Der kanonische Repräsentant im Bildbereich berechnet sich dann über das lexikographische Sortieren innerhalb der Blöcke $P \in \mathfrak{P}$. Somit sind aber die Stabilisatoren der kanonischen Repräsentanten wiederum kanonische Young-Untergruppen.

Die zweite Verfeinerung $V^{(\mathrm{im})}$ werden wir auch die *innere Kanonisierung* nennen. Sie trägt nicht zur Verfeinerung der Partition $\mathfrak{P}$ bei, sondern wird ausschließlich Auswirkung auf die Untergruppe H von $H \rtimes S_{\mathfrak{P}}$ haben. Wir werden im nachfolgenden Abschnitt die Strategie zur Definition einer Verfeinerung $V^{(\mathrm{im})}$ aufzeigen. Eine detaillierte (problemspezifische) Beschreibung findet sich in den nachfolgenden Kapiteln. Die Bereitstellung dieser Funktion wird den Hauptteil unserer Arbeit zur Kanonisierung linearer Codes über endlichen Kettenringen bilden.

Die Partitionierung werden wir, wie bei der Graphenkanonisierung, über eine Individualisierung vornehmen. Wir wählen also zu einem gegebenen Paar $(x, H \rtimes S_{\mathfrak{P}}(g;\pi)) \in X \times \overline{\mathcal{C}}(G \rtimes S_{\mathfrak{P}_0})$ mit nicht diskreter Partition $\mathfrak{P}$ einen nicht trivialen Block $P \in \mathfrak{P}$ und definieren $I(x, H \rtimes S_{\mathfrak{P}}(g;\pi)) := H \rtimes \mathrm{Stab}_{S_{\mathfrak{P}}}(\min(P))$. Die Auswahl des Blocks P hat dabei $G \rtimes S_{\mathfrak{P}_0}$-invariant zu erfolgen. Offensichtlich liegt mit dieser Wahl auch $H \rtimes \mathrm{Stab}_{S_{\mathfrak{P}}}(\min(P))$ in $\overline{\mathcal{L}}(G \rtimes S_{\mathfrak{P}_0})$.

Ist $\mathfrak{D}$ die diskrete kanonische Partition von $[n]$, so werden wir für Knoten der Gestalt $((H \rtimes S_{\mathfrak{D}})(g;\pi), v)$ in $\overline{T}(x, G \rtimes S_{\mathfrak{P}_0})$, anstatt einer Partitionierung von H, das Bahnenelement $(g;\pi)x$ direkt unter der Operation von H kanonisieren. Wir schrumpfen also – zumindest in der Modellierung – den Teilbaum, den wir an dieser Stelle noch untersuchen müssten, auf einen einzigen Knoten zusammen. Diese Sichtweise erlaubt es uns, in Abschnitt 3.3.2 einen isomorphen Suchbaum $\overline{T}(Gx, S_{\mathfrak{P}_0})$ zu definieren.

Das hier entworfene Vorgehen ist, wie bereits oben erwähnt, vor allem durch unsere Problemstellungen aus der Codierungstheorie motiviert. Wir werden nämlich Gruppenoperation ${}_GX$ untersuchen, bei denen wir in der Lage sind, effiziente Kanonisierungsalgorithmen für die Untergruppen $H \leq G$ zur Verfügung zu stellen. In diesem Fall müssen wir also nicht das Problem über eine Partitionierung in leichtere Problemstellungen überführen.

3.3.1. Innere Kanonisierung

Für die Definition der Verfeinerung $V^{(\mathrm{im})}$ setzen wir voraus, dass für jedes $i \in [n]$ ein $(G \rtimes_\varphi \mathrm{Stab}_{S_{\mathfrak{P}_0}}(i))$-Homomorphismus $\Pi_i : X \to Y^{(i)}$ in eine Menge $Y^{(i)}$ gegeben sei, welcher $\mathrm{Stab}_{S_{\mathfrak{P}_0}}(i)$-invariant ist. Auf $Y^{(i)}$ stehe des Weiteren für jede Untergruppe $H \leq G$ ein Kanonisierer $\mathrm{Can}_H^{Y^{(i)}}$ zur Verfügung. Mit Y wollen wir die disjunkte[10] Vereinigung der $Y^{(i)}$ bezeichnen. Die Abbildung $\Pi := (\Pi_0, \ldots, \Pi_{n-1}) : X \to Y^n$ ist dann auch ein G-Homomorphismus.

3.3.3 Beispiel. Für das Beispiel aus der Codierungstheorie wählen wir die Projektion $\Pi_i : R^{k\times n,(m,\ldots,m)} \to (R^k)_R, \quad \Gamma \mapsto \Gamma_{*,i}$ auf die i-te Spalte der Generatormatrix. Ein Gruppenelement

$$(A, \varphi; \alpha, \pi) \in ((\mathrm{GL}_k(R) \times (R^*)^n) \rtimes \mathrm{Aut}(R)) \rtimes \mathrm{Stab}_{S_n}(i)$$

operiert dann über $(A, \varphi; \alpha, \pi)v := A\alpha(v)\varphi_i^{-1}$ auf $v \in (R^k)_R$.

Das Bild $f^{(\mathrm{im})}_{H\rtimes S_{\mathfrak{P}}}((g;\pi)x)$ am Knoten $H \rtimes S_{\mathfrak{P}}(g;\pi)$ wollen wir nun dadurch gewinnen, dass wir für eine ausgewählte Teilfolge $(i_0, \ldots, i_{\ell-1})$ von Koordinaten aus der Fixpunktmenge $\mathrm{Fix}_{S_{\mathfrak{P}}}([n])$ die Projektion

$$f^{(\mathrm{im})}_{H\rtimes S_{\mathfrak{P}}}((g;\pi)x) := \big(\Pi_{i_0}((g;\pi)x), \ldots, \Pi_{i_{\ell-1}}((g;\pi)x)\big)$$

bilden. Die Auswahl der Koordinatenfolge $(i_0, \ldots, i_{\ell-1})$ erfolgt nicht starr, sondern ist über eine Funktion $(i_0, \ldots, i_{\ell-1}) := F(x, H \rtimes S_{\mathfrak{P}}(g;\pi))$ auf die aktuelle Situation zugeschnitten.

Im weiteren Verlauf soll nun die Wahl der Koordinaten $(i_0, \ldots, i_{\ell-1})$ näher beschrieben werden, da wieder alle formalen Voraussetzungen an eine Verfeinerung zu erfüllen sind.

[10] Gegebenenfalls erreichen wir dies über eine künstliche Unterscheidung der Elemente, d.h. durch Betrachtung des kartesischen Produkts $Y^{(i)} \times \{i\}$.

Wir geben zunächst alle zulässigen Zuordnungen eines Knotens $H \rtimes S_{\mathfrak{P}}(g;\pi)$ auf die Folge $(i_0,\ldots,i_{\ell-1})$ mit Hilfe der folgenden Definition an:

3.3.4 Definition (Fixierreihenfolge)**.** Wir nennen eine Funktion

$$F : X \times \overline{\mathcal{L}}(G \rtimes S_{\mathfrak{P}_0}) \to \bigcup_{i=0}^{n} [n]^i =: [n]^{\leq n}$$

eine *Fixierreihenfolge*, falls F die folgenden Eigenschaften für $x \in X$ und $(H \rtimes S_{\mathfrak{P}}) \in \overline{\mathcal{L}}(G \rtimes S_{\mathfrak{P}_0})$ erfüllt:

- $F(x,(H \rtimes S_{\mathfrak{P}}))$ ist injektiv und $F(x, H \rtimes S_{\mathfrak{P}}) = F((h;\pi)x, H \rtimes S_{\mathfrak{P}})$ für alle $(h;\pi) \in H \rtimes S_{\mathfrak{P}}$,
- die Einträge des Vektors $F(x,(H \rtimes S_{\mathfrak{P}}))$ sind Fixpunkte unter $S_{\mathfrak{P}}$.

3.3.5 Notation. Ist $(i_0,\ldots,i_{\ell-1}) := F(x, H \rtimes S_{\mathfrak{P}})$, so bezeichne

$$\Pi_{F(x,H\rtimes S_{\mathfrak{P}})} := \left(\Pi_{i_0},\ldots,\Pi_{i_{\ell-1}}\right)$$

das direkte Produkt, der durch $F(x, H \rtimes S_{\mathfrak{P}})$ ausgewählten Homomorphismen.

3.3.6 Hilfssatz. *Es sei $(H \rtimes S_{\mathfrak{P}}) \in \overline{\mathcal{L}}(G \rtimes S_{\mathfrak{P}_0})$ beliebig und F eine gegebene Fixierreihenfolge. Die Funktion*

$$f^{(\mathrm{im})}_{H\rtimes S_{\mathfrak{P}}} : X \to Y^{\leq n}, \quad x \mapsto \Pi_{F(x,H\rtimes S_{\mathfrak{P}})}(x)$$

definiert einen $H \rtimes S_{\mathfrak{P}}$-Homomorphismus und eine $S_{\mathfrak{P}}$-Invariante.

Beweis. Es sei $x \in X$ und $(H \rtimes S_{\mathfrak{P}}) \in \overline{\mathcal{L}}(G \rtimes S_{\mathfrak{P}_0})$ beliebig. Weiter sei $(i_0,\ldots,i_{\ell-1}) := F(x, H \rtimes S_{\mathfrak{P}})$ und $j \in [l]$ beliebig. Dann ist die Gruppe $H \rtimes S_{\mathfrak{P}}$ eine Untergruppe von $G \rtimes \mathrm{Stab}_{S_{\mathfrak{P}_0}}(i_j)$ und somit

$$\Pi_{i_j}((h;\pi)x) = (h;\pi)\Pi_{i_j}(x) = h\Pi_{i_j}(x) \text{ für alle } (h;\pi) \in H \rtimes S_{\mathfrak{P}}.$$

Damit folgt aber sofort, dass $f^{(\mathrm{im})}_{H\rtimes S_{\mathfrak{P}}}$ ein H-Homomorphismus und $S_{\mathfrak{P}}$-invariant ist. □

Mit Hilfe der Homomorphismen $f^{(\mathrm{im})}_{H\rtimes S_{\mathfrak{P}}}$ definieren wir nun die Verfeinerung $V^{(\mathrm{im})}$ über den Satz 3.2.4. Die Kanonisierer im Bildbereich gewinnen wir dabei über das iterative Anwenden der Kanonisierer auf das kartesische Produkt $Y^{(i_0)} \times \ldots \times Y^{(i_{\ell-1})}$ für $(i_0,\ldots,i_{\ell-1}) := F(x, H \rtimes S_{\mathfrak{P}})$.

3.3.7 Bemerkung. Bei der Kanonisierung linearer Codes über endlichen Körper kann man die Fixierreihenfolge derart wählen, dass die Folgen $F(x, H \rtimes S_{\mathfrak{P}})$ – unabhängig von $x \in X$ – zu $(H \rtimes S_{\mathfrak{P}}) \in \overline{\mathcal{L}}(G \rtimes S_{\mathfrak{P}_0})$ stets alle Fixpunkte $\mathrm{Fix}_{S_{\mathfrak{P}}}([n])$ umfassen.

Für lineare Codes über endliche Kettenringe wird sich später herausstellen, dass nur Spalten hinzugenommen werden sollten, die bestimmte Eigenschaften im Hinblick auf die Operation von H erfüllen. Über die Fixierreihenfolge beschränken wir die tatsächlich im Backtracking zu $x \in X$ vorkommenden Nebenklassen $(H \rtimes S_{\mathfrak{P}})(g;\pi) \in \overline{\mathcal{C}}(G \rtimes S_{\mathfrak{P}_0})$ derart, dass wir stets einen effiziente Kanonisierung für die Operation von H auf $\Pi_{F((g;\pi)x, H \rtimes S_{\mathfrak{P}})}((g;\pi)x)$ zur Verfügung stellen können. Diese Beobachtung bildet die Motivation für das in diesen Abschnitt beschriebene Vorgehen.

Wir wollen nun noch abschließend die berechneten Bahnenelemente $(g;\pi)x$ zu den Knoten $((H \rtimes S_{\mathfrak{P}})(g;\pi), i)$ in $\overline{T}(x, G \rtimes S_{\mathfrak{P}_0})$ genauer beschreiben. Zunächst sei $f \subseteq [n]^j$ ein beliebiges injektives Wort der Länge $j \in [n+1]$.

3.3.8 Definition (f-semikanonischer Repräsentant)**.** Es sei $j \in [n+1]$ und $f \subseteq [n]^j$ injektiv. Wir nennen ein Element $x \in X$ mit $\mathrm{CF}_G(\Pi_f(x)) = \Pi_f(x)$ einen *f-semikanonischen* Repräsentanten der Bahn Gx. Für beliebiges $x \in X$ sei dann

$$G^{(f,x)} := \mathrm{Stab}_G(\mathrm{CF}_G(\Pi_f(x)))$$

der Stabilisator des Bildes eines f-semikanonischen Repräsentanten der Bahn.

3.3.9 Definition. Sind $e, f \in [n]^{\leq n}$ injektiv, so definieren wir die Differenz $f \setminus e$ als diejenige Teilfolge von f, welche alle Folgenglieder enthält, die nicht in der Folge e auftreten. Unter $e + f := (e, f \setminus e)$ verstehen wir dann das Wort, welches sich durch Anfügen von $f \setminus e$ an e ergibt.

Zu einem Knoten $(H \rtimes S_{\mathfrak{P}}(g;\pi), i)$ von $\overline{T}(x, G \rtimes S_{\mathfrak{P}_0})$ sei $\overline{F}(x, H \rtimes S_{\mathfrak{P}}(g;\pi), i) \in [n]^{\leq n}$ dasjenige injektive Wort, welches die Fixierreihenfolge der Koordinaten auf dem Weg von der Wurzel zum aktuellen Knoten $(H \rtimes S_{\mathfrak{P}}(g;\pi), i)$ in $\overline{T}(x, G \rtimes S_{\mathfrak{P}_0})$ wiedergibt. Wir können $\overline{F}(x, H \rtimes S_{\mathfrak{P}}(g;\pi), i)$ induktiv definieren:

- Für die Wurzel sei $\overline{F}(x, G \rtimes S_{\mathfrak{P}_0}, 0) := ()$.
- Andernfalls sei $(H' \rtimes S_{\mathfrak{Q}}(g;\pi), j)$ der Vater von $(H \rtimes S_{\mathfrak{P}}(g;\pi), i)$, dann ist

$$\overline{F}(x, H \rtimes S_{\mathfrak{P}}(g;\pi), i) := \begin{cases} \overline{F}(x, H' \rtimes S_{\mathfrak{Q}}(g;\pi), j), & \text{falls } i \text{ gerade} \\ \overline{F}(x, H' \rtimes S_{\mathfrak{Q}}(g;\pi), j) + F((g;\pi)x, H' \rtimes S_{\mathfrak{Q}}), & \text{sonst.} \end{cases}$$

3.3.10 Bemerkung. Aus der rekursiven Definition ergibt sich auch leicht, dass die Funktion $\overline{F}$ eine $(G \rtimes S_{\mathfrak{P}_0})$-Invariante ist, d.h. für ein beliebiges $(g_0, \pi_0) \in G \rtimes S_{\mathfrak{P}_0}$ ist

$$\overline{F}\big((g_0,\pi_0)x, H \rtimes S_{\mathfrak{P}}(g;\pi)(g_0,\pi_0)^{-1}, i\big) = \overline{F}(x, H \rtimes S_{\mathfrak{P}}(g;\pi), i)$$

3.3.11 Hilfssatz. *Es sei $(H \rtimes S_{\mathfrak{P}}(g;\pi), i)$ ein Knoten von $\overline{T}(x, G \rtimes S_{\mathfrak{P}_0})$ und $f := \overline{F}(x, H \rtimes S_{\mathfrak{P}}(g;\pi), i)$. Dann ist $(g;\pi)x$ ein f-semikanonischer Repräsentant von $G\pi x$ und $H = G^{(f,\pi x)}$.*

Beweis. Wir beweisen die Aussage über eine Induktion nach der Tiefe t des Knotens $(H \rtimes S_{\mathfrak{P}}(g;\pi), i)$ im Suchbaum $\overline{T}(x, G \rtimes S_{\mathfrak{P}_0})$. Für die Wurzel ist nichts zu zeigen. Ist t gerade, so haben wir diesen Knoten entweder über eine Individualisierung oder über eine Anwendung der Verfeinerung $V^{(a)}$ erreicht. In beiden Fällen bleibt die Gruppe $H = H'$ des Vaterknotens $((H' \rtimes S_{\mathfrak{Q}})(g';\pi'), j)$ unverändert und es wurde ein Gruppenelement $(h;\sigma) \in H \rtimes S_{\mathfrak{Q}}$ angewandt. Die fixierten Koordinaten f sind ebenfalls für beide Knoten identisch. Die Gleichung

$$\begin{aligned}\Pi_f((g;\pi)x) &= \Pi_f((h;\sigma)(g';\pi')x) = h\Pi_f((g';\pi')x) = \Pi_f((g';\pi')x)\\ &= \mathrm{CF}_G(\Pi_f((g';\pi')x)) = \mathrm{CF}_G(\Pi_f((g;\pi)x))\end{aligned}$$

impliziert sofort, dass $(g;\pi)x$ ein f-semikanonischer Repräsentant der Bahn $G\pi x$ ist.

Ist die Tiefe t ungerade, so ist auch i ungerade. Der Knoten $(H \rtimes S_{\mathfrak{P}}(g;\pi), i)$ ist also über die Verfeinerung $V^{(\mathrm{im})}$ aus dem Vater $(H' \rtimes S_{\mathfrak{P}}(g';\pi'), i-1)$ hervorgegangen. Wir können ohne Beschränkung der Allgemeinheit $\pi = \pi'$ annehmen. Es sei nun $e := \overline{F}(x, H' \rtimes S_{\mathfrak{P}}(g';\pi'), i-1)$. Dann ist $(g';\pi)x$ ein e-semikanonischer Repräsentant und $H' = G^{(e,\pi x)}$ nach der Induktionsvoraussetzung. Aus der Definition der inneren Kanonisierung

$$(H \rtimes S_{\mathfrak{P}})(g;\pi) = V^{(\mathrm{im})}(x, H' \rtimes S_{\mathfrak{P}}(g';\pi)) = (H \rtimes S_{\mathfrak{P}}) \underbrace{\mathrm{TR}_{H'}(\Pi_f((g';\pi)x))(g';\pi)}_{=(g;\pi)}$$

lässt sich leicht folgern, dass

$$\begin{aligned}\Pi_f((g;\pi)x) &= \big(\Pi_e((g;\pi)x), \Pi_{f\setminus e}((g;\pi)x)\big)\\ &= \big(\Pi_e((g;\pi)x), \Pi_{f\setminus e}(\mathrm{TR}_{H'}(\Pi_f((g';\pi)x))(g';\pi)x)\big)\\ &= \big(\mathrm{CF}_G(\Pi_e((g;\pi)x)), \mathrm{CF}_{H'}(\Pi_{f\setminus e}((g;\pi)x))\big)\\ &= \mathrm{CF}_G(\Pi_f((g;\pi)x))\end{aligned}$$

gilt. Hieraus schließt man leicht, dass die berechnete Untergruppe

$$\begin{aligned}H &:= \mathrm{Stab}_{H'}(\mathrm{CF}_{H'}(\Pi_f((g';\pi)x))) = \mathrm{Stab}_{H'}(\Pi_f((g;\pi)x))\\ &= \mathrm{Stab}_G(\Pi_f((g;\pi)x)) = G^{(f,\pi x)}\end{aligned}$$

der Stabilisator der Projektion eines f-semikanonischen Repräsentanten ist. □

3.3.2. Gleichwertiger Algorithmenentwurf

In diesem Abschnitt möchten wir nun einen Suchbaum $\overline{T}(Gx, S_{\mathfrak{P}_0})$ zur Kanonisierung von Gx unter der Operation von $S_{\mathfrak{P}_0}$ definieren, welcher zu $\overline{T}(x, G \rtimes S_{\mathfrak{P}_0})$ isomorph ist. Damit können wir zeigen, dass der von uns entwickelte Kanonisierer sowohl über die Suchbäume $\overline{T}(Gx, S_{\mathfrak{P}_0})$ als auch $\overline{T}(x, G \rtimes_{\varphi} S_{\mathfrak{P}_0})$ definiert werden kann.

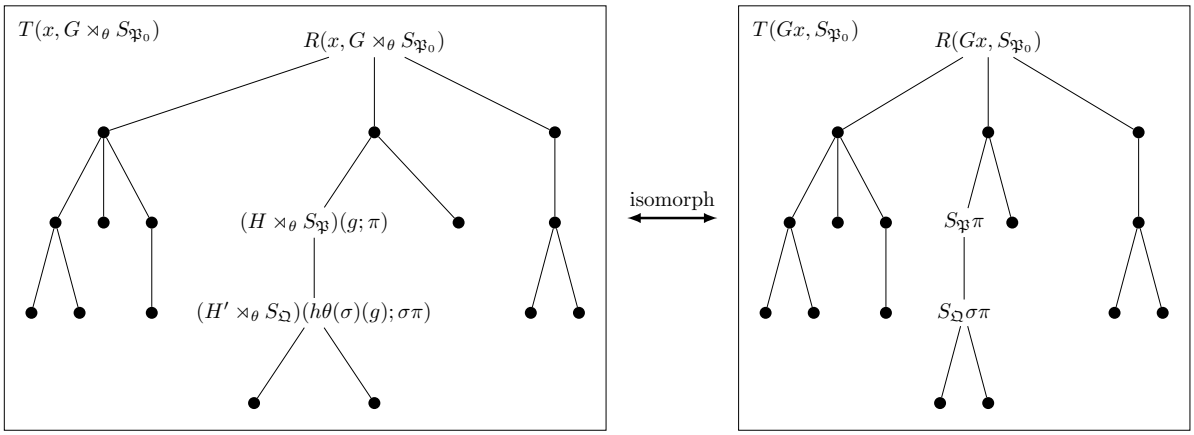


Abbildung 3.7.: Isomorphie der Suchbäume

Abbildung 3.7 zeigt eine Visualisierung des Isomorphismus für die ursprünglichen Bäume $T(Gx, S_{\mathfrak{P}_0})$ und $T(x, G \rtimes S_{\mathfrak{P}_0})$, welchen wir herleiten wollen. Diese Beobachtung stellt dann auch die Verbindung zu dem in meiner Diplomarbeit [22] entwickelten Kanonisierer für lineare Codes über endlichen Körpern her.

Einen zu $\overline{T}(x, G \rtimes S_{\mathfrak{P}_0})$ isomorphen Wurzelbaum $\overline{T}(Gx, S_{\mathfrak{P}_0})$ erhalten wir, wie bereits in Abbildung 3.7 beschrieben, über die Bijektion $(H \rtimes S_{\mathfrak{P}}(g;\pi), i) \mapsto (S_{\mathfrak{P}}\pi, i)$. Die Partitionierungsvorschrift ergibt sich offensichtlich durch die Individualisierung des jeweils gleichen Blocks.

Es bleibt zu zeigen, dass der Wurzelbaum $\overline{T}(Gx, S_{\mathfrak{P}_0})$ tatsächlich auch über die Wahl geeigneter Verfeinerungen gewonnen werden kann. Für die Definition der Verfeinerungen genügt es nicht, nur jeweils eine Familie $\overline{f}^{(a)}_{S_{\mathfrak{P}}}$ bzw. $\overline{f}^{(\mathrm{im})}_{S_{\mathfrak{P}}}$ von $S_{\mathfrak{P}}$-Homomorphismen zur Verfügung zu stellen, da diese offensichtlich die unterschiedlichen auftretenden Untergruppen $H \leq G$ nicht ausreichend modellieren können. Wir geben aus diesem Grund für jede mögliche Tiefe $t \in \mathbb{N}$ des Backtracking eine Familie $\overline{f}^{(t)}_{\mathfrak{P}}$ von $S_{\mathfrak{P}}$-Homomorphismen an, welche einen passenden Verfeinerer $\overline{V}_t$ definieren.

Der Beweis ist etwas technisch, wir geben später an, wie wir tatsächlich den Algorithmus umsetzen werden. Zunächst definieren wir die Homomorphismen, welche zu der inneren Kanonisierung korrespondieren.

3.3.12 Hilfssatz. *Ist $t \geq 0$ gerade und $\mathfrak{P} \preceq \mathfrak{P}_0$ eine kanonische Partition, so definiert*

$$\overline{f}^{(t)}_{\mathfrak{P}} : G\backslash\backslash X \to Y^{\leq n}$$
$$Gx \mapsto \begin{cases} \mathrm{CF}_H\left(f^{(\mathrm{im})}_{H \rtimes S_{\mathfrak{P}}}(\overline{x})\right), & \exists\, \overline{x} \in Gx, H \rtimes S_{\mathfrak{P}} \in \overline{\mathcal{L}}(G \rtimes S_{\mathfrak{P}_0}) \text{ sd. } (H \rtimes S_{\mathfrak{P}}, i) \\ & \text{ein Knoten der Tiefe } t \text{ in } \overline{T}(\overline{x}, G \rtimes S_{\mathfrak{P}_0}) \\ (), & \text{sonst} \end{cases}$$

eine $S_{\mathfrak{P}}$-Invariante. Dabei bezeichne $()$ die leere Folge.

Beweis. Die Abbildungsvorschrift ist wohldefiniert:

- Aus der Partitionierung von $\mathfrak{P}_0$ folgt zunächst, dass es für beliebiges $\overline{x} \in Gx$ höchstens einen Knoten $((H \rtimes S_\mathfrak{P}), i)$ der Tiefe t in $\overline{T}(\overline{x}, G \rtimes S_{\mathfrak{P}_0})$ dieser Gestalt geben kann. Alle weiteren Knoten dieser Tiefe sind Rechtsnebenklassen mit nicht trivialer Permutationskomponente.

- Existiert ein weiteres Bahnenelement $g_0\overline{x}$, $g_0 \in G$, mit dieser Eigenschaft, so ist $(H \rtimes S_\mathfrak{P})g_0^{-1}$ der eindeutig bestimmte Knoten in $\overline{T}(g_0\overline{x}, G \rtimes S_{\mathfrak{P}_0})$ der Tiefe t, der als Rechtsnebenklasse mit trivialer Permutation geschrieben werden kann. In diesem Fall ist dann $g_0 \in H$ und $\mathrm{CF}_H\Big(f^{(\mathrm{im})}_{H \rtimes S_\mathfrak{P}}(\overline{x})\Big) = \mathrm{CF}_H\Big(f^{(\mathrm{im})}_{H \rtimes S_\mathfrak{P}}(g_0\overline{x})\Big)$.

Nun zu der $S_\mathfrak{P}$-Invarianz der Abbildungsvorschrift. Es genügt die Situation zu betrachten, in welcher zu $Gx \in G\backslash\backslash X$ ein solches Gruppenelement $\overline{x} \in Gx$ existiert (für den anderen Fall ist die Aussage offensichtlich wahr). Ist $\pi \in S_\mathfrak{P}$ beliebig, so ist wieder $((H \rtimes S_\mathfrak{P})\pi^{-1} = (H \rtimes S_\mathfrak{P}), i)$ ein Knoten in $\overline{T}(\pi\overline{x}, G \rtimes S_{\mathfrak{P}_0})$ der Tiefe t. Somit folgt:

$$\overline{f}^{(t)}_\mathfrak{P}(\pi Gx) = \mathrm{CF}_H\Big(f^{(\mathrm{im})}_{H \rtimes S_\mathfrak{P}}(\pi\overline{x})\Big) = \mathrm{CF}_H\Big(f^{(\mathrm{im})}_{H \rtimes S_\mathfrak{P}}(\overline{x})\Big) = \overline{f}^{(t)}_\mathfrak{P}(Gx) \qquad \square$$

Analog beweist man für Knoten auf ungerader Tiefe t:

3.3.13 Hilfssatz. *Ist $t \geq 0$ ungerade, $t \not\equiv v \mod v+1$ und $\mathfrak{P} \preceq \mathfrak{P}_0$, so definiert*

$$\overline{f}^{(t)}_\mathfrak{P} : G\backslash\backslash X \to Z^n \cup \{()\}$$
$$Gx \mapsto \begin{cases} f^{(a)}_{H \rtimes S_\mathfrak{P}}(\overline{x}), & \exists\, \overline{x} \in Gx, H \rtimes S_\mathfrak{P} \in \overline{\mathcal{L}}(G \rtimes S_{\mathfrak{P}_0}) \text{ sd. } (H \rtimes S_\mathfrak{P}, i) \\ & \text{ein Knoten der Tiefe } t \text{ in } \overline{T}(\overline{x}, G \rtimes S_{\mathfrak{P}_0}) \\ (), & \text{sonst} \end{cases}$$

einen $S_\mathfrak{P}$-Homomorphismus.

Beweis. Die Wohldefiniertheit der Abbildungsvorschrift zeigt man, wie im vorausgegangenen Hilfssatz, indem man die H-Invarianz der Funktion $f^{(a)}_{H \rtimes S_\mathfrak{P}}$ benutzt. Für die $S_\mathfrak{P}$-Homomorphie genügt es wieder, diejenigen $Gx \in G\backslash\backslash X$ zu untersuchen, für welche ein solches Bahnenelement $\overline{x} \in Gx$ und eine Untergruppe $H \rtimes S_\mathfrak{P} \in \overline{\mathcal{L}}(G \rtimes S_{\mathfrak{P}_0})$ existiert. Ist $\pi \in S_\mathfrak{P}$ beliebig, so folgt:

$$\overline{f}^{(t)}_\mathfrak{P}(\pi Gx) = \overline{f}^{(t)}_\mathfrak{P}(\pi G\overline{x}) = f^{(a)}_{H \rtimes S_\mathfrak{P}}(\pi\overline{x}) = \pi f^{(a)}_{H \rtimes S_\mathfrak{P}}(\overline{x}) = \pi\overline{f}^{(t)}_\mathfrak{P}(Gx) \qquad \square$$

Induktiv lässt sich nun leicht zeigen, dass man über diese Familien von Homomorphismen genau den gewünschten Suchbaum $\overline{T}(Gx, S_{\mathfrak{P}_0})$ entwickelt. Wir geben nur die zentrale Idee für den Beweis am Beispiel der Knoten auf ungerader Tiefe t:

Ist $(H \rtimes S_{\mathfrak{P}}(g;\pi), i)$ ein Knoten in $\overline{T}(x, G \rtimes S_{\mathfrak{P}_0})$ der Tiefe t und wird für diesen die Verfeinerung $V^{(a)}$ gerufen, so ist das Ergebnis:

$$\begin{aligned}(H \rtimes S_{\mathfrak{Q}})(h;\sigma)(g;\pi) &:= V^{(a)}(x, H \rtimes S_{\mathfrak{P}}(g;\pi)) \\ &= \mathrm{Stab}_{H \rtimes S_{\mathfrak{P}}}\big(\mathrm{CF}_{H \rtimes S_{\mathfrak{P}}}(f^{(a)}((g;\pi)x))\big)\, \mathrm{TR}_{H \rtimes S_{\mathfrak{P}}}(f^{(a)}((g;\pi)x))(g;\pi)\end{aligned}$$

Für den isomorphen Knoten $(S_{\mathfrak{P}}\pi, i)$ in $\overline{T}(Gx, S_{\mathfrak{P}_0})$ ergibt sich dann an dieser Stelle als Ergebnis der entsprechend definierten Verfeinerung $\overline{V}_t$:

$$\overline{V}_t(Gx, S_{\mathfrak{P}}\pi) = \mathrm{Stab}_{S_{\mathfrak{P}}}\Big(\mathrm{CF}_{S_{\mathfrak{P}}}(\overline{f}^{(t)}_{\mathfrak{P}}(\pi Gx))\Big)\, \mathrm{TR}_{H \rtimes S_{\mathfrak{P}}}(\underbrace{\overline{f}^{(t)}_{\mathfrak{P}}(\pi Gx)}_{=f^{(a)}((g;\pi)x)})\pi = S_{\mathfrak{Q}}\sigma\pi$$

Wir geben nun noch abschließend einige Hinweise zur tatsächliche Algorithmenimplementierung:

- Zu jedem Knoten $(S_{\mathfrak{P}}\pi, i)$ in $\overline{T}(Gx, S_{\mathfrak{P}_0})$ verwalten wir im Hintergrund die injektive Folge e derjenigen Koordinaten, für welche die innere Kanonisierung bereits durchgeführt wurde. Zu dieser Folge e ist ein e-semikanonischer Repräsentant $x^{(\pi,e)}$ von $G\pi x$ und dessen Stabilisator $H = G^{(e,\pi x)} = \mathrm{Stab}_G\big(x^{(e,\pi)}\big)$ bestimmt.

- Ist i gerade, so bestimmen wir zunächst $f := F(x^{(e,\pi)}, H \rtimes S_{\mathfrak{P}})$ und bewerten den Knoten mit der $S_{\mathfrak{P}}$-Invarianten $\Pi_f(x^{(\pi,f)}) = \mathrm{CF}_H\big(\Pi_f(x^{(e,\pi)})\big)$ und setzen als Kind $(S_{\mathfrak{P}}\pi, i+1)$.

- Ist $i < r$ ungerade, so verfeinern wir die Partition $\mathfrak{P}$ über die Kanonisierung von $f^{(a)}_{H \rtimes S_{\mathfrak{P}}}(x^{(e,\pi)})$ unter Ausnutzung des Homomorphieprinzips. Ist

  $$(h;\sigma) := \mathrm{TR}_{H \rtimes S_{\mathfrak{P}}}\Big(f^{(a)}_{H \rtimes S_{\mathfrak{P}}}(x^{(e,\pi)})\Big)$$

  und

  $$S_{\mathfrak{P}} := \mathrm{Stab}_{S_{\mathfrak{P}}}\Big(\sigma f^{(a)}_{H \rtimes S_{\mathfrak{P}}}(x^{(e,\pi)}\Big),$$

  so ergibt sich der neue e-semikanonische Repräsentant $x^{(e,\sigma\pi)} := \sigma x^{(e,\pi)}$. Die Gruppe H muss nicht abgeändert werden und wir setzen als Kind $(S_{\mathfrak{Q}}\sigma\pi, i+1)$.

Die Isomorphie der beiden Suchbäume gibt uns nun auch einen Hinweis, wie wir das Abschneiden von Teilbäumen in $\overline{T}(x, G \rtimes S_{\mathfrak{P}_\circ})$ mit Hilfe der bekannten Automorphismen von x durchführen werden. Wir verwalten nur den Permutationsanteil, d.h. die Gruppe $\mathrm{Stab}_{S_{\mathfrak{P}_\circ}}(Gx)$, über ein Labelled Branching mit Basis $(0, \ldots, n-1)$. Das Abschneiden von Teilbäumen erfolgt wieder über den Hilfssatz 3.2.26.

Unter der Annahme, dass für alle $x \in X$ der Stabilisator $\mathrm{Stab}_G(x) = \{1_G\}$ trivial ist, können wir sogar zeigen, dass dies zu keinem Verlust von Informationen führt:

3.3.14 Hilfssatz. *Es sei* $A \leq G \rtimes_\varphi S_{\mathfrak{P}_0}$ *mit* $\overline{A} := \{\pi \mid (g;\pi) \in A\}$ *und* $|A| = |\overline{A}|$. *Ist* $T_{\overline{A}}$ *eine Linkstransversale von* $S_{\mathfrak{P}_0}/\overline{A}$, *so definiert* $T_A := \{(g;\pi) \mid g \in G, \pi \in T_{\overline{A}}\}$ *eine Linkstransversale von* $(G \rtimes_\varphi S_{\mathfrak{P}_0})/A$.

Beweis. Wir nehmen an, in der Menge T_A seien $(g_0;\pi_0), (g_1;\pi_1)$ Nebenklassenrepräsentanten zu der gleichen Linksnebenklassen von A. Es ist also

$$(g_0;\pi_0)^{-1}(g_1;\pi_1) = (\varphi(\pi_0^{-1})(g_0^{-1});\pi_0^{-1})(g_1;\pi_1) = (\varphi(\pi_0^{-1})(g_0^{-1}g_1);\pi_0^{-1}\pi_1) \in A$$

Hieraus schließen wir $\pi_0^{-1}\pi_1 \in \overline{A}$ und damit $\pi_0 = \pi_1$. Aufgrund unserer Voraussetzung $|A| = |\overline{A}|$ impliziert dies bereits $g_0 = g_1$. Die Vollständigkeit von T_A folgt sofort aus Anzahlgründen. □

3.3.15 Folgerung. *Ist* $A \leq G \rtimes S_{\mathfrak{P}_0}$ *wie oben,* $T_{\overline{A}}$ *eine Linkstransversale und* $(H \rtimes S_{\mathfrak{P}})(g;\pi) \in \overline{C}(G \rtimes S_{\mathfrak{P}_0})$ *beliebig, so gilt:*

$$T_A \cap (H \rtimes S_{\mathfrak{P}})(g;\pi) = \emptyset \Longleftrightarrow T_{\overline{A}} \cap S_{\mathfrak{P}}\pi = \emptyset$$

Beweis.

$$\begin{aligned} T_{\overline{A}} \cap S_{\mathfrak{P}}\pi \neq \emptyset &\Longleftrightarrow \exists\, \overline{\pi} \in T_{\overline{A}}, \sigma \in S_{\mathfrak{P}} : \overline{\pi} = \sigma\pi \\ &\Longleftrightarrow \exists\, (\overline{g};\overline{\pi}) \in T_A, (h;\sigma) \in H \rtimes S_{\mathfrak{P}} : (\overline{g};\overline{\pi}) = (h;\sigma)(g;\pi) \\ &\Longleftrightarrow T_A \cap (H \rtimes S_{\mathfrak{P}})(g;\pi) \neq \emptyset \end{aligned}$$

□

4. Endliche Kettenringe

Wir werden nun zunächst noch weitere Eigenschaften endlicher Kettenringe besprechen, welche wir bei der Formulierung eines Kanonisierers für lineare Codes benötigen werden. Der erste Abschnitt fasst hier im Wesentlichen Resultate aus [44] zusammen. Insbesondere wird auch auf die Struktur der additiven und multiplikativen Gruppe eingegangen. Anschließend leiten wir eine Strukturaussage über endliche Kettenringe her, welche sich zum Beispiel auch in [59] findet. Der nachfolgende Abschnitt behandelt dann die Automorphismengruppe eines Kettenrings.

Es sei weiter R stets ein endlicher Kettenring der Kettenlänge m mit $R/\operatorname{Rad}(R) \simeq \mathbb{F}_q$ und $q = p^r$ für eine Primzahl p und Exponenten r. Außerdem bezeichne $\theta \in R$ einen fest gewählten Erzeuger des Jacobson-Radikals $\operatorname{Rad}(R) = R\theta$.

4.1. Weitere Grundlagen

4.1.1 Definition (Teichmüller-Menge)**.** Eine Teilmenge $T \subseteq R$ nennen wir *Teichmüller-Menge* von R, falls $0 \in T$ und $T^* := T \setminus \{0\}$ ein multiplikativ abgeschlossenes Vertretersystem der Menge $(R/\operatorname{Rad}(R))^* \simeq \mathbb{F}_q^*$ ist. Wir nennen T^* auch eine *Teichmüller-Gruppe*.

4.1.2 Fakt. *Der endliche Kettenring R besitzt mindestens eine Teichmüller-Menge. Die Teichmüller-Gruppen sind gerade die zyklischen Untergruppen von R^* der Ordnung $q-1$. Teichmüller-Mengen gehen durch Konjugation mit Einheiten auseinander hervor. Ist R kommutativ, dann ist also die Teichmüller-Menge T eindeutig.*

4.1.3 Definition (Schiefpolynomring)**.** Es sei R ein Ring und $\sigma \in \operatorname{Aut}(R)$ ein Ringautomorphismus. Wir wollen mit $R[X;\sigma]$ den *Schiefpolynomring* über R zum Automorphismus σ bezeichnen. Dieser Ring unterscheidet sich von dem klassischen Polynomring $R[X]$ nur in der Definition der Multiplikation $Xa := \sigma(a)X$ eines Ringelements $a \in R$ mit der Unbestimmten X.

Über die Schiefpolynomringe können wir nichtkommutative Kettenringe konstruieren, siehe Fakt 4.1.25.

4.1.4 Beispiel. Es sei τ der Frobenius-Automorphismus von $\mathbb{F}_4$. Der Kettenring $R = \mathbb{F}_4[X;\tau]/(X^2)$ hat die folgenden Teichmüller-Mengen $T_0 = \{0, 1, a, a^2\} = \mathbb{F}_4$, $T_1 = \{0, 1, a + X, a^2 + X\}$, $T_2 = \{0, 1, a + aX, a^2 + aX\}$ und $T_3 = \{0, 1, a + a^2X, a^2 + a^2X\}$.

Im weiteren Verlauf dieser Arbeit sei stets T eine fest gewählte Teichmüller-Menge von R und ξ ein Erzeuger der Teichmüller-Gruppe T^*. Es gilt weiter:

4.1.5 Fakt (θ-adische Entwicklung)**.** *Der Kettenring R besitzt q^m Elemente und jedes beliebige $a \in R$ lässt sich eindeutig in der Form $a = \sum_{i=0}^{m-1} a_i\theta^i$, mit $a_i \in T$ schreiben. Wir nennen eine solche Darstellung die* θ-adische Entwicklung *von a bezüglich T. Das Ideal $\mathrm{Rad}(R)^k$, $k \in [m+1]$ hat somit genau q^{m-k} Elemente.*

4.1.6 Beispiel. Die Teichmüller-Menge von $\mathbb{Z}_9$ ist $\{0, 1, 8\}$ und $\theta = 3$ ist ein Erzeuger von $\mathrm{Rad}(\mathbb{Z}_9)$. Die 3-adische Entwicklung von 2 ist $2 = 8 \cdot 3^0 + 1 \cdot 3^1$. Neben dem Ringelement $\theta = 3$ können wir auch $\theta' = 6$ als Erzeuger des Jacobson-Radikals wählen. Die 6-adische Entwicklung von 2 ist dann $2 = 8 \cdot 6^0 + 8 \cdot 6^1$

4.1.7 Definition. Wir werden im Folgenden zu $i \in [m]$ mit $\mathrm{coeff}^{(i)} : R \to T, a \mapsto a_i$ diejenige Abbildung bezeichnen, welche das Ringelement $a \in R$ auf seinen eindeutigen Koeffizienten $a_i \in T$ der θ-adischen Entwicklung $a = \sum_{i=0}^{m-1} a_i\theta^i$ abbildet.

Den Ring R können wir (über die additive Gruppe) auch auf natürliche Weise als $\mathbb{Z}$-Modul interpretieren, indem wir für $z \in \mathbb{Z}$ und $a \in R$ die Skalarmultiplikation über

$$z \cdot a := \begin{cases} \underbrace{a + \ldots + a}_{z-\text{mal}}, & \text{falls } z \geq 0 \\ (-z) \cdot (-a), & \text{falls } z < 0 \end{cases}$$

definieren. Wir identifizieren dann die ganze Zahl z auch mit $z \cdot 1_R \in R$. Die größte Zahl $\varepsilon \in \mathbb{N}$ mit $p \cdot 1_R \in \mathrm{Rad}(R)^\varepsilon$ nennen wir den *Verzweigungsindex* zu R. Es ist stets $\varepsilon \geq 1$, da die Gleichung $\overline{p \cdot 1_R} = p \cdot 1_{\mathbb{F}_q} = 0_{\mathbb{F}_q}$ bereits $p \in \mathrm{Rad}(R)$ impliziert.

Wir nutzen nun diese Beobachtung aus, um die θ-adische Entwicklung eines Ringelements weiter zu verfeinern. Diese verfeinerte Beschreibung gibt uns dann die Möglichkeit, bequem eine Totalordnung auf R zu definieren.

4.1.8 Hilfssatz ((ξ, θ)-adische Entwicklung)**.** *Ist ξ ein Erzeuger der Teichmüller-Gruppe T^* von R, so gibt es zu jedem $a \in R$ eine eindeutige Darstellung der Gestalt*

$$a = \sum_{i=0}^{r-1} \sum_{j=0}^{m-1} a_{ij}\xi^{r-1-i}\theta^j = (\xi^{r-1}, \ldots, \xi^0) \underbrace{\begin{pmatrix} a_{0,0} & \ldots & a_{0,m-1} \\ \vdots & \ddots & \vdots \\ a_{r-1,0} & \ldots & a_{r-1,m-1} \end{pmatrix}}_{=:\mathrm{coeff}(a)} \begin{pmatrix} \theta^0 \\ \vdots \\ \theta^{m-1} \end{pmatrix}$$

mit $\mathrm{coeff}(a) \in [p]^{r \times m}$. Wir nennen sie die (ξ, θ)-adische Entwicklung *von a.*

Beweis. Besitzt jedes Ringelement eine (ξ, θ)-adische Entwicklung, so folgt die Eindeutigkeit sofort aus Anzahlgründen. Wir beweisen die Existenz einer (ξ, θ)-adischen Entwicklung nun induktiv für die Ringelemente $a \in \mathrm{Rad}(R)^k$ für alle $k \in [m+1]$. Für das Nullelement – gleichzeitig der Induktionsstart $\mathrm{Rad}(R)^m = \{0_R\}$ – gibt es offensichtlich eine solche Darstellung.

Es sei nun $k \in [m]$ und die Behauptung für alle Ringelemente aus dem Ideal $\mathrm{Rad}(R)^{k+1}$ bereits bewiesen. Wir wählen $a \in \mathrm{Rad}(R)^k \setminus \overline{\mathrm{Rad}(R)^{k+1}}$ beliebig. Da T ein Vertretersystem von $R/\mathrm{Rad}(R) \simeq \mathbb{F}_q$ bildet, können wir $\overline{\mathrm{coeff}^{(k)}(a)}$ eindeutig über die Linearkombination $\sum_{i=0}^{r-1} a_{ik}\overline{\xi}^{r-1-i}, a_{ik} \in [p]$ der $\mathbb{F}_p$-Basis $(1, \overline{\xi}, \ldots, \overline{\xi}^{r-1})$ von $\mathbb{F}_q$ darstellen. Es existiert also ein Element $b \in \mathrm{Rad}(R)$ mit $\mathrm{coeff}^{(k)}(a) - \sum_{i=0}^{r-1} a_{ik}\xi^{r-1-i} = b$ und es folgt

$$a' := a - \sum_{i=0}^{r-1} a_{ik}\xi^{r-1-i}\theta^k = b\theta^k + \sum_{j=k+1}^{m-1} \mathrm{coeff}^{(j)}(a)\theta^j \in \mathrm{Rad}(R)^{k+1}.$$

Für das Ringelement $a' \in \mathrm{Rad}(R)^{k+1}$ existiert nach der Induktionsvoraussetzung aber eine (ξ, θ)-adische Entwicklung $a' = \sum_{i=0}^{r-1}\sum_{j=k+1}^{m-1} a_{ij}\xi^{r-1-i}\theta^j$. Es ist also

$$a = \sum_{i=0}^{r-1}\sum_{j=k}^{m-1} a_{ij}\xi^{r-1-i}\theta^j$$

die gesuchte (ξ, θ)-adische Entwicklung von a. □

4.1.9 Definition. Wir werden im Folgenden zu $i \in [r]$ und $j \in [m]$ mit

$$\mathrm{coeff}^{(i,j)} : R \to [p], a \mapsto a_{ij} = \mathrm{coeff}(a)_{i,j}$$

diejenige Abbildung bezeichnen, welche das Ringelement $a \in R$ auf seinen eindeutigen Koeffizienten $a_{ij} \in [p]$ der (ξ, θ)-adischen Entwicklung $a = \sum_{i=0}^{r-1}\sum_{j=0}^{m-1} a_{ij}\xi^{r-1-i}\theta^j$ abbildet.

Häufig werden wir die Matrix $\mathrm{coeff}(a) \in [p]^{r\times m}$ der (ξ, θ)-adischen Entwicklung von $a \in R$ auch über dem Körper $\mathbb{F}_p$ auffassen. Statt $\overline{\mathrm{coeff}(a)}$ schreiben wir dann auch $\overline{\mathrm{coeff}}(a)$. Gleiches gilt für die Koeffizienten der θ-adischen Entwicklung $\mathrm{coeff}^{(i)}(a)$ bezüglich dem Körper $\mathbb{F}_q$.

4.1.10 Bemerkung. Bei der Definition einer Totalordnung auf R wird sich die zunächst kontraintuitive Zuordnung des Koeffizienten a_{ij} zu $\xi^{r-1-i}\theta^j$ als vorteilhaft erweisen.

4.1.11 Beispiel. Es ist $R = \mathbb{F}_4[X]/(X^2)$ ein Kettenring der Kettenlänge $m = 2$ mit Teichmüller-Menge $T := \{0, 1, \xi, \xi^2\} = \mathbb{F}_4$ und $\theta := X$. Das Ringelement $1 + \xi^2 X \in R$ hat die (ξ, X)-adische Entwicklung $1 + \xi^2 X = 0 \cdot \xi^1 X^0 + 1 \cdot \xi^0 X^0 + 1 \cdot \xi^1 X^1 + 1 \cdot \xi^0 X^1$.

4.1.12 Folgerung. *Für beliebiges $a \in R$ und $j \leq \min\{\mathrm{ht}(a), m-1\}$ ist*

$$\overline{\mathrm{coeff}}^{(j)}(a) = \sum_{i=0}^{r-1} \overline{\mathrm{coeff}}^{(i,j)}(a)\overline{\xi}^{r-1-i}.$$

Beweis. Folgt sofort aus dem Beweis zur Existenz der (ξ, θ)-adischen Entwicklung von $a \in R$. □

4.1.13 Bemerkung. Die Teichmüller-Menge T ist im Allgemeinen nicht additiv abgeschlossen. Es ist somit für beliebiges $a \in R$ und $j \in [m]$ mit $j > \mathrm{ht}(a)$ im Allgemeinen

$$\overline{\mathrm{coeff}}^{(j)}(a) \neq \sum_{i=0}^{r-1} \overline{\mathrm{coeff}}^{(i,j)}(a)\overline{\xi}^{r-1-i}.$$

Eine ähnliche Aussage gilt für Summen von Ringelementen:

4.1.14 Hilfssatz. *Es seien $a, b \in R$ beliebig und $j \in [m] : j \leq \max\{\mathrm{ht}(a), \mathrm{ht}(b)\}$. Dann gilt:*

$$\overline{\mathrm{coeff}}^{(j)}(a+b) = \overline{\mathrm{coeff}}^{(j)}(a) + \overline{\mathrm{coeff}}^{(j)}(b)$$

und

$$\overline{\mathrm{coeff}}^{(i,j)}(a+b) = \overline{\mathrm{coeff}}^{(i,j)}(a) + \overline{\mathrm{coeff}}^{(i,j)}(b),\ \forall\ i \in [r]$$

Beweis. Folgt sofort aus den θ-adischen bzw. (ξ, θ)-adischen Entwicklungen von a, b und $a + b$. □

Wir nutzen nun die Bijektion von R nach $[p]^{r\times m}$ um ausgehend von einer Totalordnung auf $[p]^{r\times m}$ eine Totalordnung auf R zu definieren. Hierzu lesen wir die Matrizen spaltenweise und vergleichen die resultierenden Vektoren lexikographisch:

4.1.15 Definition (Totalordnung auf R)**.** Für alle $a, b \in R$ sei

$$a \leq b :\Longleftrightarrow \mathrm{coeff}(a) \leq \mathrm{coeff}(b)$$

4.1.16 Beispiel. Die Totalordnung auf dem Kettenring $R := \mathbb{F}_4[X]/(X^2)$ bestimmt sich wie in Tabelle 4.1 angegeben über die letzte Tabellenspalte.

Die Abbildungen $\mathrm{coeff}^{(i,j)}$ erlauben es uns nicht nur, eine Totalordnung auf R zu definieren, sondern auch die Struktur der Gruppen $(R, +)$ und $(R^*, \cdot)$ genauer zu untersuchen. Hierzu definieren wir zunächst zu $i \in [r+1]$ und $j \in [m]$ die Mengen

$$\begin{aligned} R^{(i,j)} &:= \left\{ a \in \mathrm{Rad}(R)^j \mid \forall\ 0 \leq \nu < i : \mathrm{coeff}^{(\nu,j)}(a) = 0 \right\} \\ &= \left\{ \sum_{\nu=i}^{r-1} a_{\nu,j}\xi^{r-1-\nu}\theta^j \mid a_{\nu,j} \in [p],\ \forall\ i \leq \nu \leq r-1 \right\} + \mathrm{Rad}(R)^{j+1}. \end{aligned}$$

4.1.17 Folgerung. *Für beliebiges $i \in [r]$ und $j \in [m]$ gilt:*

- $\forall\ a \in \mathrm{Rad}(R)^{j+1},\ \forall\ b \in \mathrm{Rad}(R)^j \setminus \mathrm{Rad}(R)^{j+1} : a < \theta^j \leq b$
- $R^{(i+1,j)} \subset R^{(i,j)}$ *und* $|R^{(i,j)}| = p^{r(m-j)-i}$
- $\forall\ a \in R^{(i+1,j)},\ \forall\ b \in R^{(i,j)} \setminus R^{(i+1,j)} : a < \xi^{r-1-i}\theta^j \leq b$

X-adische Entwicklung von $a \in R$	(ξ, X)-adische Entwicklung von a	coeff(a)
0	0	$\left(\begin{smallmatrix}0&0\\0&0\end{smallmatrix}\right)$
X	$1 \cdot X$	$\left(\begin{smallmatrix}0&0\\0&1\end{smallmatrix}\right)$
ξX	$1 \cdot \xi X + 0 \cdot X$	$\left(\begin{smallmatrix}0&1\\0&0\end{smallmatrix}\right)$
$\xi^2 X$	$1 \cdot \xi X + 1 \cdot X$	$\left(\begin{smallmatrix}0&1\\0&1\end{smallmatrix}\right)$
1	$1 \cdot 1 + 0 \cdot \xi X + 0 \cdot X$	$\left(\begin{smallmatrix}0&0\\1&0\end{smallmatrix}\right)$
$1 + X$	$1 \cdot 1 + 0 \cdot \xi X + 1 \cdot X$	$\left(\begin{smallmatrix}0&0\\1&1\end{smallmatrix}\right)$
$\vdots$	$\vdots$	$\vdots$
$\xi^2 + \xi^2 X$	$1 \cdot \xi + 1 \cdot 1 + 1 \cdot \xi X + 1 \cdot X$	$\left(\begin{smallmatrix}1&1\\1&1\end{smallmatrix}\right)$

Tabelle 4.1.: Totalordnung auf $\mathbb{F}_4[X]/(X^2)$

Beweis. Anschaulich bedeutet der Übergang von $R^{(i+1,j)}$ nach $R^{(i,j)}$ die Freiheit einen weiteren Eintrag der Matrix coeff(a) beliebig zu belegen. Dabei wird analog zur Definition der Totalordnung auf R vorgegangen. An den Übergängen $R^{(0,j+1)}$ zu $R^{(r,j)}$ nutzt man die Gleichheit der Mengen $R^{(0,j+1)} = \mathrm{Rad}(R)^{j+1} = R^{(r,j)}$. □

4.1.18 Hilfssatz. *Für jedes $i \in [r]$ und $j \in [m]$ ist die Teilmenge $R^{(i,j)}$ eine Untergruppe der abelschen Gruppe $(R, +)$ und*

$$R = R^{(0,0)} \rhd \ldots \rhd R^{(r,0)} = R^{(0,1)} \rhd \ldots \rhd R^{(r,m-1)} = \{0_R\}$$

ist eine Kompositionsreihe von $(R, +)$, bei welcher jeder Faktor $R^{(i,j)}/R^{(i+1,j)}, i \in [r]$ isomorph zu $\mathbb{Z}_p$ ist.

Beweis. Wir zeigen die Behauptung für ein fest gewähltes $j \in [m]$ über eine Induktion nach $i \in [r]$. Offensichtlich sind die Mengen $R^{(0,j)} = \mathrm{Rad}(R)^j$ Untergruppen und wegen der Kommutativität auch Normalteiler. Die Abbildung

$$\overline{\mathrm{coeff}}^{(i,j)} : R^{(i,j)} \to \mathbb{Z}_p, \quad a \mapsto \overline{\mathrm{coeff}}^{(i,j)}(a)$$

definiert einen Epimorphismus, dessen Kern gleich $R^{(i+1,j)}$ ist. □

Auf gleiche Weise gewinnen wir eine Normalreihe von $(R^*, \cdot)$ mit zyklischen Faktoren. Im weiteren Verlauf dieser Arbeit werden wir die zu den Faktorgruppen isomorphen Gruppen jeweils über dem Normalteilerzeichen notieren:

4.1.19 Hilfssatz. *Für $i \in [r+1]$ und $j \in [m] : j \geq 1$ sei $R^{*(i,j)} := 1 + R^{(i,j)}$. Dann bilden die Mengen $R^{*(i,j)}$ Untergruppen von $(R^*, \cdot)$ und*

$$R^* \overset{\mathbb{F}_q^*}{\trianglerighteq} 1 + \mathrm{Rad}(R) = R^{*(0,1)} \overset{\mathbb{Z}_p}{\rhd} \ldots \overset{\mathbb{Z}_p}{\rhd} R^{*(r,1)} = R^{*(0,2)} \overset{\mathbb{Z}_p}{\rhd} \ldots \overset{\mathbb{Z}_p}{\rhd} R^{*(r,m-1)} = \{1_R\}$$

ist eine Normalreihe von R^* mit zyklischen Faktoren.

Beweis. Zunächst zeigen wir, dass für jedes $i \in [r]$ und $j \in [m]$ die Mengen $R^{*(i,j)}$ multiplikativ abgeschlossen sind. Dazu wählen wir $a, b \in R^{(i,j)}$ beliebig. Es gilt:

$$(1+a)(1+b) = 1 + \underbrace{(a+b)}_{\in R^{(i,j)}} + \underbrace{ab}_{\in \mathrm{Rad}(R)^{2j} \subseteq \mathrm{Rad}(R)^{j+1}} \in 1 + R^{(i,j)} = R^{*(i,j)}$$

Der erste Normalteiler $1 + \mathrm{Rad}(R) \lhd R^*$ der Kette ergibt sich sofort aus dem Gruppenepimorphismus $^- \colon R^* \to \mathbb{F}_q^*, a \mapsto \overline{a}$. Die Normalteilereigenschaft der Untergruppen $R^{*(i+1,j)}$ in $R^{*(i,j)}$ mit $i \in [r]$ und $j \in [m] : j \geq 1$ erhält man aus dem Epimorphismus $\overline{\mathrm{coeff}}^{(i,j)} : R^{*(i,j)} \to \mathbb{Z}_p$, dessen Kern gleich $R^{*(i+1,j)}$ ist. □

Kettenringe R mit Verzweigungsindex $\varepsilon = 1$ nennen wir *Galois-Ringe*. Das Jacobson-Radikal $\mathrm{Rad}(R) = pR$ wird also von einer Primzahl p erzeugt. Wir vereinbaren, für Galois-Ringe stets den Erzeuger $\theta = p$ zu wählen.

4.1.20 Fakt ([59, Theorem 4.2])**.** *Galois-Ringe sind stets kommutativ. Zu einer beliebigen Primzahl p und natürlichen Zahlen $m, r > 0$ existiert ein bis auf Isomorphie eindeutiger Galois-Ring* $\mathrm{GR}(p^m, r)$ *der Charakteristik p^m und Kardinalität p^{rm}. Zur Konstruktion des Galois-Ring* $\mathrm{GR}(p^m, r) = \mathbb{Z}_{p^m}[X]/(f)$ *wählt man ein normiertes Polynom*[1] *$f \in \mathbb{Z}_{p^m}[X]$ vom Grad r, dessen Bild modulo p in $\mathbb{Z}_p[X]$ irreduzibel ist.*

4.1.21 Beispiel. Der endliche Körper $\mathbb{F}_{p^r}$ ist isomorph zum Galois-Ring $\mathrm{GR}(p, r)$. Der Kettenring $\mathbb{Z}_{p^m}$ ist isomorph zu dem Galois-Ring $\mathrm{GR}(p^m, 1)$.

Die Galois-Ringe sind nicht nur wegen ihrer Eindeutigkeit und Konstruktion den endlichen Körpern sehr verwandt:

4.1.22 Fakt ([59, Theorem 4.5])**.** *Die Automorphismengruppe* $\mathrm{Aut}(\mathrm{GR}(p^m, r))$ *eines Galois-Rings ist zyklisch und von der Ordnung r. Sie wird von dem* Frobenius-Automorphismus $\tau \in \mathrm{Aut}(\mathrm{GR}(p^m, r))$ *erzeugt, der die Elemente $a \in \mathrm{GR}(p^m, r)$ in p-adischer Entwicklung $a = \sum_{i=0}^{m-1} a_i p^i$ auf $\tau(a) := \sum_{i=0}^{m-1} a_i^p p^i$ abbildet.*

4.1.23 Bemerkung. Da die Abbildung $\varphi : \mathrm{Aut}(\mathrm{GR}(p^m, r)) \to \mathrm{Aut}(\mathbb{F}_{p^r})$ mit $\overline{\sigma}(\overline{a}) := \varphi(\sigma)(\overline{a}) := \overline{\sigma(a)}$ für alle $\sigma \in \mathrm{Aut}(\mathrm{GR}(p^m, r))$ und $a \in R$ einen kanonischen Isomorphismus zwischen beiden Automorphismengruppen definiert, werden wir im Folgenden auch den Frobenius-Automorphismus des Körpers $\mathbb{F}_{p^r}$ mit dem gleichen Symbol τ bezeichnen.

[1] In [73, Tabelle 1.1] gibt J. Zwanzger eine Liste solcher Polynome. Sie bilden überdies eine Verallgemeinerung der sogenannten Conway-Polynome über endlichen Körpern, vergleiche [64]. Sie sind ebenfalls für gegebenes q und m eindeutig bestimmt und können somit zur standardisierten Darstellung der Elemente eines Galois-Rings herangezogen werden. Überdies gewährleisten sie eine gewisse Kompatibilität in der Darstellung des Galois-Rings und seinen Galois-Unterringen.

Es sei p^{m_S} die Charakteristik des endlichen Kettenrings R. Nach [12, 59] besitzt R einen bis auf Isomorphie eindeutig bestimmten Galois-Unterring $S \simeq \mathrm{GR}(p^{m_S}, r) \subseteq R$ mit $\overline{S} = \overline{R} = \mathbb{F}_{p^r}$. Man nennt den Ring S auch einen Koeffizientenring zu R, da man als eine Teichmüller-Menge von R auch die Teichmüller-Menge T von S wählen kann[2]. Damit zeigt man aber auch leicht über Konjugation, dass das Ringerzeugnis einer jeden Teichmüller-Menge T von R einen solchen Unterring $S \simeq \mathrm{GR}(p^{m_S}, r)$ von R ergibt. Es liegen also ohne Beschränkung der Allgemeinheit alle Koeffizienten der θ-adischen Entwicklung eines Elements $a \in R$ in dem Galois-Unterring S. Ist ε der Verzweigungsindex zu R, so gilt somit für eine beliebige Teichmüller-Menge T von R und $a, b, c \in T$:

$$\overline{a+b} = \overline{c} \Longleftrightarrow a + b - c \in \mathrm{Rad}(R) \cap S = Sp \subseteq Rp = \mathrm{Rad}(R)^{\varepsilon}$$

Wir benötigen noch einige weitere Eigenschaft der endlichen Kettenringe, bevor wir zu der Kanonisierung der linearen Codes übergehen können. Die nachfolgende Diskussion folgt dem Übersichtsartikel [59]. Die Beweise für unsere Aussagen finden sich in [58]. Unsere Ausführungen werden bis zu der Tatsache vordringen, dass sich jeder endliche Kettenring als Quotient eines Schiefpolynomrings über einem beliebigen Koeffizientenring darstellen lässt.

4.1.24 Fakt ([59, Proposition 5.17])**.** *Es sei R ein endlicher Kettenring und $S \simeq \mathrm{GR}(p^{m_S}, r)$ ein Koeffizientenring von R. Dann gibt es einen Erzeuger θ von $\mathrm{Rad}(R)$ und einen ausgezeichneten Automorphismus $\tau^e \in \mathrm{Aut}(S)$, so dass die Ringelemente $a \in S$ über die Vorschrift $\theta a = \tau^e(a)\theta$ mit θ vertauschen.*

Wir vereinbaren also weiter, dass zu dem gegebenen Kettenring R mit Koeffizientenring S der Erzeuger θ und der Ringautomorphismus $\tau^e \in \mathrm{Aut}(S)$ durch Fakt 4.1.24 bestimmt sind. Dann lässt sich auch über die θ-adische Entwicklung eines Ringelements $a = \sum_{i=0}^{m-1} a_i\theta^i$ mit $a_i \in T \subseteq S$ die Multiplikation von a mit θ von links leicht beschreiben:

$$\theta a = \theta \cdot \sum_{i=0}^{m-1} a_i\theta^i = \left(\sum_{i=0}^{m-1} \tau^e(a_i)\theta^i\right) \cdot \theta$$

Wir schließen mit einer Charakterisierung der endliche Kettenringe ohne die auftretenden Begriffe exakt zu definieren:

4.1.25 Fakt ([59, Theorem 5.18])**.** *Zu jedem endlichen Kettenring R mit Koeffizientenring S existiert ein spezielles Eisensteinpolynom $f \in S[X;\tau^e]$ über dem Schiefpolynomring $S[X;\tau^e]$, so dass $R \simeq S[X;\tau^e]/\left(f, p^{m_S-1}X^{m_R-(m_S-1)\epsilon}\right)$.*

Weitergehende Informationen findet man in [59], insbesondere auch zu der Definition eines spezielles Eisensteinpolynoms und den weiteren Einschränkungen, die man zum Beispiel an den Grad von f machen muss. Eine Klassifikation der endlichen Kettenringe bis auf Isomorphie ist immer noch offen.

[2] Dies ist sofort ersichtlich, da die Teichmüller-Gruppen genau die multiplikativ abgeschlossene Teilmengen von R bzw. S mit $q-1$ Elementen sind.

4.2. Automorphismen

In diesem Abschnitt soll die Struktur der Automorphismengruppe $\mathrm{Aut}(R)$ des Kettenrings R untersucht werden. Ohne Beschränkung der Allgemeinheit nehmen wir hierbei an, dass der Kettenring ein nichttriviales Jacobson-Radikal besitzt. Andernfalls handelt es sich um einen endlichen Körper, dessen Automorphismengruppe wohlbekannt ist. Eine allgemeine Beschreibung der Automorphismengruppe eines beliebigen Kettenrings ist noch nicht bekannt. Teilresultate finden sich aber in [1] und [68].

Wir folgen an dieser Stelle zunächst wieder den Ausführungen aus [73, Kapitel 5.1]. Wir werden die dort erzielten Resultate verfeinern, um ein Erzeugendensystem der Automorphismengruppe $\mathrm{Aut}(R)$ zu berechnen. Einen Algorithmus A.1, welcher diese Berechnungen durchführt, geben wir im Anhang A.

Es sei S ein zu R isomorpher Kettenring und $\psi \in S^*$ ein beliebiger Erzeuger einer Teichmüller-Gruppe von S und $\omega \in \mathrm{Rad}(S) \setminus \mathrm{Rad}(S)^2$ ein Erzeuger des Jacobson-Radikals. Über die eindeutige θ-adische Entwicklung $a = \sum_{i=0}^{m-1} a_i\theta^i$ eines beliebigen Ringelements $a \in R$ lässt sich eine Bijektion

$$\chi_\psi^\omega : R \to S, \quad \sum_{i=0}^{m-1} a_i\theta^i \mapsto \sum_{i=0}^{m-1} b_i\omega^i \text{ mit } b_i := \begin{cases} \psi^{j_i}, & \text{falls } a_i = \xi^{j_i} \text{ für ein } j_i \in [q-1] \\ 0, & \text{falls } a_i = 0 \end{cases}$$

von R nach S erklären.

4.2.1 Fakt. *Jeder Ringisomorphismus $\alpha : R \to S$ ist bereits über die Angabe von $\alpha(\xi)$ und $\alpha(\theta)$ eindeutig bestimmt und es ist $\alpha = \chi_{\alpha(\xi)}^{\alpha(\theta)}$. Jedoch definiert nicht jede Bijektion χ_ψ^ω mit einem beliebigen Erzeuger $\psi \in S^*$ einer Teichmüller-Gruppe von S und $\omega \in \mathrm{Rad}(S) \setminus \mathrm{Rad}(S)^2$ auch einen Ringhomomorphismus.*

4.2.2 Beispiel. Im Kettenring $\mathbb{Z}_9$ können wir $\theta = 3$ und $\omega = 6$ setzen. Dies definiert eine Bijektion auf $\mathbb{Z}_9$ jedoch keinen Ringautomorphismus, denn es gibt keine nicht-trivialen Automorphismen von $\mathbb{Z}_9$.

Obiges Beispiel zeigt, dass wir im Allgemeinen eine größere Wahlfreiheit für die Festlegung von ψ und ω haben, als wir sie durch die Menge aller Ringisomorphismen abdecken könnten.

4.2.3 Folgerung. *Es sei $\alpha : R \to S$ ein Ringisomorphismus. Der Ring R ist über die (ξ, θ)-adische Entwicklung seiner Elemente totalgeordnet. Genauso definiert auch die $(\alpha(\xi), \alpha(\theta))$-adische Entwicklung der Elemente in S eine Totalordnung $\preceq$ auf S. Bezüglich dieser Ordnungen ist α ordnungserhaltend.*

Beweis. Es gilt zunächst

$$\alpha(a) = \alpha\left((\xi^{r-1}, \ldots, \xi^0) \cdot \mathrm{coeff}(a) \cdot \begin{pmatrix} \theta^0 \\ \vdots \\ \theta^{m-1} \end{pmatrix}\right)$$

$$= (\alpha(\xi)^{r-1}, \ldots, \alpha(\xi)^0) \cdot \mathrm{coeff}(a) \cdot \begin{pmatrix} \alpha(\theta)^0 \\ \vdots \\ \alpha(\theta)^{m-1} \end{pmatrix}$$

Da die $(\alpha(\xi), \alpha(\theta))$-adische Entwicklung eindeutig ist, bleiben somit die Koeffizienten beim Übergang nach S unverändert. Sind nun $a, b \in R$ beliebig, dann gilt:

$$a \leq b \Longleftrightarrow \mathrm{coeff}(a) \leq \mathrm{coeff}(b) \Longleftrightarrow \alpha(a) \preceq \alpha(b) \qquad \square$$

Wir werden nun die Struktur der Automorphismengruppe näher untersuchen. Für jede Einheit $a \in R^*$ definiert die Konjugation mit a, d.h. die Abbildung $\chi_{a\xi a^{-1}}^{a\theta a^{-1}}$, einen Automorphismus von R. Die Konjugation mit Einheiten

$$\Psi : R^* \to \mathrm{Aut}(R), \quad a \mapsto \chi_{a\xi a^{-1}}^{a\theta a^{-1}}$$

definiert bekanntlich einen Gruppenhomomorphismus. Wir nennen einen Ringautomorphismus $\alpha \in \Psi(R^*) =: \mathrm{Inn}(R)$ im Bild von Ψ einen *inneren Automorphismus*. Die Einheiten $a \in R^*$ mit $\Psi(a) = \alpha$ sind also nur bis auf Multiplikation mit dem Kern $Z(R) := \Psi^{-1}(\mathrm{id}_R) = \{a \in R^* \mid \forall\, b \in R : ab = ba\} \trianglelefteq R^*$ von Ψ eindeutig bestimmt. Wir sagen dann auch, die Einheit a induziert den inneren Ringautomorphismus α. Die Einheiten $b \in Z(R)$ nennen wir auch *zentral*, um anzudeuten, dass diese auch im Zentrum von R liegen.

Die Untergruppe $\mathrm{Inn}(R)$ ist bekanntlich ein Normalteiler der Automorphismengruppe $\mathrm{Aut}(R)$. Weiter seien mit

$$\mathrm{Aut}_T := \mathrm{Stab}_{\mathrm{Aut}(R)}(T) := \{\alpha \in \mathrm{Aut}(R) \mid \alpha(\xi) \in T\}$$

und

$$\mathrm{Aut}_\xi := \mathrm{Stab}_{\mathrm{Aut}(R)}(\xi) := \{\alpha \in \mathrm{Aut}(R) \mid \alpha(\xi) = \xi\}$$

diejenigen Untergruppen der Automorphismengruppe bezeichnet, welche die Teichmüller-Menge T von R beziehungsweise den fest vorgeschriebenen Teichmüller-Erzeuger $\xi \in R$ fix lassen.

Unter einem Ringautomorphismus wird die Teichmüller-Menge T wieder auf eine Teichmüller-Menge von R abgebildet. Da diese aber durch Konjugation – also über die Anwendung eines inneren Automorphismus – auseinander hervorgehen, lässt sich nachfolgende Aussage leicht beweisen:

4.2.4 Fakt ([73], Lemma 5.3)**.** *Es ist* $\mathrm{Aut}(R) = \mathrm{Inn}(R) \circ \mathrm{Aut}_T$*, d.h. zu beliebigen* $\alpha \in \mathrm{Aut}(R)$ *gibt es einen inneren Automorphismus* $\rho \in \mathrm{Inn}(R)$ *und* $\sigma \in \mathrm{Aut}_T$ *mit* $\alpha = \rho \circ \sigma$*.*

Es wird sich in Abschnitt 5.1.1 zeigen, dass die inneren Automorphismen bei der Modellierung der Äquivalenz linearer Codes als Gruppenoperation eine vernachlässigbare Rolle spielen werden. Wir werden uns dort, mit Hilfe einer geeigneten Umformulierung des Problems, auf die Faktorgruppe $\mathrm{Out}(R) := \mathrm{Aut}(R)/\mathrm{Inn}(R)$ der sogenannten *äußeren Automorphismen* zurückziehen können.

4.2.5 Hilfssatz. *Es gilt:*

$$\mathrm{Out}(R) \simeq \mathrm{Aut}_T\,/(\mathrm{Aut}_T \cap \mathrm{Inn}(R)) = \mathrm{Aut}_T\,/(\mathrm{Aut}_\xi \cap \mathrm{Inn}(R))$$

Beweis. Die Isomorphie der Gruppen $\mathrm{Out}(R)$ und $\mathrm{Aut}_T\,/(\mathrm{Aut}_T \cap \mathrm{Inn}(R))$ ergibt sich sofort aus Fakt 4.2.4 und dem ersten Isomorphiesatz für Gruppen. Die Gleichheit der Gruppen $(\mathrm{Aut}_T \cap \mathrm{Inn}(R)) = (\mathrm{Aut}_\xi \cap \mathrm{Inn}(R))$ sieht man wie folgt ein: Es sei $\alpha \in \mathrm{Aut}_T$ ein inneren Automorphismus, welcher von einer Einheit $a \in R^*$ induziert wird. Da nach der Voraussetzung $\alpha(\xi)$ ebenfalls ein Element der Teichmüller-Menge T ist und aber $\overline{\xi} = \overline{a\xi a^{-1}}$ gilt, folgt sofort $\xi = a\xi a^{-1} = \alpha(\xi)$. □

Wir werden unsere Untersuchungen also auf die Untergruppe Aut_T einschränken und eine ähnliche Strukturaussage für diese Gruppe herleiten, wie wir sie auch für die multiplikative Gruppe R^* beobachten konnten. Anschließend behandeln wir dann den Normalteiler $\mathrm{Inn}_\xi := \mathrm{Aut}_\xi \cap \mathrm{Inn}(R)$.

4.2.6 Hilfssatz. *Es ist* $\mathrm{Aut}_\xi \trianglelefteq \mathrm{Aut}_T$ *und* $\mathrm{Aut}_T\,/\,\mathrm{Aut}_\xi$ *isomorph zu einer zyklischen Gruppe von einer Ordnung* $r' \mid r$.

Beweis. Es sei $\alpha \in \mathrm{Aut}_\xi$ und $\beta \in \mathrm{Aut}_T$ beliebig. Aus [73, Lemma 5.2] ergibt sich zunächst, dass $\beta(\xi) = \xi^{p^s}$ für ein $s \in [r]$ gilt. Damit erhält man

$$(\beta \circ \alpha \circ \beta^{-1})(\xi) = (\beta \circ \alpha)(\xi^{p^{r-s}}) = \beta(\alpha(\xi)^{p^{r-s}}) = \left(\xi^{p^{r-s}}\right)^{p^s} = \xi^q = \xi$$

und somit $(\beta\circ\alpha\circ\beta^{-1}) \in \mathrm{Aut}_\xi$. Die zweite Aussage wird in [73, Lemma 5.4] bewiesen. □

4.2.7 Hilfssatz. *Es seien* $\alpha, \beta \in \mathrm{Aut}_\xi$ *Automorphismen von* R *mit*

$$1 < \mathrm{ht}(\alpha(\theta) - \theta) =: i \leq j := \mathrm{ht}(\beta(\theta) - \theta) \leq m.$$

Weiter seien $\alpha(\theta) - \theta =: \sum_{\ell=i}^{m-1} a_\ell \theta^\ell$ *und* $\beta(\theta) - \theta =: \sum_{\ell=j}^{m-1} b_\ell \theta^\ell$ *die* θ*-adischen Entwicklungen. Dann gilt:*

$$(\alpha \circ \beta)(\theta) \equiv \theta + \sum_{\ell=i}^{j-1} a_\ell \theta^\ell + (a_j + b_j)\theta^j \mod \mathrm{Rad}(R)^{j+1}$$

Beweis. Zunächst rechnet man leicht nach, dass

$$\alpha(\theta^j) + \mathrm{Rad}(R)^{j+1} = \left(\theta + \sum_{\ell=i}^{m-1} a_\ell \theta^\ell\right)^j + \mathrm{Rad}(R)^{j+1} = \theta^j + \mathrm{Rad}(R)^{j+1}$$

gilt. Dies können wir nun zum Beweis der Aussage wie folgt ausnutzen:

$$\begin{aligned}(\alpha\circ\beta)(\theta) = \ & \alpha\left(\theta+\sum_{\ell=j}^{m-1} b_\ell\theta^\ell\right) = \alpha(\theta)+\sum_{\ell=j}^{m-1}\alpha(b_\ell\theta^\ell)\\ \equiv \ & \theta+\sum_{\ell=i}^{j} a_\ell\theta^\ell+\alpha(b_j)\alpha(\theta^j)\equiv\theta+\sum_{\ell=i}^{j} a_\ell\theta^\ell+b_j\theta^j\\ \equiv\ & \theta+\sum_{\ell=i}^{j-1} a_\ell\theta^\ell+(a_j+b_j)\theta^j \qquad \mod \operatorname{Rad}(R)^{j+1}\end{aligned}$$ □

4.2.8 Hilfssatz. *Für $i\in[r+1]$ und $j\in[m]: j\geq 2$ sei*

$$\operatorname{Aut}_\xi^{(i,j)} := \{\alpha\in\operatorname{Aut}_\xi \mid (\alpha(\theta)-\theta)\in R^{(i,j)}\}.$$

Dann ist für jedes $i<r$ die Menge $\operatorname{Aut}_\xi^{(i+1,j)}$ ein Normalteiler in $\operatorname{Aut}_\xi^{(i,j)}$ und es gilt entweder $\operatorname{Aut}_\xi^{(i,j)}=\operatorname{Aut}_\xi^{(i+1,j)}$ oder es ist $\operatorname{Aut}_\xi^{(i,j)}/\operatorname{Aut}_\xi^{(i+1,j)}\simeq\mathbb{Z}_p$. Somit definiert

$$\operatorname{Aut}_\xi \overset{U}{\trianglerighteq} \operatorname{Aut}_\xi^{(0,2)} \trianglerighteq \ldots \trianglerighteq \operatorname{Aut}_\xi^{(r,2)} = \operatorname{Aut}_\xi^{(0,3)} \trianglerighteq \ldots \trianglerighteq \operatorname{Aut}_\xi^{(r,m-1)} = \{\mathrm{id}_R\}$$

eine Normalreihe von Aut_ξ mit einem geeigneten $U\leq\mathbb{F}_q^*$.

Beweis. Wir beweisen die Aussage mit ähnlichen Mitteln, welche wir auch in Hilfssatz 4.1.19 angewandt haben. Zunächst zeigen wir, dass die Menge $\operatorname{Aut}_\xi^{(0,2)}$ ein Normalteiler von Aut_ξ ist. Hierzu beweisen wir, dass die Abbildung

$$\operatorname{Aut}_\xi\to\mathbb{F}_q^*,\quad \alpha\mapsto\overline{\operatorname{coeff}}^{(1)}(\alpha(\theta))$$

einen Gruppenhomomorphismus definiert. Es gilt nämlich

$$\begin{aligned}\overline{\operatorname{coeff}}^{(1)}(\alpha\beta(\theta)) &= \overline{\operatorname{coeff}}^{(1)}\Big(\alpha\Big(\operatorname{coeff}^{(1)}(\beta(\theta))\cdot\theta\Big)\Big) = \overline{\operatorname{coeff}}^{(1)}\Big(\operatorname{coeff}^{(1)}(\beta(\theta))\cdot\alpha(\theta)\Big)\\ &= \overline{\operatorname{coeff}}^{(1)}(\beta(\theta))\cdot\overline{\operatorname{coeff}}^{(1)}(\alpha(\theta)) = \overline{\operatorname{coeff}}^{(1)}(\alpha(\theta))\cdot\overline{\operatorname{coeff}}^{(1)}(\beta(\theta)).\end{aligned}$$

Genauso zeigt man induktiv mit der Aussage aus dem vorausgegangenen Hilfssatz, dass für $i\in[r]$ und $j\in[m]: j\geq 2$ die Abbildung

$$\operatorname{Aut}_\xi^{(i,j)}\to R^{(i,j)}/R^{(i+1,j)}\simeq(\mathbb{Z}_p,+),\quad \alpha\mapsto\alpha(\theta)-\theta+R^{(i+1,j)}$$

ebenfalls einen Gruppenhomomorphismus definiert, dessen Kern gleich $\operatorname{Aut}_\xi^{(i+1,j)}$ ist. □

Damit haben wir schließlich für die Gruppe Aut_T ebenfalls eine Normalreihe

$$\operatorname{Aut}_T \trianglerighteq \operatorname{Aut}_\xi \trianglerighteq \operatorname{Aut}_\xi^{(0,2)} \trianglerighteq \ldots \trianglerighteq \operatorname{Aut}_\xi^{(r,2)} = \operatorname{Aut}_\xi^{(0,3)} \trianglerighteq \ldots \trianglerighteq \operatorname{Aut}_\xi^{(r,m-1)} = \{\mathrm{id}_R\} \quad (4.1)$$

mit zyklischen Faktoren hergeleitet. Mit Hilfe von Algorithmus A.1 können wir ein an diese Normalreihe angepasstes Erzeugendensystem der Gruppe Aut_T berechnen.

Für eine Beschreibung der Gruppe $\mathrm{Out}(R)$ führen wir nun noch eine Untersuchung der Gruppe $\mathrm{Inn}_\xi := \mathrm{Aut}_\xi \cap \mathrm{Inn}(R)$ durch. Für $j \in [m] : j \geq 2$ sei mit

$$\mathrm{Inn}_\xi^{(j)} := \mathrm{Aut}_\xi^{(0,j)} \cap \mathrm{Inn}(R) = \left\{\alpha \in \mathrm{Inn}(R) \mid \alpha(\xi) = \xi \wedge (\alpha(\theta) - \theta) \in \mathrm{Rad}(R)^j\right\}$$

der entsprechende Schnitt mit den Untergruppen $\mathrm{Aut}_\xi^{(0,j)}$ bezeichnet. Damit haben wir auch die Gruppenhomomorphismen aus dem Beweis zu Hilfssatz 4.2.8 zur Verfügung und wir erhalten die Normalreihe

$$\mathrm{Inn}_\xi^{(1)} := \mathrm{Inn}_\xi \trianglerighteq \mathrm{Inn}_\xi^{(2)} \trianglerighteq \ldots \trianglerighteq \mathrm{Inn}_\xi^{(m-1)}.$$

Wir werden als ersten Schritt den Zentralisator $Z(\xi) := \{a \in R^* \mid a\xi a^{-1} = \xi\} \leq R^*$ von ξ untersuchen. Wir können diese Untergruppe auch als das Urbild von Inn_ξ unter dem Gruppenhomomorphismus

$$\Psi : R^* \to \mathrm{Inn}(R), \quad a \mapsto \chi_{a\xi a^{-1}}^{a\theta a^{-1}}$$

interpretieren. Damit ist $\mathrm{Inn}_\xi \simeq Z(\xi)/Z(R)$. Wir werden dann im Folgenden beweisen, dass für $j \in [m], j \geq 2$ die Gruppen $\mathrm{Inn}_\xi^{(j)}$ und $\Psi\Big(Z(\xi) \cap (1 + \mathrm{Rad}(R)^{j-1})\Big)$ identisch sind.

4.2.9 Hilfssatz. *Es sei $\chi_\xi^{a\theta} \in \mathrm{Inn}_\xi$ ein innerer Automorphismus für ein $a \in R^*$. Dann gilt:*

$$\mathrm{coeff}^{(1)}(\alpha(\theta)) = \mathrm{coeff}^{(0)}(a) \in T_0 := \left\{t^{1-p^e} \mid t \in T^*\right\} = \langle \xi^{1-p^e} \rangle \leq (T^*, \cdot).$$

Umgekehrt induziert ein beliebiges Element $t \in T^*$ der Teichmüller-Menge den inneren Ringautomorphismus $\chi_\xi^{t^{1-p^e}\theta}$.

Beweis. Es sei $b = t + b_1\theta \in R^*$ eine Einheit mit $t \in T^*$ und $b_1 \in R$, welche den inneren Ringautomorphismus $\chi_\xi^{a\theta}$ induziert. Dann ist $b^{-1} = t^{-1} + b_2\theta$ für ein $b_2 \in R$ und es gilt:

$$a\theta \equiv t\theta t^{-1} = t\tau^e(t^{-1})\theta = t^{1-p^e}\theta \mod \mathrm{Rad}(R)^2$$

Wir schließen hieraus $\mathrm{coeff}^{(0)}(a) = t^{1-p^e}$ und damit, dass die erste Behauptung gilt. Die zweite ergibt sich durch simples Nachrechnen. □

Ist nun $\mathbb{F}_{p^s} \leq \mathbb{F}_q$ der Fixkörper von τ^e, also $s = \mathrm{ggT}(r, e)$, so ist

$$\mathrm{Inn}_\xi \,/\, \mathrm{Inn}_\xi^{(2)} \simeq T_0 \simeq \overline{T_0} \simeq \mathbb{F}_q^*/\mathbb{F}_{p^s}^*$$

und somit isomorph zu einer zyklischen Gruppe der Ordnung $\frac{p^r-1}{p^s-1}$.

Über die weiteren Quotienten $\mathrm{Inn}_\xi^{(j)} \,/\, \mathrm{Inn}_\xi^{(j+1)}$ für $2 \leq j \leq m-1$ wissen wir, dass sie isomorph zu Untergruppen von $\mathbb{Z}_p^r$ sind.

4.2.10 Hilfssatz. *Es sei $a \in (1+\mathrm{Rad}(R)) \cap Z(\xi)$ und $a \neq 1$. Dann ist $\frac{r}{\mathrm{ggT}(r,e)}$ ein Teiler von $\mathrm{ht}(1-a)$.*

Beweis. Es sei $i := \mathrm{ht}(1-a) \geq 1$ und die Einheit a von der Form $a = 1 + a_0\theta^i + a_1\theta^{i+1}$ mit $a_0 \in T^*$, $a_1 \in R$. Dann folgt aus

$$\begin{aligned}\xi = a\xi a^{-1} &\equiv (1+a_0\theta^i)\xi(1+a_0\theta^i)^{-1} \\ &\equiv (\xi + a_0\tau^{ei}(\xi)\theta^i)(1-a_0\theta^i) \equiv \xi + (a_0\tau^{ei}(\xi) - \xi a_0)\theta^i \mod \mathrm{Rad}(R)^{i+1}\end{aligned}$$

bereits $(a_0\tau^{ei}(\xi) - \xi a_0) \in \mathrm{Rad}(R)$. Da sowohl $a_0\tau^{ei}(\xi)$ als auch ξa_0 in der Teichmüller-Menge liegen, ist $\overline{a_0\tau^{ei}(\xi)} = \overline{\xi a_0}$ und damit $\tau^{ei}(\overline{\xi}) = \overline{\xi}$. Somit ist r ein Teiler von ei beziehungsweise $\frac{r}{\mathrm{ggT}(r,e)}$ ein Teiler von $i = \mathrm{ht}(1-a)$. □

4.2.11 Folgerung. *Ist $x := \frac{r}{\mathrm{ggT}(r,e)}$, so wird $Z(\xi)$ von den Einheiten*

$$T^* \cup \left\{ 1 + a_i\theta^{ix} \mid a_i \in T, i \in \left[\left\lfloor \frac{m-1}{x} \right\rfloor + 1 \right] \right\}$$

erzeugt.

Beweis. Die Inklusionsrichtung „$\supseteq$“ ergibt sich durch einfaches Nachrechnen.

Ist umgekehrt $a \in Z(\xi)$ beliebig und $a = \sum_{i=0}^{m-1} a_i\theta^i$ die θ-adische Entwicklung, so liegt auch $a_0^{-1}a = 1 + \sum_{i=1}^{m-1} a_0^{-1}a_i\theta^i$ in $Z(\xi)$. Wir können also ohne Beschränkung der Allgemeinheit bereits annehmen, dass $a \in (1+\mathrm{Rad}(R)) \cap Z(\xi)$ gilt.

Für die Gruppenelemente $a \in (1+\mathrm{Rad}(R)) \cap Z(\xi)$ führen wir nun einen induktiven Beweis nach der Höhe von $1-a$. Ist $\mathrm{ht}(1-a) = m$, so ist $a = 1$ und das Ringelement a offensichtlich auch in der rechten Seite enthalten. Andernfalls ist $\mathrm{ht}(1-a) = ix$ für ein $i \in \left[\left\lfloor \frac{m-1}{x} \right\rfloor + 1\right]$. Die Einheit $1 + a_{ix}\theta^{ix}$ liegt ebenfalls in $Z(\xi)$ und damit auch das Element $\widetilde{a} = (1+a_{ix}\theta^{ix})^{-1}a$. Nun ist aber $\mathrm{ht}(1+\widetilde{a}) > ix$ und somit lässt sich $\widetilde{a}$ nach der Induktionsvoraussetzung von den Elementen der rechten Seite erzeugen. □

Aus dieser Folgerung ergeben sich nun sehr weitreichende Konsequenzen für die Gruppe Inn_ξ. Zunächst ist klar, dass jede Einheit $a \in Z^{(i-1)}(\xi) := (1+\mathrm{Rad}(R)^{i-1}) \cap Z(\xi)$ einen inneren Automorphismus $\chi_\xi^{a\theta a^{-1}} \in \mathrm{Inn}_\xi^{(i)}$ induziert. Wir werden nun im nächsten Hilfssatz zunächst nachrechnen, dass die Einschränkung

$$\Psi^{(i)} : Z^{(i-1)}(\xi) \to \mathrm{Inn}_\xi^{(i)}$$

des Homomorphismus Ψ auf die Untergruppen $Z^{(i-1)}(\xi)$ sogar surjektiv ist, also die oben angekündigte Gleichheit von $\mathrm{Inn}_\xi^{(i)}$ und $\Psi\big(Z^{(i-1)}(\xi)\big)$ gilt. Ist nun

$$\begin{aligned}Z^{(i-1)}(R) &:= \ker(\Psi^{(i)}) = \ker(\Psi) \cap Z^{(i-1)}(\xi) \\ &= Z(R) \cap \Big((1+\mathrm{Rad}(R)^{i-1}) \cap Z(\xi)\Big) = Z(R) \cap (1+\mathrm{Rad}(R)^{i-1}),\end{aligned}$$

so gilt damit auch $\mathrm{Inn}_\xi^{(i)} \simeq Z^{(i-1)}(\xi)/Z^{(i-1)}(R)$.

4.2.12 Hilfssatz. *Es sei $\chi_\xi^{\theta+a\theta^i} \in \mathrm{Inn}_\xi^{(i)}$ ein innerer Automorphismus für ein $a \in R^*$ und $i \geq 2$. Dann gibt es eine Einheit $b \in Z^{(i-1)}(\xi)$, welche $\chi_\xi^{\theta+a\theta^i}$ induziert.*

Beweis. Da $\chi_\xi^{\theta+a\theta^i}$ ein innerer Automorphismus ist, wissen wir, dass ein $c \in Z(\xi)$ existiert mit $c\theta c^{-1} = \theta + a\theta^i$. Wir nehmen zunächst an, dass $c = c_0 + c_1\theta$ für ein $c_0 \in T^*, c_0 \neq 1$ und $c_1 \in R$ sei. Dann gilt:

$$\begin{aligned} &\theta \equiv (c_0 + c_1\theta)\theta(c_0 + c_1\theta)^{-1} \equiv c_0\theta c_0^{-1} \equiv c_0\tau^e(c_0^{-1})\theta \mod \mathrm{Rad}(R)^2 \\ \iff &c_0\tau^e(c_0^{-1}) \in 1 + \mathrm{Rad}(R). \end{aligned}$$

Da aber $c_0, \tau^e(c_0^{-1})$ und auch $c_0\tau^e(c_0^{-1})$ in der Teichmüller-Menge liegen, folgt weiter

$$c_0\tau^e(c_0^{-1}) = 1 \iff c_0 = \tau^e(c_0) \iff c_0\theta = \theta c_0.$$

Das Ringelement $c_0 \in T$ kommutiert also mit allen Ringelementen und es definiert daher $c_0^{-1}c = 1 + c_0^{-1}c_1\theta \in 1 + \mathrm{Rad}(R)$ den gleichen inneren Automorphismus $\chi_\xi^{\theta+a\theta^i}$.

Wir haben hiermit gezeigt, dass der innere Automorphismus $\chi_\xi^{\theta+a\theta^i}$ bereits von einem Ringelement $c \in 1 + \mathrm{Rad}(R)$ erzeugt wird. Wir nehmen dies als Induktionsstart und zeigen für $2 \leq j \leq i-1$, dass aus der Existenz eines Elements $c \in Z^{(j-1)}(\xi)$ mit $\theta + a\theta^i = c\theta c^{-1}$ auch die Existenz eines Elements $c' \in Z^{(j)}(\xi)$ mit $\theta + a\theta^i = c'\theta c'^{-1}$ folgt: Liegt c bereits in $Z^{(j)}(\xi)$, so ist nichts zu zeigen. Es sei also $c = 1 + c_0\theta^{j-1} + c_1\theta^j$ mit $c_0 \in T^*$ und $c_1 \in R$. Aus Hilfssatz 4.2.10 schließen wir außerdem, dass $\frac{r}{\mathrm{ggT}(r,e)}$ ein Teiler von $(j-1)$ ist. Dann folgt aus

$$\begin{aligned} \theta &\equiv c\theta c^{-1} \equiv (1 + c_0\theta^{j-1} + c_1\theta^j)\theta(1 + c_0\theta^{j-1} + c_1\theta^j)^{-1} \\ &\equiv (1 + c_0\theta^{j-1})\theta(1 - c_0\theta^{j-1}) \equiv \theta + (c_0 - \tau^e(c_0))\theta^j \mod \mathrm{Rad}(R)^{j+1} \end{aligned}$$

bereits

$$(c_0 - \tau^e(c_0)) \in \mathrm{Rad}(R) \iff c_0 = \tau^e(c_0) \iff c_0\theta = \theta c_0.$$

Hierdurch ergibt sich, dass das Ringelement $(1 - c_0\theta^{j-1})$ sowohl mit ξ als auch mit θ kommutiert und also der von $(1 - c_0\theta^{j-1})$ erzeugte innere Automorphismus trivial ist. Somit ist der von $c' := (1 - c_0\theta^{j-1})c \in 1 + \mathrm{Rad}(R)^j$ erzeugte innere Automorphismus ebenfalls gleich $\chi_\xi^{\theta+a\theta^i}$. □

4.2.13 Folgerung. *Es sei $i \in [m]$, $i \geq 2$ mit $\frac{r}{\mathrm{ggT}(r,e)} \nmid (i-1)$, dann ist $\mathrm{Inn}_\xi^{(i)} = \mathrm{Inn}_\xi^{(i+1)}$, d.h. es gibt in der Menge $\mathrm{Aut}_\xi^{(0,i)} \setminus \mathrm{Aut}_\xi^{(0,i+1)}$ keine inneren Automorphismen.*

Beweis. Jede Einheit $b \in Z^{(i-1)}(\xi) = Z^{(i)}(\xi)$ induziert einen Automorphismus $\chi_\xi^{b\theta b^{-1}} \in \mathrm{Inn}_\xi^{(i+1)}$. □

4.2.14 Folgerung. *Es sei $i \in [m]$, $i \geq 2$ mit $\frac{r}{\mathrm{ggT}(r,e)} \mid (i-1)$. Dann ist*

$$\left\{\overline{\mathrm{coeff}}^{(i)}(\alpha(\theta)) \mid \alpha \in \mathrm{Inn}_\xi^{(i)}\right\} = \{a - \tau^e(a) \mid a \in \mathbb{F}_q\}$$

Beweis. Für einen Ringautomorphismus $\alpha \in \mathrm{Inn}_\xi^{(i+1)}$ ist $\mathrm{coeff}^{(i)}(\alpha(\theta)) = 0 = 1 - \tau^e(1)$. Wir müssen also nur die Ringautomorphismen $\alpha \in \mathrm{Inn}_\xi^{(i)} \setminus \mathrm{Inn}_\xi^{(i+1)}$ betrachten. Ein solcher werde von dem Ringelement $1 + t\theta^{i-1} + b'\theta^{i+1}$ mit $t \in T^*$ und $b' \in R$ induziert. Es folgt:

$$\theta + \mathrm{coeff}^{(i)}(\alpha(\theta))\theta^i \equiv (1 + t\theta^{i-1})\theta(1 - t\theta^{i-1}) \equiv \theta + (t - \tau^e(t))\theta^i \mod \mathrm{Rad}(R)^{i+1}$$

Damit haben wir die Inklusionsrichtung „$\subseteq$“ bewiesen. Umgekehrt sieht man leicht, dass für $t \in T^*$ die Einheit $1 + t\theta^{i-1} \in Z(\xi)$ einen inneren Automorphismus $\alpha \in \mathrm{Inn}_\xi^{(i)}$ induziert mit $\overline{\mathrm{coeff}}^{(i)}(\alpha(\theta)) = \bar{t} - \tau^e(\bar{t})$. □

Es sei wieder $\mathbb{F}_{p^s} \leq \mathbb{F}_q$ der Fixkörper von τ^e, also $s = \mathrm{ggT}(r,e)$. Die Abbildung $\mathbb{F}_q \to \mathbb{F}_q, x \mapsto x - \tau^e(x)$ ist dann $\mathbb{F}_{p^s}$-linear. Die Menge $T_1 := \{a - \tau^e(a) \mid a \in \mathbb{F}_q\}$ aus dem vorangegangenem Hilfssatz ist somit ein $\mathbb{F}_p$-Unterraum der Dimension $r - s$. Die vorausgegangenen Ergebnisse über die Gruppen $Z(\xi)$, $Z(R)$ und Inn_ξ fassen wir nun in dem folgenden Satz zusammen:

4.2.15 Satz. *Es sei $x := \frac{r}{\mathrm{ggT}(r,e)}$ und $l := \left\lfloor \frac{m-1}{x} \right\rfloor$. Dann sind*

$$\begin{array}{ccccccccccccc}
Z(\xi) & \overset{\mathbb{F}_q^*}{\trianglerighteq} & Z^{(x)}(\xi) & \overset{\mathbb{F}_p^r}{\trianglerighteq} & Z^{(2x)}(\xi) & \overset{\mathbb{F}_p^r}{\trianglerighteq} & \dots & \overset{\mathbb{F}_p^r}{\trianglerighteq} & Z^{(lx)}(\xi) & \overset{\mathbb{F}_p^r}{\trianglerighteq} & Z^{((l+1)x)}(\xi) & \trianglerighteq & \{1_R\} \\
Z(R) & \overset{\mathbb{F}_{p^s}^*}{\trianglerighteq} & Z^{(x)}(R) & \overset{\mathbb{F}_p^s}{\trianglerighteq} & Z^{(2x)}(R) & \overset{\mathbb{F}_p^s}{\trianglerighteq} & \dots & \overset{\mathbb{F}_p^s}{\trianglerighteq} & Z^{(lx)}(R) & \overset{\mathbb{F}_p^s}{\trianglerighteq} & Z^{((l+1)x)}(R) & \trianglerighteq & \{1_R\} \\
\mathrm{Inn}_\xi & \overset{\mathbb{F}_q^*/\mathbb{F}_{p^s}^*}{\trianglerighteq} & \mathrm{Inn}_\xi^{(x+1)} & \overset{\mathbb{F}_p^{r-s}}{\trianglerighteq} & \mathrm{Inn}_\xi^{(2x+1)} & \overset{\mathbb{F}_p^{r-s}}{\trianglerighteq} & \dots & \overset{\mathbb{F}_p^{r-s}}{\trianglerighteq} & \mathrm{Inn}_\xi^{(lx+1)} & \overset{\mathbb{F}_p^{r-s}}{\trianglerighteq} & \mathrm{Inn}_\xi^{((l+1)x+1)} & = & \{\mathrm{id}_R\}
\end{array}$$

die Normalreihen von $Z(\xi)$, $Z(R)$ und Inn_ξ. Ist $(l+1)x = m - 1 \iff x \mid (m-1)$, so ist $Z^{((l+1)x)}(R) = Z^{((l+1)x)}(\xi) \simeq \mathbb{F}_p^r$. Andernfalls sind beide Gruppen trivial.

4.2.16 Bemerkung. Das hier erzielte Ergebnis zu $Z(R)$ deckt sich mit dem Resultat von A. Nechaev über das Zentrum [59, Theorem 5.18 (d)] des Rings R.

Wir schließen das Kapitel mit einer Beschreibung der Gruppe $\mathrm{Out}(R)$ über eine Normalreihe. Hierzu seien für $i \in [r+1]$ und $j \in [m], j \geq 2$ die Gruppen

$$\mathrm{Inn}_\xi^{(i,j)} := \mathrm{Aut}_\xi^{(i,j)} \cap \mathrm{Inn}(R) \qquad \text{und} \qquad \mathrm{Out}_\xi^{(i,j)} := \mathrm{Aut}_\xi^{(i,j)} \,/\, \mathrm{Inn}_\xi^{(i,j)}\,.$$

Aus dem ersten Isomorphiesatz für Gruppen ergibt sich wieder die Isomorphie

$$\mathrm{Out}_\xi^{(i,j)} \simeq \left(\mathrm{Aut}_\xi^{(i,j)} \circ \mathrm{Inn}(R)\right) / \mathrm{Inn}(R)$$

und damit auch, dass wir $\mathrm{Out}_\xi^{(i+1,j)}$ als Normalteiler in $\mathrm{Out}_\xi^{(i,j)}$ auffassen können. Weiter ist genau dann $\mathrm{Out}_\xi^{(i+1,j)} = \mathrm{Out}_\xi^{(i,j)}$, wenn $\mathrm{Aut}_\xi^{(i+1,j)} = \mathrm{Aut}_\xi^{(i,j)}$ oder $\mathrm{Inn}_\xi^{(i+1,j)} \neq \mathrm{Inn}_\xi^{(i,j)}$ gilt. Andernfalls ist die Faktorgruppe $\mathrm{Out}_\xi^{(i,j)} / \mathrm{Out}_\xi^{(i+1,j)}$ isomorph zu $\mathbb{Z}_p$. Zusammenfassend ergibt sich der folgende Satz:

4.2.17 Satz. *Es sei* $x := \frac{r}{\mathrm{ggT}(r,e)}$ *und*

$$I := \Big\{ i \in [r] \mid \exists\ t \in T \text{ sd. } \forall\ i' < i : \mathrm{coeff}^{(i',0)}(t - \tau^e(t)) = 0 \wedge \mathrm{coeff}^{(i,0)}(t - \tau^e(t)) \neq 0 \Big\}.$$

Dann ist

$$\mathrm{Out}(R) \trianglerighteq \mathrm{Out}_\xi \trianglerighteq \mathrm{Out}_\xi^{(0,2)} \trianglerighteq \ldots \trianglerighteq \mathrm{Out}_\xi^{(r,2)} = \mathrm{Out}_\xi^{(0,3)} \trianglerighteq \ldots \trianglerighteq \mathrm{Out}_\xi^{(r,m-1)} \tag{4.2}$$

eine Normalreihe mit zyklischen Faktoren. Dabei ist

- $\mathrm{Out}_T / \mathrm{Out}_\xi \simeq \mathbb{Z}_{r'}$ *mit* $r' \mid r$ *aus Hilfssatz 4.2.6 und*
- $\mathrm{Out}_\xi / \mathrm{Out}_\xi^{(0,2)} \simeq \mathbb{Z}_{s'}$ *mit* $s' \mid (p^{\mathrm{ggT}(r,e)} - 1)$.

Für $i \in [r]$ *und* $j \in [m] : j \geq 2$ *gilt weiter*

- $\mathrm{Out}_\xi^{(i,j)} / \mathrm{Out}_\xi^{(i+1,j)} \simeq \mathbb{Z}_p$, *falls* $\mathrm{Aut}_\xi^{(i,j)} \neq \mathrm{Aut}_\xi^{(i+1,j)}$ *und* $x \nmid (j-1) \vee i \notin I$ *und*
- $\mathrm{Out}_\xi^{(i,j)} = \mathrm{Out}_\xi^{(i+1,j)}$ *in allen anderen Fällen.*

Beweis. Mit Ausnahme von $\mathrm{Out}_\xi / \mathrm{Out}_\xi^{(0,2)} \simeq \mathbb{Z}_{s'}$ mit $s' \mid (p^{\mathrm{ggT}(r,e)} - 1)$ wurden alle Aussagen bereits weiter oben behandelt. Die verbliebene Aussage lässt sich aber mit dem Homomorphismus $\Psi : \mathrm{Aut}_\xi \to \mathbb{F}_q^*$, $\alpha \mapsto \overline{\mathrm{coeff}}^{(1)}(\alpha(\theta))$ und der Beobachtung $\Psi(\mathrm{Inn}_\xi) = \left\langle \xi^{1-p^e} \right\rangle = \left\langle \xi^{p^{\mathrm{ggT}(r,e)}-1} \right\rangle$ sofort folgern. □

Die notwendigen Modifikationen an Algorithmus A.1 zur Berechnung eines Erzeugendensystems von $\mathrm{Out}(R)$ geben wir in Bemerkung A.5 im Anhang A.

5. Lineare Codes über endlichen Kettenringen

Wir werden nun im Folgenden lineare Codes der Länge n über einem Kettenring R mit fest vorgegebenen Umriss $\lambda = (\lambda_0, \ldots, \lambda_{k-1})$ untersuchen. Ziel dieses Kapitels ist es, einen Kanonisierer für die Operation der Gruppe der semilinearen Isometrien von R^n auf dieser Menge zu entwickeln.

Da wir einen linearen Code C in diesem Algorithmus über eine Generatormatrix darstellen möchten, werden wir zunächst die Menge aller Generatormatrizen von C über eine Gruppenoperation beschreiben. Wie wir bereits in Beispiel 2.3.18 gesehen haben, führt – bei nicht freien Codes – die Multiplikation einer Generatormatrix Γ von C mit einer invertierbaren Matrix $A \in \mathrm{GL}_k(R)$ nicht notwendigerweise zu einer weiteren Generatormatrix von C. Wir wollen daher zunächst die Gruppe $\mathrm{GL}_k(R)$ derart auf eine Untergruppe $\mathrm{GL}_\lambda(R)$ einschränken, dass wir die Menge aller Generatormatrizen von C wieder über die Bahnenmenge $\mathrm{GL}_\lambda(R)\Gamma$ erhalten. Einen eindeutigen Repräsentanten dieser Bahn erhalten wir über eine Verallgemeinerung der reduzierten Zeilenstufenform.

In einem weiteren Schritt transformieren wir dann die Problemstellung auf eine Operation der Gruppe $((\mathrm{GL}_\lambda(R) \times (R^*)^n) \rtimes \mathrm{Aut}_T) \rtimes S_n$ auf der Menge $R^{k\times n,\lambda}$ aller Generatormatrizen zum Umriss λ. Diese Operation können wir mit den Algorithmen aus Kapitel 3 behandeln. Eine entscheidende Rolle wird hierbei der Operation der Untergruppe $((\mathrm{GL}_\lambda(R) \times (R^*)^n) \rtimes \mathrm{Aut}_T)$ zukommen. Für diese werden wir in dem Abschnitt 5.1.2 einen effizienten Kanonisierer bereitstellen.

Der zweite Abschnitt 5.2 beschreibt dann schlussendlich den gewünschten Kanonisierer für die Operation von $((\mathrm{GL}_\lambda(R) \times (R^*)^n) \rtimes \mathrm{Aut}_T) \rtimes S_n$ auf der Menge $R^{k\times n,\lambda}$. Diesen werden wir durch Bereitstellung einer inneren Kanonisierung und einer äußeren Verfeinerung über den in Abschnitt 3.3 beschriebenen Basisalgorithmus gewinnen.

5.1. Generatormatrizen

Wir wollen nun, wie bereits angedeutet, zunächst die Untergruppe $\mathrm{GL}_\lambda(R)$ definieren und die Operation dieser Gruppe auf der Menge $R^{k\times n,\lambda}$ untersuchen.

5.1.1 Definition. Zu einem Umriss $\lambda = (\lambda_0, \ldots, \lambda_{k-1})$ und $i \in [m]$ sei

$$k_i^\lambda := |\{j \in [k] \mid \lambda_j = m - i\}|$$

und $\mathrm{GL}_\lambda(R)$ die Menge aller Blockmatrizen der Gestalt $\left(A^{(i,j)}\right)_{i,j\in[m]} \in R^{k\times k}$ mit

$$A^{(i,j)} \in \begin{cases} R^{k_i^\lambda \times k_j^\lambda}, & \text{falls } 0 \le i < j \le m-1 \\ \mathrm{GL}_{k_i^\lambda}(R), & \text{falls } 0 \le i = j \le m-1 \\ \theta^{i-j} R^{k_i^\lambda \times k_j^\lambda}, & \text{falls } 0 \le j < i \le m-1. \end{cases}$$

5.1.2 Fakt ([73, Lemma 2.21])**.** *Die Menge* $\mathrm{GL}_\lambda(R)$ *bildet eine Untergruppe von* $\mathrm{GL}_k(R)$*.*

5.1.3 Bemerkung. Nach [73, Lemma 2.20] ist eine Blockmatrix $\left(A^{(i,j)}\right)_{i,j\in[m]}$ mit

$$A^{(i,j)} \in \begin{cases} R^{k_i^\lambda \times k_j^\lambda}, & \text{falls } 0 \le i \le j \le m-1 \\ \mathrm{Rad}(R)^{k_i^\lambda \times k_j^\lambda}, & \text{falls } 0 \le j < i \le m-1 \end{cases}$$

genau dann invertierbar, wenn alle $A^{(i,i)}$ invertierbar sind.

5.1.4 Fakt ([73, Satz 2.22])**.** *Ist* $\Gamma \in R^{k\times n,\lambda}$ *eine Generatormatrix zu einem Code* C *vom Umriss* λ*, so ist die Menge aller Generatormatrizen von* C *gleich der Bahn* $\mathrm{GL}_\lambda(R)\Gamma$*.*

Für nicht freie Codes führt diese Gruppe aber zu einer nicht treuen Gruppenoperation:

5.1.5 Fakt ([73, Lemma 2.23])**.** *Für eine beliebige Generatormatrix* $\Gamma \in R^{k\times n,\lambda}$ *ist*

$$\mathrm{Stab}_{\mathrm{GL}_\lambda(R)}(\Gamma) = N_\lambda(R) := I_k + \left\{ \left(v_0\theta^{\lambda_0}, \ldots, v_{k-1}\theta^{\lambda_{k-1}}\right) \mid v_i \in (R^k)_R \right\}.$$

Insbesondere ist für $\lambda \neq (m, \ldots, m)$ *– d.h. für nicht freie Codes – die Gruppenoperation von* $\mathrm{GL}_\lambda(R)$ *auf* $R^{k\times n,\lambda}$ *nicht treu. Die Untergruppe* $N_\lambda(R)$ *ist als Schnitt aller Stabilisatoren ein Normalteiler von* $\mathrm{GL}_\lambda(R)$*.*

5.1.6 Bemerkung. Im Folgenden werden wir die Operation der Gruppe $\mathrm{GL}_\lambda(R)$ auf $R^{k\times n,\lambda}$ durch die natürliche Operation der Faktorgruppe $\mathrm{GL}_\lambda(R)/N_\lambda(R)$ ersetzen.

Aus Gründen der Übersichtlichkeit werden wir aber keine Unterscheidung zwischen den Gruppenelementen $A \in \mathrm{GL}_\lambda(R)$ und $AN_\lambda(R) \in \mathrm{GL}_\lambda(R)/N_\lambda(R)$ machen. Aussagen, wie zum Beispiel „$AN_\lambda(R)$ ist von oberer Dreiecksgestalt“, beziehen sich dann auf die Existenz eines Nebenklassenvertreters $A \in \mathrm{GL}_\lambda(R)$ mit dieser Eigenschaft.

5.1.7 Folgerung. *Zwei Generatormatrizen* $\Gamma, \Gamma' \in R^{k\times n,\lambda}$ *erzeugen genau dann semilinear isometrische Codes, wenn es ein Gruppenelement*

$$(AN_\lambda(R), \varphi; \alpha, \pi) \in (\mathrm{GL}_\lambda(R)/N_\lambda(R) \times (R^*)^n) \rtimes (\mathrm{Aut}(R) \times S_n)$$

gibt mit $(AN_\lambda(R), \varphi; \alpha, \pi)\Gamma = (A, \varphi; \alpha, \pi)\Gamma = \Gamma'$*.*

Über einem endlichen Körper $\mathbb{F}_q$ können wir jedem linearen Code C eine eindeutige Generatormatrix zuordnen, indem wir diejenige Generatormatrix $\Gamma \in \mathbb{F}_q^{k\times n}$ auszeichnen, welche in reduzierter Zeilenstufenform vorliegt. Sie ist bekanntermaßen eindeutig in der Bahn $\mathrm{GL}_k(\mathbb{F}_q)\Gamma$ und bildet somit einen kanonischen Repräsentanten für diese Operation. Wir wollen nun zeigen, dass dies im Fall eines linearen Codes über einem endlichen Kettenring genauso möglich ist. Wir verallgemeinern zunächst die Definition der reduzierten Zeilenstufenform einer Matrix:

5.1.8 Definition (reduzierte Zeilenstufenform)**.** Wir sagen eine Matrix $\Gamma \in R^{k\times n}$ ist in *Zeilenstufenform*, falls es eine Folge von Spaltenindizes $0 \leq j_0 < \ldots < j_{k'-1} \leq n-1$ mit $k' \leq k$ gibt, so dass

- für alle $i \in [k']$ die Einträge Γ_{i,j_i} *Pivotelemente* sind, d.h. $\Gamma_{i,j_i} \neq 0$ und für alle $i \in [k']$ und $j \in [j_i]$ ist $\Gamma_{i',j} = 0$, und
- alle weiteren Zeilen $\Gamma_{i,*}$ für $k' \leq i < k$ Nullzeilen sind.

Die Spalten Γ_{j_i} nennen wir auch *Pivotspalten*. Die Zeilenstufenform ist *reduziert*, falls sie überdies die folgenden Eigenschaften erfüllt:

- für alle $i \in [k']$ ist das Pivotelement $\Gamma_{i,j_i} = \theta^{m-\lambda_i}$ mit $\lambda_i := \mathrm{per}(\Gamma_{i,*})$,
- die Folge $(\lambda_0, \ldots, \lambda_{k'-1})$ ist monoton fallend und
- alle weiteren Einträge der Spalte Γ_{*,j_i} sind modulo dem Pivotelement reduziert, d.h. $\Gamma_{\ell,j_i} = \min_{a\in R}\big(\Gamma_{\ell,j_i} + a\theta^{m-\lambda_i}\big),\ \forall\ \ell \neq i$.

5.1.9 Bemerkung. Aufgrund der von uns gewählten Ordnung auf R beschreibt die letzte Bedingung, dass die letzten λ_i Summanden der θ-adischen Entwicklung von Γ_{ℓ,j_i} gleich Null sind, d.h. $\Gamma_{\ell,j_i} = \sum_{h=0}^{m-\lambda_i-1} \mathrm{coeff}^{(h)}(\Gamma_{\ell,j_i}) \cdot \theta^h$. Insbesondere falls $\lambda_i = m$ gilt, ist somit die Spalte Γ_{*,j_i} gleich dem i-ten Einheitsvektor.

5.1.10 Folgerung. *Die ersten k' Zeilen einer Matrix $\Gamma \in R^{k\times n}$ in reduzierter Zeilenstufenform bilden eine Generatormatrix des von Γ erzeugten linearen Codes.*

Beweis. Man zeigt leicht, dass die Zeilen unabhängig sind. Aufgrund der geforderten Anordnung der Perioden ist $\Gamma_{[k'],*}$ also eine Generatormatrix. □

5.1.11 Beispiel. Die Matrix $\Gamma = \begin{pmatrix}2 & 1\end{pmatrix} \in \mathbb{Z}_4^{1\times 2,(2)}$ ist eine Generatormatrix eines linearen Codes in Zeilenstufenform. Sie ist jedoch nicht reduziert. Da alle weiteren Generatormatrizen von C durch Linksmultiplikation mit Einheiten aus Γ hervorgehen, gibt es also für C keine Generatormatrix in reduzierter Zeilenstufenform.

Das Beispiel verdeutlicht, dass nicht jede Bahn $\mathrm{GL}_k(R)\Gamma$ einer Matrix $\Gamma \in R^{k\times n}$ ein Element in reduzierter Zeilenstufenform enthält. Umgekehrt zeigt aber der nachfolgende Hilfssatz, dass sie im Fall der Existenz eindeutig bestimmt ist und damit durchaus zur Definition eines kanonischen Repräsentanten geeignet ist.

5.1.12 Hilfssatz. *Ist $\Gamma \in R^{k\times n}$ eine Matrix in reduzierter Zeilenstufenform, so existiert in der Bahn $\mathrm{GL}_k(R)\Gamma$ keine weitere Matrix in reduzierter Zeilenstufenform.*

Beweis. Wir nehmen an, dass $\widetilde{\Gamma} \in \mathrm{GL}_k(R)\Gamma$ eine weitere Matrix in reduzierter Zeilenstufenform sei. Ist nun k' der Rang des von Γ erzeugten linearen Codes C und λ dessen Umriss, so sind also $\Gamma_{[k'],*}$ und $\widetilde{\Gamma}_{[k'],*}$ beides Generatormatrizen von C und beide Teilmatrizen in reduzierter Zeilenstufenform.

Die weiteren Zeilen von Γ und $\widetilde{\Gamma}$ sind Nullzeilen und somit können wir ohne Beschränkung der Allgemeinheit annehmen, dass $k' = k$ gilt. Die Matrix $\widetilde{\Gamma}$ erhalten wir also durch Linksmultiplikation von Γ mit einer geeigneten Matrix $A \in \mathrm{GL}_\lambda(R)$. Wir zeigen nun über eine Induktion nach k, dass A bereits in $N_\lambda(R)$ liegt und somit beide Matrizen gleich sind.

Für $k = 1$ ist die Behauptung leicht einzusehen. Für den Induktionsschritt seien $j_0, \ldots, j_{k-1}$ sowie $\ell_0, \ldots, \ell_{k-1}$ die Indizes der Pivotspalten von Γ bzw. $\widetilde{\Gamma}$. Man überlegt sich leicht, dass $j_0 = \ell_0$ gelten muss. Aus der Gleichung $A\Gamma_{*,j_0} = \widetilde{\Gamma}_{*,j_0}$ erhält man damit $A_{0,0} \in 1 + \mathrm{Rad}(R)^{\lambda_0}$ und $A_{i,0} \in \mathrm{Rad}(R)^{\lambda_0}$ für alle $i > 0$. Die Teilmatrizen $\Gamma_{\geq 1,*}$ und $\widetilde{\Gamma}_{\geq 1,*}$ erzeugen aus diesem Grund den gleichen linearen Code und sind Generatormatrizen in reduzierter Zeilenstufenform. Für sie gilt also die Induktionsvoraussetzung und es ist $\Gamma_{\geq 1,*} = \widetilde{\Gamma}_{\geq 1,*}$. Damit liegen auch die weiteren Pivotspalten an identischen Koordinatenpositionen, d.h. $j_\mu = \ell_\mu$ für alle $\mu \in [k]$.

Abschließend zeigt man über eine Induktion für $\mu > 0$, dass $A_{0,\mu} \in \mathrm{Rad}(R)^{\lambda_\mu}$ liegt. Wir betrachten hierzu die Gleichung

$$\widetilde{\Gamma}_{0,j_\mu} = (A\Gamma)_{0,j_\mu} = A_{0,0}\Gamma_{0,j_\mu} + \sum_{i=1}^{\mu-1} \underbrace{A_{0,i}\Gamma_{i,j_\mu}}_{=0} + A_{0,\mu}\theta^{m-\lambda_\mu} = \Gamma_{0,j_\mu} + A_{0,\mu}\theta^{m-\lambda_\mu}.$$

Die Matrixeinträge $\widetilde{\Gamma}_{0,j_\mu} =$ und Γ_{0,j_μ} sind modulo $\theta^{m-\lambda_\mu}$ reduziert, d.h. Nebenklassenrepräsentanten von $R/\mathrm{Rad}(R)^{m-\lambda_\mu}$. Damit folgt die Behauptung $A_{0,\mu} \in \mathrm{Rad}(R)^{\lambda_\mu}$. □

5.1.13 Definition (systematische Generatormatrix)**.** Eine Generatormatrix $\Gamma \in R^{k\times n,\lambda}$ in reduzierter Zeilenstufenform heißt *systematisch*, falls sich die Pivotspalten an den Positionen $j_0 = 0$ bis $j_{k-1} = k - 1$ befinden.

Für lineare Codes über endlichen Körpern kennt man bereits die Aussage, dass ein beliebiger linearer Code permutationsäquivalent zu einem Code mit systematischer Generatormatrix ist. Für Kettenringe gilt diese Aussage ebenfalls:

5.1.14 Fakt ([73, Satz 2.1])**.** *Es sei $\Gamma \in R^{k\times n}$ eine beliebige Matrix, welche einen linearen Code C vom Umriss $\mathrm{shp}(C) = \lambda = (\lambda_0, \ldots, \lambda_{k'-1})$ erzeugt. Dann gibt es eine Permutation $\pi \in S_n$, so dass der lineare Code $\pi \cdot C$ eine systematische Generatormatrix $\widetilde{\Gamma} \in R^{k\times n,\lambda}$ besitzt.*

Insbesondere ist der Beweis zu dem Satz konstruktiv. Es wird also in [73] explizit ein Algorithmus zur Berechnung der Permutation $\pi \in S_n$ und einer Matrix $A \in \mathrm{GL}_k(R)$ mit $A\Gamma P^{(\pi)^{-1}} = \begin{pmatrix} \widetilde{\Gamma} \\ \mathbf{0}_{(\mathbf{k}-\mathbf{k}')\times\mathbf{n}} \end{pmatrix}$ angegeben.

Will man nun für beliebige Matrizen $\Gamma \in R^{k\times n}$ einen kanonischen Repräsentanten $\mathrm{CF}_{\mathrm{GL}_k(R)}(\Gamma)$ unter der Operation von $\mathrm{GL}_k(R)$ bestimmen, so ist es einzig nötig, die Wahl der Permutation $\pi \in S_n$ aus Fakt 5.1.14 $\mathrm{GL}_k(R)$-invariant vorzunehmen. Dies ist in der Tat auch möglich: Man kann sich leicht überlegen, dass es genügt, die Auswahl eines Elements $a_{i,j}$ im Beweis zu [73, Satz 2.1] derart vorzunehmen, dass j stets minimal ist. Dann bestimmt man den kanonischen Repräsentanten $\mathrm{CF}_{\mathrm{GL}_k(R)}(\Gamma)$ von Γ über die folgenden Schritte:

1. Bestimme eine Matrix $A \in \mathrm{GL}_k(R)$, so dass $A({\Gamma P^{(\pi)}}^{-1})$ bis auf weitere Nullzeilen die eindeutige systematische Generatormatrix in der Bahn $\mathrm{GL}_k(R)({\Gamma P^{(\pi)}}^{-1})$ ist.

2. Definiere $\mathrm{CF}_{\mathrm{GL}_k(R)}(\Gamma) := A\Gamma = \left(A\Gamma {P^{(\pi)}}^{-1}\right)P^{(\pi)}$. Diese Auswahl ist somit ebenfalls eindeutig in der Bahn $\mathrm{GL}_k(R)\Gamma$.

5.1.15 Bemerkung. Man kann sich leicht überlegen, dass dieses Vorgehen mit der Definition der sog. *Fuller Canonical Form*[1] für Galois-Ringe, gemäß [54, Exercise (XVI.7)] bzw. deren weiterer Verallgemeinerung auf beliebige Kettenringe in [17], übereinstimmt.

Ist $\Gamma \in R^{k\times n}$ eine Generatormatrix in reduzierter Zeilenstufenform, so ist sie mit dieser Definition der kanonische Repräsentant ihrer eigenen Bahn. Wir wollen nun noch diejenigen Matrizen auszeichnen, welche ohne Permutation der Spalten auf reduzierte Zeilenstufenform gebracht werden können:

5.1.16 Definition (umrisstreu)**.** Wir sagen eine Matrix $\Gamma \in R^{k\times n}$ mit $\mathrm{shp}(\Gamma) = \lambda$ ist *umrisstreu*, falls für alle $i \in [n]$ die Teilmatrizen $\Gamma_{*,[i]}$ den Umriss $\mathrm{shp}(\Gamma_{*,[i]}) = \lambda_{\left[\mathrm{rg}(\Gamma_{*,[i]})\right]}$ haben. Dabei bezeichnet $\mathrm{rg}(\Gamma_{*,[i]})$, wie vereinbart, den Rang des von $\Gamma_{*,[i]}$ erzeugten Zeilenraums.

5.1.17 Hilfssatz. *Eine Matrix $\Gamma \in R^{k\times n}$ ist genau dann umrisstreu, wenn sie durch Linksmultiplikation mit einer Matrix $A \in \mathrm{GL}_k(R)$ auf reduzierte Zeilenstufenform $A\Gamma$ transformiert werden kann.*

Beweis. Jede Matrix $\Gamma' \in R^{k\times n}$ in reduzierter Zeilenstufenform ist ganz offensichtlich umrisstreu. Die Eigenschaft umrisstreu zu sein, ist aber eine $\mathrm{GL}_k(R)$-Invariante. Somit ist auch jedes weitere Element der Bahn $\mathrm{GL}_k(R)\Gamma'$ umrisstreu.

Es bleibt also nur noch zu zeigen, dass sich jede umrisstreue Matrix $\Gamma \in R^{k\times n}$ auch auf reduzierte Zeilenstufenform transformieren lässt. Wir zeigen dies über eine Induktion nach der Anzahl der Spalten. Für den Induktionsstart $n = 0$ ist die Aussage trivial.

[1] Sie ist motiviert durch die Arbeit [27].

Für den Induktionsschritt von $n-1$ nach $n \geq 1$ sei $k' = \mathrm{rg}(\Gamma_{*,[n-1]})$ und $\lambda = \mathrm{shp}(\Gamma)$. Die Matrix $\Gamma_{*,[n-1]}$ ist ebenfalls umrisstreu und wir können daher ohne Beschränkung der Allgemeinheit annehmen, dass diese in reduzierter Zeilenstufenform vorliegt:

$$\Gamma = \begin{pmatrix} \Gamma_{[k'],[n-1]} & \Gamma_{[k'],n-1} \\ \mathbf{0}_{(\mathbf{k-k'})\times(\mathbf{n-1})} & \Gamma_{[k]\setminus[k'],n-1} \end{pmatrix}$$

Ist $\Gamma_{[k]\setminus[k'],n-1} = \mathbf{0}_{(\mathbf{k-k'})\times\mathbf{1}}$, d.h. $k' = \mathrm{rg}(\Gamma)$, so ist Γ in reduzierter Zeilenstufenform. Hierzu macht man sich leicht klar, dass für $i \in [k']$ die Annahme $\Gamma_{i,n-1} \notin \mathrm{Rad}(R)^{m-\lambda_i}$ sofort der Eigenschaft umrisstreu zu sein widerspräche.

Im Fall $\Gamma_{[k]\setminus[k'],n-1} \neq \mathbf{0}_{(\mathbf{k-k'})\times\mathbf{1}}$ schließt man analog, dass für alle $i \in [k']$ bereits $\Gamma_{i,n-1} \in \mathrm{Rad}(R)^{m-\lambda_i}$ gelten muss. Des Weiteren ist dann auch $\mathrm{per}(\Gamma_{[k]\setminus[k'],n-1}) = \lambda_{k'}$ und es existiert ein Eintrag $\Gamma_{j,n-1}$ mit $k' \leq j < k$ und $\mathrm{per}(\Gamma_{j,n-1}) = \lambda_{k'}$. Der Stabilisator $\mathrm{Stab}_{\mathrm{GL}_k(R)}\big(\Gamma_{*,[n-1]}\big)$ beinhaltet aber nun alle Matrizen der Gestalt

$$\begin{pmatrix} I_{k'} & B \\ \mathbf{0}_{(\mathbf{k-k'})\times\mathbf{k'}} & A \end{pmatrix} \text{ mit } A \in \mathrm{GL}_{k-k'}(R) \text{ und } B \in R^{k'\times(k-k')},$$

welche es uns erlauben, Γ auf reduzierte Zeilenstufenform zu transformieren. □

5.1.18 Hilfssatz. *Ist die Matrix $\Gamma \in R^{k\times n}$ umrisstreu und $(\varphi;\alpha) \in (R^*)^n \rtimes \mathrm{Aut}(R)$ beliebig, so ist auch $(\varphi;\alpha)\Gamma$ umrisstreu.*

Beweis. Die Gruppenoperation mit $(\varphi;\alpha)$ definiert einen Verbandsautomorphismus auf $\mathrm{PHG}({}_RR^n)$. Insbesondere bleibt also der Umriss einer Matrix unberührt. Dies gilt natürlich auch für alle Teilmatrizen $((\varphi;\alpha)\Gamma)_{*,[i]}$, $i \in [n+1]$. Somit ist $(\varphi;\alpha)\Gamma$ ebenfalls umrisstreu. □

5.1.1. Reformulierung der Gruppenoperation

In Folgerung 5.1.7 haben wir das Kanonisierungsproblem für lineare Codes bereits auf eine Gruppenoperation der Gruppe

$$(\mathrm{GL}_\lambda(R)/N_\lambda(R) \times (R^*)^n) \rtimes (\mathrm{Aut}(R) \times S_n)$$

auf der Menge $R^{k\times n,\lambda}$ zurückgeführt. Insbesondere haben wir auch festgestellt, dass die Untergruppe $\mathrm{GL}_\lambda(R)$ nicht treu operiert und dies durch den Übergang zu der Faktorgruppe $\mathrm{GL}_\lambda(R)/N_\lambda(R)$ behoben. Eine analoge Aussage wollen wir nun auch für die Spalten der Generatormatrix herleiten. Hierdurch werden wir die Komponenten $(R^*)^n$ und S_n der operierenden Gruppe noch weiter einschränken.

5.1.19 Hilfssatz. *Es sei $G := (\mathrm{GL}_\lambda(R)/N_\lambda(R) \times (R^*)^n) \rtimes \mathrm{Aut}(R)$. Die Abbildung*

$$f^{(\mathrm{per})} : R^{k\times n,\lambda} \to [m+1]^n, \quad \Gamma \mapsto (\mathrm{per}(\Gamma_{*,i}))_{i\in[n]},$$

welche jeder Spalte von $\Gamma \in R^{k\times n,\lambda}$ ihre Periode[2] zuordnet, ist eine G-Invariante und ein S_n-Homomorphismus.

[2] Zur Erinnerung: Die Periode $\mathrm{per}(v)$ eines Spaltenvektors $v \in R_R^k$ ist definiert als die minimale ganze Zahl mit $v\theta^{\mathrm{per}(v)} = \mathbf{0}_{\mathbf{1}\times\mathbf{k}}$.

Beweis. Die Periode eines Spaltenvektors $v \in (R^k)_R$ bleibt invariant unter der Multiplikation mit invertierbaren Matrizen von links, der Multiplikation von Einheiten von rechts und auch unter der komponentenweisen Anwendung eines Ringautomorphismus. Somit ist die Funktion invariant unter der Operation mittels G im Definitionsbereich. Sie ist außerdem ganz offensichtlich auch ein S_n-Homomorphismus. □

Zwei Generatormatrizen $\Gamma, \Gamma' \in R^{k\times n,\lambda}$ erzeugen nur dann semilinear isometrische Codes, wenn die Bilder $f^{(\mathrm{per})}(\Gamma)$ und $f^{(\mathrm{per})}(\Gamma')$ in der gleichen Bahn unter der Operation von S_n liegen.

Wir legen nun über das Homomorphieprinzip und über die Kanonisierung im Bildbereich von $f^{(\mathrm{per})}$ fest, dass die Folge $\mathrm{CF}_{S_n}\big(f^{(\mathrm{per})}(\Gamma)\big) =: \mu \in [m+1]^n$ monoton fällt. Somit sind die Perioden der Spalten jeder kanonischen Generatormatrix $\mathrm{CF}_{G\rtimes S_n}(\Gamma)$, $\Gamma \in R^{k\times n,\lambda}$ beliebig, ebenfalls monoton fallend. Wir können also im Folgenden, ohne Beschränkung der Allgemeinheit, bereits im Vorfeld voraussetzen, dass die Eingaben $\Gamma \in R^{k\times n,\lambda}$ des Kanonisierers $\mathrm{Can}_{G\rtimes S_n}$ bereits diese Bedingungen erfüllen.

In der Operation der symmetrischen Gruppe ziehen wir uns dann auf den Stabilisator $S_{\mathfrak{P}_0} := \mathrm{Stab}_{S_n}(\mu)$ zurück. Die Blöcke der kanonischen Partition $\mathfrak{P}_0$ beschreiben hierbei die Koordinaten der Spalten mit gleicher Periode.

Eine weitere Beobachtung, die wir an dieser Stelle einbringen können, bezieht sich auf eventuell auftretende Nullspalten der Generatormatrizen (dann ist $\mu_i = 0$). Vom codierungstheoretischen Standpunkt sind diese Koordinaten ohnehin redundant. Aus Sicht der Gruppenoperation gilt diese Beobachtung ebenfalls, da wir unter einem beliebigen Gruppenelement $(A, \varphi; \alpha, \pi) \in ((\mathrm{GL}_\lambda(R)/N_\lambda(R) \times (R^*)^n) \rtimes \mathrm{Aut}(R)) \rtimes S_{\mathfrak{P}_0}$ Nullspalten immer auf Nullspalten abbilden. Wir entfernen diese gegebenenfalls vor der Kanonisierung und fügen sie nach Abschluss in gleicher Zahl wieder an die kanonische Form an. Es ist klar, wie in diesem Fall der berechnete Stabilisator zu modifizieren ist.

5.1.20 Definition. Zu einem beliebigen Vektor $\mu \in (\{1,\dots,m\})^n$ sei

$$R^{k\times n,\lambda,\mu} := \big\{\Gamma \in R^{k\times n,\lambda} \mid \forall\, i \in [n] : \mathrm{per}(\Gamma_{*,i}) = \mu_i\big\}$$

die Menge aller Generatormatrizen vom Umriss λ mit fest vorgegebener Periode μ_i der i-ten Spalte.

Im Folgenden sei also auch die monoton fallende Folge $\mu \in (\{1,\dots,m\})^n$ fest vorgeben und es werden nur noch Generatormatrizen aus der Teilmenge $R^{k\times n,\lambda,\mu}$ betrachtet. Zu dem Vektor μ sei im weiteren Verlauf $\mathfrak{P}_0$ die kanonische Partition von $[n]$, welche über den Stabilisator $S_{\mathfrak{P}_0} := \mathrm{Stab}_{S_n}(\mu)$ eindeutig bestimmt ist.

5.1.21 Bemerkung. Einen Vektor $v \in (R^k)_R$ mit $\mathrm{per}(v) = m$ wollen wir als *fett* bezeichnen. Er erzeugt den Punkt vR in der projektiven Rechts-Hjelmslev-Geometrie $\mathrm{PHG}((R^k)_R)$. Besitzen die auftretenden Generatormatrizen nur fette Spaltenvektoren (d.h. $\mu = (m,\dots,m)$), so können wir nach Folgerung 3.1.10 dies auch als ein Kanonisierungsproblem von Punktkonfigurationen in $\mathrm{PHG}((R^k)_R)$ auffassen.

Liegen in der zu kanonisierenden Generatormatrix auch nicht fette Spalten vor (d.h. $\mu \neq (m, \ldots, m)$), so führt die Multiplikation dieser Spalten mit Einheiten zu einer nicht treuen Gruppenoperation:

5.1.22 Hilfssatz. *Für beliebiges $\Gamma \in R^{k\times n,\lambda,\mu}$ ist*

$$\mathrm{Stab}_{(R^*)^n}(\Gamma) = (R^*)^\mu := \bigtimes_{j\in[n]} (1 + \mathrm{Rad}(R)^{\mu_j}).$$

Die Untergruppe $(R^*)^\mu$ ist somit ein Normalteiler von $(R^*)^n$.

Beweis. Für beliebiges $\Gamma \in R^{k\times n,\lambda,\mu}$, $j \in [n]$ und $a \in R^*$ gilt:

$$\begin{aligned}\Gamma_{*,j}a^{-1} = \Gamma_{*,j} &\iff \Gamma_{*,j}(1-a) = 0\\ &\iff (1-a) \in \mathrm{Rad}(R)^{\mathrm{per}(\Gamma_{*,j})} = \mathrm{Rad}(R)^{\mu_j}\\ &\iff a \in 1 + \mathrm{Rad}(R)^{\mu_j}\end{aligned}$$

□

5.1.23 Bemerkung. Wieder werden wir, aus Gründen der Übersichtlichkeit, im weiteren Verlauf dieser Arbeit keine explizite Unterscheidung zwischen den Gruppenelementen aus $(R^*)^n$ und $(R^*)^n/(R^*)^\mu$ vornehmen.

Abschließend wollen wir auch noch beweisen, dass man sich bei der Definition der Operation auch bei der Komponente $\mathrm{Aut}(R)$ auf die Untergruppe der äußeren Automorphismen beschränken kann. Es sei $a \in R^*$ eine beliebige Einheit, dann ist für alle $\Gamma \in R^{k\times n,\lambda,\mu}$

$$\left(a^{-1} \cdot I_k, (a^{-1}) \cdot \mathbf{1_n}; \chi_{a\xi a^{-1}}^{a\theta a^{-1}}\right)\Gamma = a^{-1}\chi_{a\xi a^{-1}}^{a\theta a^{-1}}(\Gamma)a = a^{-1}a\Gamma a^{-1}a = \Gamma.$$

Die Operation mit einem inneren Automorphismen $\chi_{a\xi a^{-1}}^{a\theta a^{-1}} \in \mathrm{Inn}(R)$ lässt sich also durch eine Multiplikation mit der Einheit a^{-1} von links bzw. mit a von rechts an die Generatormatrix wieder ausgleichen.

5.1.24 Hilfssatz. *Die Menge*

$$I := \left\{ \left(a^{-1} \cdot I_k, (a^{-1}) \cdot \mathbf{1_n}; \chi_{a\xi a^{-1}}^{a\theta a^{-1}}\right) \mid a \in R^* \right\}$$

ist ein Normalteiler von $(\mathrm{GL}_\lambda(R) \times (R^*)^n) \rtimes \mathrm{Aut}(R)$.

Beweis. Zunächst zeigen wir, dass die Menge I abgeschlossen unter der Multiplikation ist. Wir wählen hierzu $(a^{-1} \cdot I_k, (a^{-1}) \cdot \mathbf{1_n}; \alpha), (b^{-1} \cdot I_k, (b^{-1}) \cdot \mathbf{1_n}; \beta) \in I$ beliebig. Es gilt:

$$\begin{aligned}&\left(a^{-1} \cdot I_k, (a^{-1}) \cdot \mathbf{1_n}; \alpha\right)\left(b^{-1} \cdot I_k, (b^{-1}) \cdot \mathbf{1_n}; \beta\right)\\ &= \left(a^{-1}\alpha(b^{-1}) \cdot I_k, (a^{-1}\alpha(b^{-1})) \cdot \mathbf{1_n}; \alpha\beta\right)\\ &= \left(a^{-1}ab^{-1}a^{-1} \cdot I_k, (a^{-1}ab^{-1}a^{-1}) \cdot \mathbf{1_n}; \alpha\beta\right)\\ &= \left((ab)^{-1} \cdot I_k, (ab)^{-1} \cdot \mathbf{1_n}; \chi_{(ab)^{-1}\xi ab}^{(ab)^{-1}\theta ab}\right) \in I.\end{aligned}$$

Ist nun $(B, \psi; \gamma) \in (\mathrm{GL}_\lambda(R) \times (R^*)^n) \rtimes \mathrm{Aut}(R)$ beliebig, so erhält man durch Konjugation

$$\begin{aligned}
&(B, \psi; \gamma)\big(a^{-1} \cdot I_k, (a^{-1}) \cdot \mathbf{1_n}; \alpha\big)(B, \psi; \gamma)^{-1} \\
&= (B\gamma(a^{-1}), \psi\gamma(a^{-1}); \gamma\alpha)(\gamma^{-1}(B^{-1}), \gamma^{-1}(\psi^{-1}); \gamma^{-1}) \\
&= \Big(B\gamma(a^{-1}) \cdot \big((\gamma\alpha\gamma^{-1})(B^{-1})\big), \psi\gamma(a^{-1}) \cdot \big((\gamma\alpha\gamma^{-1})(\psi^{-1})\big); \underbrace{\gamma\alpha\gamma^{-1}}_{x \mapsto \gamma(a)x\gamma(a)^{-1}}\Big) \\
&= (B\gamma(a^{-1})\gamma(a)B^{-1}\gamma(a)^{-1}, \psi\gamma(a^{-1})\gamma(a)\psi^{-1}\gamma(a)^{-1}; \gamma\alpha\gamma^{-1}) \\
&= \Big(\gamma(a)^{-1} \cdot I_k, (\gamma(a)^{-1}) \cdot \mathbf{1_n}, \chi_{\gamma(a)\xi\gamma(a)^{-1}}^{\gamma(a)\theta\gamma(a)^{-1}}\Big)
\end{aligned}$$

wieder ein Element der Untergruppe I. □

Unsere Beobachtungen führen nun zu der folgenden, abschließenden Problemformulierung, zu welcher wir dann im weiteren Verlauf der Arbeit einen Kanonisierer entwickeln möchten.

5.1.25 Satz. *Zwei Generatormatrizen $\Gamma, \Gamma' \in R^{k\times n,\lambda,\mu}$ erzeugen genau dann semilinear isometrische Codes, wenn es ein Gruppenelement*

$$(A, \varphi; \alpha, \pi) \in \big(\mathrm{GL}_\lambda(R)/N_\lambda(R) \times \big((R^*)^n/(R^*)^\mu\big)\big) \rtimes (\mathrm{Aut}_T \times S_{\mathfrak{P}_0})$$

gibt mit $A((\varphi; \pi)\alpha(\Gamma)) = \Gamma'$.

Beweis. Die Normalteiler $N_\lambda(R)$, $(R^*)^\mu$ und I lassen jede beliebige Generatormatrix $\Gamma \in R^{k\times n,\lambda,\mu}$ fix. Wir können sie also ohne Beschränkungen heraus teilen. Zur Vereinfachung des Quotienten nutzen wir anschließend die Zerlegung der Automorphismengruppe von R aus Fakt 4.2.4. □

5.1.26 Bemerkung. In dieser Beschreibung haben wir nur aus Gründen der Übersichtlichkeit darauf verzichtet, die Gruppe I vollständig zu berücksichtigen. Der eigentlich noch verfügbare Anteil $\widetilde{Z}(\xi) := \{(a \cdot I_k, a \cdot \mathbf{1_n}, \chi_\xi^{a^{-1}xa}) \in I \mid a \in Z(\xi)\} \trianglelefteq I$ des Normalteilers I geht im weiteren Verlauf dieser Arbeit nicht mehr ein. In der Implementierung des Kanonisierers haben wir aber tatsächlich mit der Gruppe

$$\Big(\big(\big((\mathrm{GL}_\lambda(R)/N_\lambda(R)) \times \big((R^*)^n/(R^*)^\mu\big)\big) \rtimes \mathrm{Aut}_T\big)/\widetilde{Z}(\xi)\Big) \rtimes S_{\mathfrak{P}_0}$$

gearbeitet.

Nach der vorangegangenen Diskussion lösen wir das Kanonisierungsproblem für lineare Codes unter der Operation der Gruppe aller semilinearen Isometrien vollständig, falls wir in der Lage sind, einen Kanonisierer $\mathrm{Can}_{G\rtimes S_{\mathfrak{P}_\circ}}^{R^{k\times n,\lambda,\mu}}$ zu der Gruppenoperation von $G \rtimes S_{\mathfrak{P}_\circ}$ mit

$$G := \big(\big(\mathrm{GL}_\lambda(R)/N_\lambda(R)\big) \times \big((R^*)^n/(R^*)^\mu\big)\big) \rtimes \mathrm{Aut}_T$$

und $S_{\mathfrak{P}_0} := \mathrm{Stab}_{S_n}(\mu)$ auf der Menge $R^{k\times n,\lambda,\mu}$ anzugeben. Hierzu folgen wir dem in Abschnitt 3.3 beschriebenen Vorgehen.

Zunächst werden wir aber weitere wichtige Vorarbeiten zu der Operation von G auf $R^{k\times n,\lambda,\mu}$ in einem gesonderten Abschnitt behandeln. Diese erlauben es uns dann, den Kanonisierer in Abschnitt 5.2 mit dem Blick für das Wesentliche zu entwickeln.

5.1.2. Die Operation von $(\mathrm{GL}_k(R) \times R^{*n}) \rtimes \mathrm{Aut}_T$

Es sei weiterhin der Umriss λ der Generatormatrizen und der Vektor $\mu \in \{1,\dots,m\}^n$ der Spaltenperioden beliebig, aber fest gewählt. Insbesondere erlauben wir in diesem Abschnitt auch, ungeordnete Vektoren $\mu \in \{1,\dots,m\}^n$ zu betrachten. Ziel ist es, die Gruppenoperation der Gruppe

$$G^{(\lambda,\mu)} := \big(\big(\mathrm{GL}_\lambda(R)/N_\lambda(R)\big) \times \big((R^*)^n/(R^*)^\mu\big)\big) \rtimes \mathrm{Aut}_T\,.$$

auf der Menge $R^{k\times n,\lambda,\mu}$ zu untersuchen. Sind die Parameter λ,μ aus dem Kontext klar, so schreiben wir auch kurz G für $G^{(\lambda,\mu)}$.

5.1.27 Definition (Totalordnung auf $R^{k\times n}$)**.** Die Menge $R^{k\times n}$ aller $(k\times n)$-Matrizen über R werden wir als lexikographisch angeordnete Menge aller n-Tupel von colexikographisch geordneten Spaltenvektoren auffassen, d.h. für alle Spaltenvektoren $v,w \in (R^k)_R$ sei

$$v < w :\Longleftrightarrow \exists\, i \in [k] \text{ mit } v_i < w_i \ \wedge\ \forall\, i < j < k : v_j = w_j.$$

und für Matrizen $A,B \in R^{k\times n}$ gelte:

$$A < B :\Longleftrightarrow \exists\, i \in [n] \text{ mit } A_{*,i} < B_{*,i} \ \wedge\ \forall\, j \in [i] : A_{*,j} = B_{*,j}.$$

Als Kanonisierungsfunktion CF_G für die Operation von G auf $R^{k\times n,\lambda,\mu}$ geben wir uns die Berechnung des minimalen Repräsentanten der Bahn fest vor, d.h. für alle $\Gamma \in R^{k\times n,\lambda,\mu}$ sei

$$\mathrm{CF}_G(\Gamma) := \min_{(A,\varphi;\alpha)\in G} A\alpha(\Gamma)\,\mathrm{diag}(\varphi)^{-1}.$$

Wir wollen zeigen, dass wir diese Kanonisierung (zumindest für eine eingeschränkte Auswahl von Eingaben) effizient implementieren können. Da das Verfahren induktiv vorgeht, werden wir auch für Teilfolgen $\mu' = \mu_{[n']}$ mit $n' \le n$ die Operation von $G^{(\lambda,\mu)}$ auf $R^{k\times n',\lambda,\mu'}$ betrachten. Hierbei bleiben bei der Multiplikation eines Gruppenelements $(A,\varphi;\alpha) \in G^{(\lambda,\mu)}$ mit $\Gamma \in R^{k\times n',\lambda,\mu'}$ die letzten $n-n'$ Einträge des Vektors φ schlicht unberücksichtigt.

Zunächst werden wir den Stabilisator $\mathrm{Stab}_G(\Gamma)$ einer beliebigen Matrix $\Gamma \in R^{k\times n,\lambda,\mu}$ untersuchen. Wir definieren hierzu

$$G^{\mathrm{lin}} := (\mathrm{GL}_\lambda(R)/N_\lambda(R)) \times \big((R^*)^n/(R^*)^\mu\big)$$

und identifizieren diese Gruppe mit dem Normalteiler $G^{\mathrm{lin}} \rtimes \{\mathrm{id}_R\} \trianglelefteq G$.

5.1.28 Hilfssatz. *Für eine beliebige Matrix $\Gamma \in R^{k\times n,\lambda,\mu}$ ist die Abbildung*

$$\begin{aligned}(.)\downarrow_{R^{*n}\rtimes \mathrm{Aut}_T} : \mathrm{Stab}_G(\Gamma) &\to (R^*)^n/(R^*)^\mu \rtimes \mathrm{Aut}_T\\ (A,\varphi;\alpha) &\mapsto (\varphi;\alpha)\end{aligned}$$

ein Monomorphismus. Definiert man die Abbildung $(.)\downarrow_{\mathrm{GL}_\lambda(R)\rtimes\mathrm{Aut}_T}$ analog, so ist diese ebenfalls ein Monomorphismus.

Beweis. Es ist nur die Aussage zur Injektivität zu beweisen. Wir nehmen an, es seien $(AN_\lambda(R),\varphi;\alpha),(A'N_\lambda(R),\varphi;\alpha)\in \mathrm{Stab}_G(\Gamma)$ beliebig. Dann ist

$$\begin{aligned}&(AN_\lambda(R),\varphi;\alpha)\Gamma = \Gamma = (A'N_\lambda(R),\varphi;\alpha)\Gamma \iff A^{-1}A'\alpha(\Gamma)=\alpha(\Gamma)\\ \iff &A^{-1}A' \in \mathrm{Stab}_{\mathrm{GL}_\lambda(R)}(\alpha(\Gamma)) = N_\lambda(R) \iff AN_\lambda(R)=A'N_\lambda(R)\end{aligned}$$ □

5.1.29 Folgerung.

- *Das Gruppenelement $(A,\varphi;\alpha)\in\mathrm{Stab}_G(\Gamma)$ ist durch die Angabe von $(\varphi;\alpha)$ beziehungsweise $(A;\alpha)$ bereits eindeutig bestimmt.*
- *Ist R kommutativ, so ist $\mathrm{Stab}_{G^{\mathrm{lin}}}(\Gamma)$ abelsch.*

Die Definitionen von $(.)\downarrow_{R^{*n}\rtimes\mathrm{Aut}_T}$ und $(.)\downarrow_{\mathrm{GL}_\lambda(R)\rtimes\mathrm{Aut}_T}$ übertragen wir auch auf ganz G. Im weiteren Verlauf sei außerdem mit $(.)\downarrow_{\mathrm{GL}_\lambda(R)}$, $(.)\downarrow_{R^{*n}}$ und $(.)\downarrow_{\mathrm{Aut}_T}$ die Projektion eines Gruppenelements $(A,\varphi;\alpha)$ auf den Matrixanteil A, die Spaltenmultiplikation φ bzw. den Automorphismenanteil α bezeichnet.

5.1.30 Definition. Zu einer Matrix $\Gamma\in R^{k\times n,\lambda,\mu}$ bezeichne $\mathfrak{p}^\Gamma$ die feinste Partition von $[k]$, so dass der Träger $\mathrm{supp}(\Gamma_{*,i}) := \{j\in[k]\mid \Gamma_{j,i}\neq 0\}$ einer jeden Spalte $\Gamma_{*,i}$, $i\in[n]$ in einem Block $P\in\mathfrak{p}^\Gamma$ enthalten ist, d.h. $\forall\, i\in[n]\ \exists\, P\in\mathfrak{p}^\Gamma : \mathrm{supp}(\Gamma_{*,i})\subseteq P$.
Zu $P\in\mathfrak{p}^\Gamma$ sei dann $\mathrm{Cols}_P(\Gamma) := \{j\in[n]\mid \mathrm{supp}(\Gamma_{*,j})\subseteq P\}$ die Menge derjenigen Koordinaten $j\in[n]$, für welche der Träger $\mathrm{supp}(\Gamma_{*,j})$ der Spalte $\Gamma_{*,j}$ in dem Block P liegt.

5.1.31 Hilfssatz. *Für eine beliebige Generatormatrix $\Gamma\in R^{k\times n,\lambda,\mu}$ und $P\in\mathfrak{p}^\Gamma$ folgt:*

$$A\in\mathrm{Stab}_G(\Gamma)\downarrow_{\mathrm{GL}_\lambda(R)} \Longrightarrow \forall\, i\in[k]\setminus P, j\in P : A_{i,j}=0.$$

Beweis. Es sei $P=\{p_0,\ldots,p_{k'-1}\}$, wobei die Zeilenindizes $p_0<\ldots<p_{k'-1}$ aufsteigend nummeriert seien. Zunächst sieht man leicht ein, dass die Projektion von Γ, auf die durch P ausgezeichnete Zeilen- und Spaltenmenge, eine Generatormatrix $\Gamma'=\Gamma_{P,\mathrm{Cols}_P(\Gamma)}$ vom Umriss $(\lambda_{p_0},\ldots,\lambda_{p_{k'-1}})$ ergibt. Insbesondere sind die Zeilen von Γ' also unabhängig.

Nun sei $(A,\varphi;\alpha) \in \mathrm{Stab}_G(\Gamma)$ und $i \in [k] \setminus P$ beliebig und wir definieren $n' := |\mathrm{Cols}_P(\Gamma)|$. Aus der Stabilisatoreigenschaft $(A,\varphi;\alpha)\Gamma = \Gamma$ folgern wir zunächst:

$$\begin{aligned}\mathbf{0}_{\mathbf{n}'} = \Gamma_{i,\mathrm{Cols}_P(\Gamma)} &= \sum_{j=0}^{k-1} A_{i,j}\alpha(\underbrace{\Gamma_{j,\mathrm{Cols}_P(\Gamma)}}_{j\notin P \Longrightarrow \Gamma_{j,\mathrm{Cols}_P(\Gamma)}=\mathbf{0}_{\mathbf{n}'}}) \operatorname{diag}\big(\varphi_{\mathrm{Cols}_P(\Gamma)}\big)^{-1} \\ &= \sum_{\ell=0}^{k'-1} A_{i,p_\ell}\alpha(\Gamma'_{\ell,*}) \operatorname{diag}\big(\varphi_{\mathrm{Cols}_P(\Gamma)}\big)^{-1}.\end{aligned}$$

Da sowohl die Anwendung des Ringautomorphismus α als auch die Multiplikation mit $\operatorname{diag}\big(\varphi_{\mathrm{Cols}_P(\Gamma)}\big)^{-1}$ die Unabhängigkeit der Zeilen von Γ' erhält, ist $A_{i,j} \in \mathrm{Rad}(R)^{\lambda_j}$ für alle $j \in P$. Nun stellt aber die Matrix A nur einen Nebenklassenrepräsentanten der Nebenklasse $AN_\lambda(R)$ dar und wir können ohne Beschränkung der Allgemeinheit $A_{i,j} = 0$ setzen, ohne die Nebenklasse damit zu verlassen. □

5.1.32 Folgerung. *Sortiert man die Zeilen der Generatormatrix Γ mittels einer Permutation $\pi \in S_k$ nach den Blöcken $P \in \mathfrak{p}^\Gamma$, so haben die Matrizen $P^{(\pi)}AP^{(\pi)^{-1}}$ für $A \in \mathrm{Stab}_G(\Gamma)\downarrow_{\mathrm{GL}_\lambda(R)}$ eine Blockdiagonalgestalt.*

Diese Beobachtung fließt nun auch in den nächsten Hilfssatz ein.

5.1.33 Hilfssatz. *Für eine beliebige Generatormatrix $\Gamma \in R^{k\times n,\lambda,\mu}$ und $P \in \mathfrak{p}^\Gamma$ definiert die Abbildung*

$$\Psi^{(P,\Gamma)} : \mathrm{Stab}_{G^{\mathrm{lin}}}(\Gamma) \to \mathrm{Stab}_{G^{\mathrm{lin}}}(\Gamma), \quad (A,\varphi) \mapsto (A^{(P)},\varphi^{(P)})$$

mit

$$A^{(P)}_{i,*} := \begin{cases} A_{i,*}, & \text{falls } i \in P \\ e_i, & \text{falls } i \notin P \end{cases} \quad \text{für } i \in [k]$$

und

$$\varphi^{(P)}_j := \begin{cases} \varphi_j, & \text{falls } \mathrm{supp}(\Gamma_{*,j}) \subseteq P \\ 1, & \text{sonst} \end{cases} \quad \text{für } j \in [n]$$

einen Gruppenhomomorphismus.

Beweis. Zunächst zeigen wir, dass für $P \in \mathfrak{p}^\Gamma$ und $(A,\varphi) \in \mathrm{Stab}_{G^{\mathrm{lin}}}(\Gamma)$ das Gruppenelement $(A^{(P)},\varphi^{(P)})$ ebenfalls im Stabilisator liegt. Es sei hierzu $i \in [k]$ und $j \in [n]$ beliebig. Es folgt:

$$\big((A^{(P)},\varphi^{(P)})\Gamma\big)_{i,j} = \begin{cases} A_{i,*}\Gamma_{*,j}\Big(\varphi^{(P)}_j\Big)^{-1} = \Gamma_{i,j}\varphi_j\Big(\varphi^{(P)}_j\Big)^{-1}, & \text{falls } i \in P \\ e_i\Gamma_{*,j}\Big(\varphi^{(P)}_j\Big)^{-1} = \Gamma_{i,j}\Big(\varphi^{(P)}_j\Big)^{-1}, & \text{sonst} \end{cases}$$

Gilt nun $i \in P \wedge \operatorname{supp}(\Gamma_{*,j}) \not\subseteq P$ oder $i \notin P \wedge \operatorname{supp}(\Gamma_{*,j}) \subseteq P$, so ist in beiden Fällen $\Gamma_{i,j} = 0 = \big((A^{(P)}, \varphi^{(P)})\Gamma\big)_{i,j}$. In den beiden verbleibenden Fällen ergibt sich die Gleichheit $\Gamma_{i,j} = \big((A^{(P)}, \varphi^{(P)})\Gamma\big)_{i,j}$ aus der Definition von $\varphi_j^{(P)} = \varphi_j$ für $j \in \operatorname{Cols}_P(\Gamma)$ bzw. $\varphi_j^{(P)} = 1$ im anderen Fall.

Mit der Eindeutigkeit aus Hilfssatz 5.1.28 zeigt man abschließend leicht, dass die Abbildungsvorschrift mit der Multiplikation verträglich ist, d.h. für alle $(A, \varphi), (B, \psi) \in \operatorname{Stab}_{G^{\mathrm{lin}}}(\Gamma)$ gilt

$$\Psi^{(P,\Gamma)}((A,\varphi)(B,\psi)) = \Psi^{(P,\Gamma)}(A,\varphi) \cdot \Psi^{(P,\Gamma)}(B,\psi). \qquad \square$$

5.1.34 Folgerung. *Die Gruppe*

$$\operatorname{Stab}_{G^{\mathrm{lin}}}(\Gamma) = \prod_{P \in \mathfrak{p}^\Gamma} \Psi^{(P,\Gamma)}(\operatorname{Stab}_{G^{\mathrm{lin}}}(\Gamma))$$

lässt sich über ein inneres direktes Produkt der Normalteiler $\Psi^{(P,\Gamma)}(\operatorname{Stab}_{G^{\mathrm{lin}}}(\Gamma))$ *ausdrücken.*

Beweis. Aus dem Hilfssatz 5.1.28 folgt die Gleichheit

$$\Psi^{(P,\Gamma)}(\operatorname{Stab}_{G^{\mathrm{lin}}}(\Gamma)) = \bigcap_{Q \in \mathfrak{p}^\Gamma : P \neq Q} \ker\big(\Psi^{(Q,\Gamma)}\big).$$

Somit ist das Bild $\Psi^{(P,\Gamma)}(\operatorname{Stab}_{G^{\mathrm{lin}}}(\Gamma))$ ebenfalls ein Normalteiler. Die Zerlegung in das angegebene direkte Produkt folgt ebenfalls aus dem Hilfssatz 5.1.28. $\square$

Nach diesen allgemeinen Aussagen über den Stabilisator einer Matrix $\Gamma \in R^{k \times n, \lambda, \mu}$ wollen wir nun den Sonderfall betrachten, dass Γ eine Generatormatrix in reduzierter Zeilenstufenform ist, für welche zusätzlich $\mathrm{CF}_G(\Gamma) = \Gamma$ gilt.

5.1.35 Hilfssatz. *Ist* $\Gamma \in R^{k \times n, \lambda, \mu}$ *eine Matrix in reduzierter Zeilenstufenform und* $\mathrm{CF}_G(\Gamma) = \Gamma$*, so gilt für beliebige* $i, j \in [k]$ *mit* $i \neq j$ *und* $(i > j \vee \lambda_i = \lambda_j)$*:*

$$A \in \operatorname{Stab}_G(\Gamma)\!\downarrow_{\mathrm{GL}_\lambda(R)} \implies A_{i,j} = 0$$

Beweis. Wir führen die Argumentation über eine Induktion nach n. Ist $n = 1$, so ist auch $k = 1$ und die Aussage trivial.

Für $n > 1$ sind zwei Fälle zu unterscheiden: Ist die letzte Spalte keine Pivotspalte der Generatormatrix Γ, so ist $\Gamma_{*,[n-1]}$ ebenfalls eine Generatormatrix in reduzierter Zeilenstufenform mit $\mathrm{CF}_G(\Gamma_{*,[n-1]}) = \Gamma_{*,[n-1]}$. Da der Stabilisator $\operatorname{Stab}_G(\Gamma)$ eine Untergruppe des Stabilisators $\operatorname{Stab}_G\big(\Gamma_{*,[n-1]}\big)$ ist, folgt die Aussage aus der Induktionsvoraussetzung.

Es bleibt also der Fall zu betrachten, dass die letzte Spalte eine Pivotspalte ist, d.h. $\Gamma = \begin{pmatrix} \Gamma_{[k-1],[n-1]} & \Gamma_{[k-1],n-1} \\ \mathbf{0_{n-1}} & \theta^{m-\lambda_{k-1}} \end{pmatrix}$. Die Teilmatrix $\Gamma' := \Gamma_{[k-1],[n-1]} \in R^{(k-1)\times(n-1),\lambda_{[k-1]},\mu_{[n-1]}}$ ist in reduzierter Zeilenstufenform und es ist $\mathrm{CF}_{G^{(\lambda_{[k-1]},\mu_{[n-1]})}}(\Gamma') = \Gamma'$. Aus der Gleichung

$$\begin{aligned}\mathbf{0_{n-1}} = \Gamma_{k-1,[n-1]} &= \Gamma_{k-1,[n-1]}\operatorname{diag}(\varphi_{n-1}) = A_{k-1,*}\alpha(\Gamma_{*,[n-1]}) \\ &= \sum_{j=0}^{k-2} A_{k-1,j}\alpha(\Gamma_{j,[n-1]}) = \sum_{j=0}^{k-2} A_{k-1,j}\alpha(\Gamma'_{j,*})\end{aligned}$$

und der Unabhängigkeit der Zeilen der Matrix $\alpha(\Gamma')$ schließen wir wie oben zunächst $A_{k-1,j} \in \mathrm{Rad}(R)^{\lambda_j}$ für alle $j \in [k-1]$. Da aber A tatsächlich die Nebenklasse $AN_\lambda(R)$ repräsentiert, können wir auch $A_{k-1,j} = 0$ setzen. Für die weiteren Einträge $A_{i,j}$, $i \in [k-1]$ der Spalte $A_{*,j}$ ergibt sich die Aussage über die Induktionsannahme, denn es ist $A_{[k-1],[k-1]} \in \left(\mathrm{Stab}_{G^{(\lambda_{[k-1]},\mu_{[n-1]}}}(\Gamma')\right)\!\downarrow_{\mathrm{GL}_{\lambda_{[k-1]}}(R)}$.

Somit bleibt die Aussage für die letzte Spalte $A_{*,k-1}$ zu beweisen. Es sei also $i \in [k-1]$ mit $\lambda_i = \lambda_{k-1}$ beliebig. Da Γ eine Generatormatrix ist, liegt der Eintrag $\Gamma_{i,n-1}$ in $\mathrm{Rad}(R)^{m-\lambda_i}$. Andererseits ist er aber auch modulo dem Pivotelement $\theta^{m-\lambda_{k-1}} = \theta^{m-\lambda_i}$ reduziert. Wir schließen, dass $\Gamma_{i,n-1} = 0$ ist, und erhalten hiermit

$$\begin{aligned}0 = \Gamma_{i,n-1}\varphi_{n-1} &= A_{i,*}\alpha(\Gamma_{*,n-1}) \\ &= \sum_{j=0}^{i-1} \underbrace{A_{i,j}}_{=0,\text{ da } j<i} \alpha(\Gamma_{j,n-1}) + \sum_{j=i}^{k-2} \underbrace{A_{i,j}}_{=0,\text{ da } \lambda_i\geq\lambda_j\geq\lambda_{k-1}} \alpha(\Gamma_{j,n-1}) + A_{i,k-1}\alpha(\theta^{m-\lambda_{k-1}}) \\ &= A_{i,k-1}\alpha(\theta^{m-\lambda_{k-1}}).\end{aligned}$$

Hieraus ergibt sich $A_{i,k-1} \in \mathrm{Rad}(R)^{\lambda_{k-1}}$ und wie oben können wir den Eintrag $A_{i,k-1} = 0$ setzen. □

5.1.36 Folgerung. *Beschreiben wir die Matrix $A \in \mathrm{Stab}_G(\Gamma)\!\downarrow_{\mathrm{GL}_\lambda(R)}$ über die Blockmatrizen $A^{(i,j)} \in R^{k_i^\lambda \times k_j^\lambda}$ für $i,j \in [m]$ gemäß der Definition 5.1.1, so ist*

$$A = \begin{pmatrix} A^{(0,0)} & A^{(0,1)} & \dots & A^{(0,m-1)} \\ & A^{(1,1)} & \dots & A^{(1,m-1)} \\ & & \ddots & \vdots \\ & & & A^{(m-1,m-1)} \end{pmatrix} \quad \text{mit Diagonalmatrizen } A^{(i,i)} \in \mathrm{GL}_{k_i^\lambda}(R).$$

Im Folgenden bezeichnen wir wieder mit S den Koeffizientenring von R, welcher von der Teichmüller-Menge T erzeugt wird. Weiter sei $e \in [r]$ der eindeutige Exponent des Frobenius-Automorphismus $\tau \in \mathrm{Aut}(S)$, für welchen $\tau^e(\xi)\theta = \theta\xi$ gilt.

5.1.37 Hilfssatz. *Ist $\Gamma \in R^{k\times n,\lambda,\mu}$ eine Matrix in reduzierter Zeilenstufenform und $\mathrm{CF}_G(\Gamma) = \Gamma$, so gilt für jedes beliebige $P \in \mathfrak{p}^\Gamma$:*

1. *Zu* $i \in P$ *und* $j \in \mathrm{Cols}_P(\Gamma)$ *gibt es ein* $x \in \left[\frac{r}{\mathrm{ggT}(r,e)}\right]$ *:* $\overline{A_{i,i}} = \tau^{xe}(\overline{\varphi_j})$ *für alle* $(A,\varphi) \in \mathrm{Stab}_{G^{\mathrm{lin}}}(\Gamma)$.
2. *Zu* $i,j \in P$ *gibt es ein* $x \in \left[\frac{r}{\mathrm{ggT}(r,e)}\right]$ *:* $\overline{A_{i,i}} = \tau^{xe}(\overline{A_{j,j}})$ *für alle* $(A,\varphi) \in \mathrm{Stab}_{G^{\mathrm{lin}}}(\Gamma)$.
3. *Zu* $i,j \in \mathrm{Cols}_P(\Gamma)$ *gibt es ein* $x \in \left[\frac{r}{\mathrm{ggT}(r,e)}\right]$ *:* $\overline{\varphi_i} = \tau^{xe}(\overline{\varphi_j})$ *für alle* $(A,\varphi) \in \mathrm{Stab}_{G^{\mathrm{lin}}}(\Gamma)$.

Beweis. Die zweite und dritte Behauptung ergeben sich offensichtlich sofort aus der Ersten. Wir beweisen diese über eine Induktion nach der Länge n des Codes. Für $n = 1$ ist sie trivial.

Es sei $n \geq 2$ beliebig und $\Gamma \in R^{k\times n,\lambda,\mu}$ eine Generatormatrix in reduzierter Zeilenstufenform mit $\mathrm{CF}_G(\Gamma) = \Gamma$. Wir definieren $\widetilde{\Gamma} := \Gamma_{*,[n-1]}$. Die Blöcke $\widetilde{P} \in \mathfrak{p}^{\widetilde{\Gamma}}$ mit $\widetilde{P} \cap \mathrm{supp}(\Gamma_{*,n-1}) = \emptyset$ treten auch in $\mathfrak{p}^{\Gamma}$ auf. Da der Index $n-1$ nicht in $\mathrm{Cols}_{\widetilde{P}}(\Gamma)$ liegt, erhalten wir die Aussage für alle $i \in \widetilde{P}$ über die Induktionsvoraussetzung.

In $\mathfrak{p}^{\Gamma}$ gibt es genau einen weiteren Block P, welcher aus der Vereinigung der Blöcke $\widetilde{P} \in \mathfrak{p}^{\widetilde{\Gamma}}$ mit $\widetilde{P} \cap \mathrm{supp}(\Gamma_{*,n-1}) \neq \emptyset$ hervorgeht. Für die Indizes $i \in P$ und $j = n-1$ ist die Behauptung noch zu beweisen.

1. Fall: Wir untersuchen zunächst den Fall, dass $\widetilde{\Gamma}$ einen linearen Code mit gleichem Umriss λ erzeugt, d.h. die letzte Spalte ist keine Pivotspalte. Zu einem Block $\widetilde{P} \in \mathfrak{p}^{\widetilde{\Gamma}}$ mit $\widetilde{P} \subseteq P$ definieren wir $\ell := \max(\widetilde{P} \cap \mathrm{supp}(\Gamma_{*,n-1}))$. Ist $(A,\varphi) \in \mathrm{Stab}_{G^{\mathrm{lin}}}(\Gamma)$ beliebig, so folgt

$$\begin{aligned}
\Gamma_{\ell,n-1} &= \left(\sum_{j=\ell}^{k-1} A_{\ell,j}\Gamma_{j,n-1}\right)\varphi_{n-1}^{-1} \\
&= \left(\sum_{j\in\widetilde{P}:j\geq\ell} A_{\ell,j}\underbrace{\Gamma_{j,n-1}}_{=0\Longleftrightarrow j>\ell} + \sum_{j\in[k]\setminus\widetilde{P}:j\geq\ell}\underbrace{A_{\ell,j}}_{=0}\Gamma_{j,n-1}\right)\varphi_{n-1}^{-1} = A_{\ell,\ell}\Gamma_{\ell,n-1}\varphi_{n-1}^{-1} \\
&\equiv A_{\ell,\ell}\tau^{e\cdot\mathrm{ht}(\Gamma_{\ell,n-1})}(\varphi_{n-1}^{-1})\Gamma_{\ell,n-1} \mod \mathrm{Rad}(R)^{\mathrm{ht}(\Gamma_{\ell,n-1})+1}
\end{aligned}$$

und damit $A_{\ell,\ell}\tau^{e\cdot\mathrm{ht}(\Gamma_{\ell,n-1})}(\varphi_{n-1}^{-1}) \in 1+\mathrm{Rad}(R)$. Die kleinste positive Zahl $x \equiv \mathrm{ht}(\Gamma_{\ell,n-1}) \mod \frac{r}{\mathrm{ggT}(r,e)}$ ist also unabhängig von der Wahl des Elements $(A,\varphi) \in \mathrm{Stab}_{G^{\mathrm{lin}}}(\Gamma)$. Für alle weiteren $i \in P$ benutzt man die Induktionsvoraussetzung und Punkt 2. der Behauptung.

2. Fall: Im zweiten zu untersuchenden Fall ist die letzte Spalte von Γ eine Pivotspalte und daher $\Gamma_{k-1,n-1} = \theta^{m-\lambda_{k-1}}$. Die Einträge $A_{k-1,k-1}$ und φ_{n-1} eines Gruppenelements $(A,\varphi) \in \mathrm{Stab}_{G^{\mathrm{lin}}}(\Gamma)$ erfüllen die Gleichung $\theta^{m-\lambda_{k-1}} = A_{k-1,k-1}\theta^{m-\lambda_{k-1}}\varphi_{n-1}^{-1}$. Dies beweist die Aussage 1. im Spezialfall $i = k-1$ und $j = n-1$.

Um die Aussage für beliebiges $i \in P$ und $j = n-1$ zu zeigen, wählen wir zunächst ein $\widetilde{P} \in \mathfrak{p}^{\widetilde{\Gamma}_{[k-1],*}}$ mit $\widetilde{P} \cap \operatorname{supp}(\Gamma_{*,n-1}) \neq \emptyset$ und setzen wieder $\ell := \max\Big(\widetilde{P} \cap \operatorname{supp}(\Gamma_{*,n-1})\Big)$ gleich dem größten Zeilenindex im Schnitt des Trägers mit dem Block $\widetilde{P}$. Es folgt

$$0 \neq \Gamma_{\ell,n-1} = \left(\sum_{j=\ell}^{k-1} A_{\ell,j}\Gamma_{j,n-1}\right)\varphi_{n-1}^{-1} = (A_{\ell,\ell}\Gamma_{\ell,n-1} + A_{\ell,k-1}\theta^{m-\lambda_{k-1}})\varphi_{n-1}^{-1}.$$

Wäre nun $\lambda_{k-1} = m$, so ist der Eintrag $\Gamma_{\ell,n-1}$ modulo dem Pivotelement $\Gamma_{k-1,n-1} = 1$ reduziert und damit gleich 0. Dies ist aber ein Widerspruch zu unserer Wahl von ℓ. Aus dem gleichen Grund gilt auch $\Gamma_{\ell,n-1} \notin \operatorname{Rad}(R)^{m-\lambda_{k-1}}$. Somit ist in dieser Situation ebenfalls $A_{\ell,\ell}\tau^{e\cdot\operatorname{ht}(\Gamma_{\ell,n-1})}(\varphi_{n-1}^{-1}) \in 1 + \operatorname{Rad}(R)$.

Die Aussagen für die weiteren, nicht diskutierten Werte $i \in P$ und $j \in \operatorname{Cols}_P(\Gamma)$ ergeben sich dann wieder über die Induktionsannahme. □

5.1.38 Folgerung. *Zu $P \in \mathfrak{p}^\Gamma$ definieren wir*

$$a_P(\Gamma) := \min\Big\{a \in \mathbb{N} \mid a > 0 \wedge \exists\, i \in P,\ (A,\varphi) \in \operatorname{Stab}_{G^{\mathrm{lin}}}(\Gamma) : \overline{A_{i,i}} = \overline{\xi}^a\Big\}.$$

Dann ist für alle $(B,\psi) \in \operatorname{Stab}_{G^{\mathrm{lin}}}(\Gamma)$: $\overline{B_{j,j}}, \overline{\psi_\ell} \in \left\langle \overline{\xi}^{a_P(\Gamma)} \right\rangle$ für alle $j \in P$ und alle $\ell \in \operatorname{Cols}_P(\Gamma)$.

Beweis. Es sei $(A,\varphi) \in \operatorname{Stab}_{G^{\mathrm{lin}}}(\Gamma)$ ein Element des Stabilisators, welches für den Index $i \in P$ den Wert $\overline{A_{i,i}} = \overline{\xi}^{a_P(\Gamma)}$ annimmt. Durch Potenzieren von (A,φ) zeigt man leicht, dass $a_P(\Gamma)$ ein Teiler von $q-1$ ist.

Wir nehmen nun an, es gäbe ein weiteres Element $(B,\psi) \in \operatorname{Stab}_{G^{\mathrm{lin}}}(\Gamma)$ mit $\overline{B_{j,j}} = \overline{\xi}^b$ für $j \in P$ oder $\overline{\psi_\ell} = \overline{\xi}^b$ für ein $\ell \in \operatorname{Cols}_P(\Gamma)$, so dass $\overline{\xi}^b \notin \langle\overline{\xi}^{a_P(\Gamma)}\rangle$. Mit dem vorausgegangenen Hilfssatz ist also $\overline{B_{i,i}} = \overline{\xi}^{bp^{ex}}$ für ein $x \in \left[\frac{r}{\operatorname{ggT}(r,e)}\right]$. Da $a_P(\Gamma)$ den Exponenten b nicht teilt, existieren ganze Zahlen y, z mit

$$ya_P(\Gamma) + zbp^{ex} = \operatorname{ggT}(a_P(\Gamma), bp^{ex}) = \operatorname{ggT}(a_P(\Gamma), b) < a_P(\Gamma).$$

Das Gruppenelement $(A,\varphi)^y(B,\psi)^z \in \operatorname{Stab}_{G^{\mathrm{lin}}}(\Gamma)$ widerspricht dann der Minimalität von $a_P(\Gamma)$. □

Das nächste Beispiel zeigt, dass tatsächlich nicht triviale Teiler $a_P(\Gamma)$ von $q-1$ auftreten werden.

5.1.39 Beispiel. Es ist $\Gamma := \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & X \end{pmatrix}$ eine Generatormatrix in reduzierter Zeilenstufenform eines linearen Codes über dem nicht kommutativen Kettenring $R = \mathbb{F}_{16}[X,\tau^2]/(X^2)$. Die Partition $\mathfrak{p}^\Gamma$ ist offensichtlich gleich $\{\{0,1\}\}$. Außerdem kann man sich leicht überlegen, dass $\operatorname{CF}_G(\Gamma) = \Gamma$ gilt.

Wegen $\lambda_0 = \lambda_1$, sind die Matrizen A für die Gruppenelemente $(A, \varphi; \alpha) \in \mathrm{Stab}_G(\Gamma)$ Diagonalmatrizen. Aus der Spalte $\Gamma_{*,2}$ lässt sich leicht die Gleichheit $A_{0,0} = A_{1,1}$ der Einträge der Diagonalen folgern. Ist nun $\alpha = \mathrm{id}$, so erhalten wir aus der letzten Spalte die beiden Gleichungen $A_{1,1} = A_{0,0} = \varphi_3$ und $X = A_{1,1} X \varphi_3^{-1} = A_{1,1} \varphi_3^{-4} X = A_{1,1}^{-3} X$. Dies zwingt die Diagonaleinträge der Matrix A in die Untergruppe $\langle \xi^5 \rangle + \mathrm{Rad}(R)$.

Insgesamt ergibt sich, dass der Stabilisator $\mathrm{Stab}_G(\Gamma)$ von den Gruppenelementen

- $((1 + aX) \cdot I_2, (1 + aX) \cdot \mathbf{1_4};\ \mathrm{id}_R)$ mit $a \in \{1, \xi, \xi^2, \xi^3\} \subseteq \mathbb{F}_{16}$ beliebig,
- $(\xi^5 \cdot I_2, \xi^5 \cdot \mathbf{1_4};\ \mathrm{id}_R)$
- und dem Frobenius-Automorphismus $\left(I_2, \mathbf{1_4};\ \chi_{\xi^2}^X\right)$ des Koeffizientenrings $\mathbb{F}_{16}$

erzeugt wird. Berücksichtigt man auch noch, dass ξ^5 die Gruppe aller zentralen Einheiten erzeugt, so ist der zweite Generator redundant.

5.1.40 Folgerung. *Ist $\Gamma \in R^{k \times n, \lambda, \mu}$ eine Matrix in reduzierter Zeilenstufenform und $\mathrm{CF}_G(\Gamma) = \Gamma$, so hat der Normalteiler $\Psi^{(P,\Gamma)}(\mathrm{Stab}_{G^{\mathrm{lin}}}(\Gamma))$ die Kardinalität $p^{\ell_P(\Gamma)} \cdot \frac{q-1}{a_P(\Gamma)}$ für ein $\ell_P(\Gamma) \in \mathbb{N}$ und eine Normalreihe*

$$\{1_G\} =: G_0^{(P)} \overset{\mathbb{Z}_p}{\lhd} G_1^{(P)} \overset{\mathbb{Z}_p}{\lhd} G_2^{(P)} \overset{\mathbb{Z}_p}{\lhd} \ldots \overset{\mathbb{Z}_p}{\lhd} G_{\ell_P(\Gamma)}^{(P)} \overset{\mathbb{Z}_x}{\trianglelefteq} \Psi^{(P,\Gamma)}(\mathrm{Stab}_{G^{\mathrm{lin}}}(\Gamma)) \tag{5.1}$$

mit zyklischen Faktoren. Dabei ist $x := \frac{q-1}{a_P(\Gamma)}$.

Beweis. Folgt sofort aus der vorangegangenen Folgerung und Hilfssatz 5.1.28. □

Es sei $\Gamma \in R^{k \times n, \lambda, \mu}$ eine Matrix in reduzierter Zeilenstufenform mit $\mathrm{CF}_G(\Gamma) = \Gamma$. Weiter sei $\mathfrak{p}^\Gamma = \{P^{(0)}, \ldots, P^{(x-1)}\}$. Wir fassen nun kurz zusammen, welche Aussagen wir über den Stabilisator bereits bewiesen haben: Es ist

$$\{1_G\} \trianglelefteq \mathrm{Stab}_{G^{\mathrm{lin}}}(\Gamma) \trianglelefteq \mathrm{Stab}_G(\Gamma).$$

und den ersten Normalteiler können wir zunächst über die Folgerung 5.1.34 wie folgt verfeinern:

$$\begin{aligned}\{1_G\} \trianglelefteq \Psi^{(P^{(0)},\Gamma)}(\mathrm{Stab}_{G^{\mathrm{lin}}}(\Gamma)) \trianglelefteq \prod_{i=0}^{1} \Psi^{(P^{(i)},\Gamma)}(\mathrm{Stab}_{G^{\mathrm{lin}}}(\Gamma)) \trianglelefteq \ldots \\ \trianglelefteq \prod_{i=0}^{x-1} \Psi^{(P^{(i)},\Gamma)}(\mathrm{Stab}_{G^{\mathrm{lin}}}(\Gamma)) = \mathrm{Stab}_{G^{\mathrm{lin}}}(\Gamma)\end{aligned}$$

Jeder Einzelschritt lässt sich dann weiter mit der Normalreihe (5.1) verfeinern. Somit kennen wir für $\mathrm{Stab}_{G^{\mathrm{lin}}}(\Gamma)$ eine Normalreihe mit zyklischen Faktoren. Den Normalteiler

$\mathrm{Stab}_{G^{\mathrm{lin}}}(\Gamma) \trianglelefteq \mathrm{Stab}_G(\Gamma)$ verfeinern wir mit der Kette (4.1) zu:

$$\begin{aligned}\mathrm{Stab}_{G^{\mathrm{lin}}}(\Gamma) \trianglelefteq \mathrm{Stab}_{G^{\mathrm{lin}} \rtimes \mathrm{Aut}_\xi^{(r-1,m-1)}}(\Gamma) \trianglelefteq \ldots \trianglelefteq \mathrm{Stab}_{G^{\mathrm{lin}} \rtimes \mathrm{Aut}_\xi^{(0,m-1)}}(\Gamma) \trianglelefteq \\ \ldots \trianglelefteq \mathrm{Stab}_{G^{\mathrm{lin}} \rtimes \mathrm{Aut}_\xi^{(0,2)}}(\Gamma) \trianglelefteq \mathrm{Stab}_{G^{\mathrm{lin}} \rtimes \mathrm{Aut}_\xi}(\Gamma) \trianglelefteq \mathrm{Stab}_{G^{\mathrm{lin}} \rtimes \mathrm{Aut}_T}(\Gamma) = \mathrm{Stab}_G(\Gamma)\end{aligned} \tag{5.2}$$

5.1.41 Bemerkung. Insgesamt können wir also davon ausgehen, dass wir eine Normalreihe zu $\mathrm{Stab}_G(\Gamma)$ mit zyklischen Faktoren kennen. In dieser Normalreihe sind höchstens $|\mathfrak{p}^\Gamma| + 2$ nicht triviale Faktoren von einer Ordnung ungleich p.

5.1.42 Bemerkung. Es sei $\Gamma \in R^{k\times n,\lambda,\mu}$ eine Matrix in reduzierter Zeilenstufenform mit $\mathrm{CF}_G(\Gamma) = \Gamma$. Weiter sei $\mathfrak{p}^\Gamma = \{P^{(0)}, \ldots, P^{(x-1)}\}$. Wir werden davon ausgehen, dass wir ein Erzeugendensystem $(E^{(P^{(0)})}, \ldots, E^{(P^{(x-1)})}, E^{(\mathrm{Aut})})$ der Gruppe $\mathrm{Stab}_G(\Gamma)$ zur Verfügung haben, welches an die oben beschriebenen Normalreihe angepasst ist. Dies bedeutet, dass:

- für einen Block $P \in \mathfrak{p}^\Gamma$ das Erzeugendensystem $E^{(P)} = \left(E_0^{(P)}, \ldots, E_{\ell_P(\Gamma)}^{(P)}\right)$ von $\Psi^{(P,\Gamma)}(\mathrm{Stab}_{G^{\mathrm{lin}}}(\Gamma))$ derart gewählt wird, dass $G_i^{(P)} := \langle E_{[i]}^{(P)} \rangle$ für alle $i \in [\ell_P(\Gamma) + 1]$ eine Normalreihe (5.1) bildet; und
- die Folge $E^{(\mathrm{Aut})}$ in diesem Sinne je einen weiteren Erzeuger gemäß der Normalreihe (5.2) bereitstellt.

Mit diesen Vorbereitungen können wir nun schließlich einen Algorithmus zur induktiven Berechnung von $\mathrm{CF}_G(\Gamma)$ angeben. Da wir obige Eigenschaften des Stabilisators einer Generatormatrix in reduzierter Zeilenstufenform benutzen wollen, müssen wir aber die Menge der zulässigen Eingaben bzw. die G-Menge, auf welcher wir die Operation betrachten, weiter einschränken. Wir wollen nun nur noch auf umrisstreuen Generatormatrizen operieren. Wir haben bereits bewiesen, dass die Eigenschaft, umrisstreu zu sein, G-invariant ist. Damit ist diese Einschränkung hinsichtlich einer wohldefinierten Gruppenoperation auch zulässig.

Für das weitere Vorgehen ist der nachfolgende Hilfssatz von zentraler Bedeutung. Er liefert uns später die Voraussetzungen zur Definition des Induktionsschritts bei der algorithmischen Beschreibung der Kanonisierung $\mathrm{CF}_G^{R^{k\times n,\lambda,\mu}}$.

5.1.43 Hilfssatz. *Es sei $\Gamma \in R^{k\times n,\lambda,\mu}$ eine Generatormatrix, so dass $\mathrm{CF}_G(\Gamma_{*,[n-1]}) = \Gamma_{*,[n-1]}$ in reduzierter Zeilenstufenform ist. Zu $i \in [k]$ und $x \in [m+1]$ definieren wir*

$$G_{\Gamma,(i,x)} := \left\{ (A, \varphi; \alpha) \in \mathrm{Stab}_G\left(\Gamma_{*,[n-1]}\right) \;\middle|\; \begin{array}{l} \left(A\alpha(\Gamma_{*,n-1})\varphi_{n-1}^{-1}\right)_j = \Gamma_{j,n-1}, \forall\ j > i \\ \left(A\alpha(\Gamma_{*,n-1})\varphi_{n-1}^{-1}\right)_i - \Gamma_{i,n-1} \in \mathrm{Rad}(R)^x \end{array} \right\}.$$

Dann ist $G_{\Gamma,(i,0)} = G_{\Gamma,(i+1,m)}$ und $G_{\Gamma,(i,x+1)} \leq G_{\Gamma,(i,x)}$ für alle $x \in [m]$.

Beweis. Die Gleichheit von $G_{\Gamma,(i,0)} = G_{\Gamma,(i+1,m)}$ ist sofort klar. Die andere Aussage zeigen wir über eine Induktion über die Paare $(i,x) \in [k] \times [m+1]$ ausgehend von dem Paar $(k-1,0)$. Für dieses beobachtet man leicht, dass $G_{\Gamma,(k-1,0)} = \mathrm{Stab}_G\big(\Gamma_{*,[n-1]}\big)$ gilt.

Nun werden wir schrittweise den Parameter x bis zum Wert $x = m$ erhöhen, anschließend zu $(i-1,0)$ übergehen und dort wieder induktiv bis zum Wert $(i-1,m)$ vordringen. Es sei also nun $(i,x) \in [k] \times [m]$ beliebig. Die Aussage werden wir über die Angabe eines Gruppenhomomorphismus

$$\begin{aligned}\Phi^{(i,x,\Gamma)} : G_{\Gamma,(i,x)} &\to \mathbb{F}_q \rtimes \big(\mathbb{F}_q^* \rtimes \mathrm{Aut}(\mathbb{F}_q)\big)\\ (A,\varphi;\alpha) &\mapsto \left(\overline{\mathrm{coeff}}^{(x)}(((A,\varphi;\alpha)\Gamma - \Gamma)_{i,n-1}), \frac{\overline{\mathrm{coeff}}^{(x)}(\alpha(\theta^x)) \cdot \overline{A_{i,i}}}{\tau^{ex}(\overline{\varphi_{n-1}})}, \overline{\alpha}\right)\end{aligned}$$

verifizieren. Die Multiplikation im Bildbereich ist hierbei über die Definition

$$(a_0,a_1,\alpha)\cdot(b_0,b_1,\beta) := (a_0 + a_1\cdot\alpha(b_0), a_1\cdot\alpha(b_1), \alpha\circ\beta)$$

für alle $(a_0,a_1,\alpha),(b_0,b_1,\beta) \in \mathbb{F}_q \rtimes \big(\mathbb{F}_q^* \rtimes \mathrm{Aut}(\mathbb{F}_q)\big)$ gegeben. Die Menge $G_{\Gamma,(i,x+1)}$ ist dann das Urbild der Untergruppe[3]

$$\big\{(0,a_1,\alpha) \mid (0,a_1,\alpha) \in \Phi^{(i,x,\Gamma)}(G_{\Gamma,(i,x)})\big\} \leq \mathbb{F}_q \rtimes \big(\mathbb{F}_q^* \rtimes \mathrm{Aut}(\mathbb{F}_q)\big)$$

und somit eine Untergruppe von $G_{\Gamma,(i,x)}$.

Wir beweisen, dass die Funktion $\Phi^{(i,x,\Gamma)} = \left(\Phi_0^{(i,x,\Gamma)}, \Phi_1^{(i,x,\Gamma)}, \Phi_2^{(i,x,\Gamma)}\right)$ einen Homomorphismus definiert, indem wir dies getrennt nach den einzelnen Komponenten $(\mathbb{F}_q,+)$, $(\mathbb{F}_q^*,\cdot)$ und $\mathrm{Aut}(\mathbb{F}_q)$ des semidirekten Produkts nachrechnen. Es gilt für beliebige $(A,\varphi;\alpha),(B,\psi;\beta) \in G_{\Gamma,(i,x)}$:

$$\begin{aligned}&\Phi_0^{(i,x,\Gamma)}\Big((A,\varphi;\alpha)(B,\psi;\beta)\Big) = \overline{\mathrm{coeff}}^{(x)}\Big(((A,\varphi;\alpha)(B,\psi;\beta)\Gamma - \Gamma)_{i,n-1}\Big)\\
&= \overline{\mathrm{coeff}}^{(x)}\left(\left(A_{i,i}\alpha\Big(((B,\psi;\beta)\Gamma)_{i,n-1}\Big) + \sum_{j=i+1}^{k-1} A_{i,j}\alpha\Big(((B,\psi;\beta)\Gamma)_{j,n-1}\Big)\right)\varphi_{n-1}^{-1} - \Gamma_{i,n-1}\right)\\
&= \overline{\mathrm{coeff}}^{(x)}\left(\left(A_{i,i}\alpha\Big(\Phi_0^{(i,x,\Gamma)}(B,\psi;\beta)\theta^x + \Gamma_{i,n-1}\Big) + \sum_{j=i+1}^{k-1} A_{i,j}\alpha(\Gamma_{j,n-1})\right)\varphi_{n-1}^{-1} - \Gamma_{i,n-1}\right)\\
&= \overline{\mathrm{coeff}}^{(x)}\Big(A_{i,i}\alpha\Big(\Phi_0^{(i,x,\Gamma)}(B,\psi;\beta)\theta^x\Big)\varphi_{n-1}^{-1}\Big) + \overline{\mathrm{coeff}}^{(x)}\Big(((A,\varphi;\alpha)\Gamma)_{i,n-1} - \Gamma_{i,n-1}\Big)\\
&= \overline{A_{i,i}}\cdot\tau^{ex}(\overline{\varphi_{n-1}}^{-1})\cdot\overline{\mathrm{coeff}}^{(x)}(\alpha(\theta^x))\cdot\overline{\alpha}\Big(\Phi_0^{(i,x,\Gamma)}(B,\psi;\beta)\Big) + \Phi_0^{(i,x,\Gamma)}(A,\varphi;\alpha)\\
&= \Phi_1^{(i,x,\Gamma)}(A,\varphi;\alpha)\cdot\Phi_2^{(i,x,\Gamma)}(A,\varphi;\alpha)\Big(\Phi_0^{(i,x,\Gamma)}(B,\psi;\beta)\Big) + \Phi_0^{(i,x,\Gamma)}(A,\varphi;\alpha)\end{aligned}$$

[3] Über die Komponente $(\mathbb{F}_q,+)$ messen wir die Veränderung an dem Eintrag $\Gamma_{i,n-1}$.

und

$$\begin{aligned}
&\Phi_1^{(i,x,\Gamma)}\Big((A,\varphi;\alpha)(B,\psi;\beta)\Big) = \Phi_1^{(i,x,\Gamma)}\Big((A\alpha(B),\varphi\alpha(\psi);\alpha\circ\beta)\Big)\\
&= \overline{\mathrm{coeff}}^{(x)}\Big(\alpha\circ\beta(\theta^x)\Big)\cdot\overline{A_{i,i}\alpha(B_{i,i})}\cdot\tau^{ex}\Big(\overline{\varphi_{n-1}\alpha(\psi_{n-1})}\Big)^{-1}\\
&= \overline{\mathrm{coeff}}^{(x)}\Big(\alpha\Big(\mathrm{coeff}^{(x)}(\beta(\theta^x))\theta^x\Big)\Big)\cdot\overline{A_{i,i}}\cdot\overline{\alpha(B_{i,i})}\cdot\tau^{ex}(\overline{\varphi_{n-1}})^{-1}\cdot\tau^{ex}\Big(\overline{\alpha(\psi_{n-1})}\Big)^{-1}\\
&= \overline{\alpha}\Big(\overline{\mathrm{coeff}}^{(x)}(\beta(\theta^x))\Big)\cdot\overline{\mathrm{coeff}}^{(x)}(\alpha(\theta^x))\cdot\overline{A_{i,i}}\cdot\overline{\alpha}(\overline{B_{i,i}})\cdot\tau^{ex}(\overline{\varphi_{n-1}})^{-1}\cdot\overline{\alpha}\Big(\tau^{ex}\big(\overline{\psi_{n-1}}\big)^{-1}\Big)\\
&= \Big(\overline{\mathrm{coeff}}^{(x)}(\alpha(\theta^x))\cdot\overline{A_{i,i}}\cdot\tau^{ex}(\overline{\varphi_{n-1}})^{-1}\Big)\cdot\overline{\alpha}\Big(\overline{\mathrm{coeff}}^{(x)}(\beta(\theta^x))\cdot\overline{B_{i,i}}\cdot\tau^{ex}\big(\overline{\psi_{n-1}}\big)^{-1}\Big)\\
&= \Phi_1^{(i,x,\Gamma)}(A,\varphi;\alpha)\cdot\overline{\alpha}\Big(\Phi_1^{(i,x,\Gamma)}(B,\psi;\beta)\Big)\\
&= \Phi_1^{(i,x,\Gamma)}(A,\varphi;\alpha)\cdot\Phi_2^{(i,x,\Gamma)}(A,\varphi;\alpha)\Big(\Phi_1^{(i,x,\Gamma)}(B,\psi;\beta)\Big)
\end{aligned}$$

sowie

$$\begin{aligned}
&\Phi_2^{(i,x,\Gamma)}\Big((A,\varphi;\alpha)(B,\psi;\beta)\Big) = \Phi_2^{(i,x,\Gamma)}\Big((A\alpha(B),\varphi\alpha(\psi);\alpha\circ\beta)\Big)\\
&= \overline{\alpha\circ\beta} = \overline{\alpha}\circ\overline{\beta} = \Phi_2^{(i,x,\Gamma)}(A,\varphi;\alpha)\circ\Phi_2^{(i,x,\Gamma)}(B,\psi;\beta)
\end{aligned}$$

□

Es sei nun wie im Hilfssatz $\Gamma\in R^{k\times n,\lambda,\mu}$ eine Generatormatrix, so dass $\mathrm{CF}_G(\Gamma_{*,[n-1]}) = \Gamma_{*,[n-1]}$ in reduzierter Zeilenstufenform ist. Die zum Beweis der Aussage für $i\in[k]$ und $x\in[m]$ eingeführte Abbildung $\Phi^{(i,x,\Gamma)}$ ermöglicht es uns nun, Algorithmen zur Berechnung von $\mathrm{Can}_G(\Gamma)$ anzugeben. Wir gehen dabei wie folgt vor:

- Der Algorithmus 5.1 beschreibt zunächst das Vorgehen, um den Koeffizienten $\overline{\mathrm{coeff}}^{(x)}(\Gamma_{i,n-1})$ des Spalteneintrags $\Gamma_{i,n-1}$ unter der Gruppenoperation von $G_{\Gamma,(i,x)}$ zu minimieren. Hier nutzen wir den Gruppenhomomorphismus $\Phi^{(i,x,\Gamma)}$ aus.

- In den Algorithmen 5.2 und 5.3 werden dann die beiden Situationen $\mathrm{shp}(\Gamma) = \mathrm{shp}(\Gamma_{*,[n-1]})$ bzw. $\mathrm{shp}(\Gamma)\neq\mathrm{shp}(\Gamma_{*,[n-1]})$ getrennt voneinander untersucht.

In den Algorithmen werden wir auch auf folgende, leicht zu verifizierende Eigenschaft der Gruppen $G_{\Gamma,(i,x)}$ zurückgreifen:

5.1.44 Hilfssatz. *Für beliebiges $i\in[k]$, $x\in[m]$ und $(A,\varphi;\alpha)\in G_{\Gamma,(i,x)}$ ist*

$$G_{(A,\varphi;\alpha)\Gamma,(i,x+1)} = (A,\varphi;\alpha)G_{\Gamma,(i,x+1)}(A,\varphi;\alpha)^{-1}$$

Beweis. Dies lässt sich leicht aus der Definition der Gruppen als Stabilisatoren herleiten. □

Algorithmus 5.1 MinStep

Input: $i \in [k-1]$, $x \in [m]$
Input: $\Gamma \in R^{k\times n,\lambda,\mu}$ so dass $\mathrm{CF}_G(\Gamma_{*,[n-1]}) = \Gamma_{*,[n-1]}$ in reduzierter Zeilenstufenform
Input: $E := (E_j)_{j\in[s+1]}$ Erzeugendensystem der Gruppe $\langle E\rangle = G_{\Gamma,(i,x)}$ mit

- $\langle E_{[j]}\rangle \trianglelefteq \langle E_{[j+1]}\rangle$ für alle $j \in [s+1]$ mit $\langle E_{[j+1]}\rangle/\langle E_{[j]}\rangle \simeq \mathbb{Z}_{e_j}$
- für alle $j \in [s]$ gilt $\Phi_2^{(i,x,\Gamma)}(E_j) = \mathrm{id}_{\mathbb{F}_q}$ und $e_j \neq p \iff e_j \mid (q-1)$
- $e_s = \mathrm{ord}(\Phi_2^{(i,x,\Gamma)}(E_s))$

Output: $\Gamma' \in G_{\Gamma,(i,x)}\Gamma$ mit $\overline{\mathrm{coeff}}^{(x)}\big(\Gamma'_{i,n-1}\big)$ minimal unter allen Bahnenelementen
Output: $T \in G_{\Gamma,(i,x)} : T\Gamma = \Gamma'$ und E' Erzeugendensystem von $G_{\Gamma',(i,x+1)}$

procedure MinStep(i, x, Γ, E)
  $E' \leftarrow ()$
  $(a_j, b_j, \alpha_j) \leftarrow \Phi^{(i,x,\Gamma)}(E_j)$ für alle $j \in [s+1]$
  $V \leftarrow \{0_{\mathbb{F}_q}\}$ // als $\mathbb{F}_p$-Untervektorraum von $(\mathbb{F}_q, +)$
  $B \leftarrow \emptyset$ // Indizes zu einer Basis von V
  $W \leftarrow \{0_{\mathbb{F}_q}\}$, $B' \leftarrow \emptyset$
  for $j \in [s+1]$ **do**
    if $j \neq s \wedge b_j = 1_{\mathbb{F}_q}$ **then**
      if $a_j \in V$ **then**
        bestimme $x \in [p]^B$, so dass $a_j + \sum_{\ell\in B} x_\ell a_\ell = 0$
        Append$(E', \big(\prod_{\ell\in B} E_\ell^{x_\ell}\big)E_j)$
      else
        $V \leftarrow \langle V, a_j\rangle$ // als $\mathbb{F}_p$-Untervektorraum von $\mathbb{F}_q$
        Append(B, j)
        Append$(E', (I_k, \mathbf{1_n}; \mathrm{id}_R))$
    else
      $(\widetilde{a}, \widetilde{b}, \widetilde{\alpha}) \leftarrow (a_j, b_j, \alpha_j)$, $z \leftarrow 1$
      $W' \leftarrow W$
      while $\nexists\, a' \in W' : \widetilde{a} + \widetilde{b}\cdot\widetilde{\alpha}(a') \in V$ **do**
        $W \leftarrow W \cup \{\widetilde{a} + \widetilde{b}\cdot\widetilde{\alpha}(a') \mid a' \in W'\}$
        $z \leftarrow z + 1$
        $(\widetilde{a}, \widetilde{b}, \widetilde{\alpha}) \leftarrow (\widetilde{a}, \widetilde{b}, \widetilde{\alpha})\cdot(a_j, b_j, \alpha_j)$ // $= (a_j, b_j, \alpha_j)^z$
      bestimme $x \in [p]^B$ und $y_{\ell'} \in [e_{\ell'}]$ für $\ell' \in B'$, so dass

$$\Phi_0^{(i,x,\Gamma)}\left(E_j^z \prod_{\ell'\in B'} E_{\ell'}^{y_{\ell'}}\right) + \sum_{\ell\in B} x_\ell a_\ell = 0$$

      Append$(E', \big(\prod_{\ell\in B} E_\ell^{x_\ell}\big)E_j^z \prod_{\ell'\in B'} E_{\ell'}^{y_{\ell'}})$
      $B' \leftarrow B' \cup \{j\}$
  // Ende for

Algorithmus 5.1 MinStep (Fortsetzung)

27: $(v, w) \leftarrow \underset{(v',w')\in V\times W}{\operatorname{argmin}} \left(\overline{\text{coeff}}^{(x)}(\Gamma_{i,n-1}) + w' + v'\right)$
28: bestimme $x \in [p]^B : \sum_{\ell\in B} x_\ell a_\ell = v$
29: bestimme $y_{\ell'} \in [e_{\ell'}]$ für $\ell' \in B'$ mit $\Phi_0^{(i,x,\Gamma)}\left(\prod_{\ell'\in B'} E_{\ell'}^{y_{\ell'}}\right) = w$
30: $T \leftarrow \left(\prod_{\ell\in B} E_\ell^{x_\ell}\right) \cdot \left(\prod_{\ell'\in B'} E_{\ell'}^{y_{\ell'}}\right)$
31: $E' \leftarrow \left(TE'_jT^{-1}\right)_{j\in[s+1]}$ // siehe Bemerkung 5.1.46
32: **return** $(T\Gamma, T, E')$

5.1.45 Hilfssatz. *Algorithmus 5.1 ist korrekt.*

Beweis. Um auf verschiedene Variablenbelegungen während des Ablaufs des Algorithmus zugreifen zu können, bezeichne die mit $[j+1]$ indizierten Variablen $V^{[j+1]}$, $W^{[j+1]}$, $B^{[j+1]}$ den Wert der Variablen V, W, B nach Abschluss von Zeile 25 in Algorithmus 5.1 zum Index $j \in [s+1]$. Außerdem sei die Variablenbelegung direkt nach der Initialisierung mit $[0]$ indiziert.

Für den Beweis zeigen wir zunächst induktiv, dass für alle $j \in [s+1]$ die folgende Hilfsaussage gilt:

$$V^{[j]} = \left\langle a_\ell \mid \ell \in B^{[j]} \right\rangle_{\mathbb{F}_p} = \Phi_0^{(i,x,\Gamma)}\left(\langle E_{[j]}\rangle \cap \ker\left(\Phi_1^{(i,x,\Gamma)} \rtimes \Phi_2^{(i,x,\Gamma)}\right)\right) \tag{5.3}$$

$$\bigcup_{w\in W^{[j]}} \left(w + V^{[j]}\right) = \Phi_0^{(i,x,\Gamma)}\left(\langle E_{[j]}\rangle\right) \tag{5.4}$$

$$\langle E'_{[j]}\rangle = {\Phi_0^{(i,x,\Gamma)}}^{-1}(0) \cap \langle E_{[j]}\rangle \tag{5.5}$$

Der Induktionsstart $j = 0$ entspricht den initialisierten Variablen. Diese erfüllen ganz offensichtlich die gemachten Aussagen. Für den Induktionsschritt von j nach $j+1$ unterscheiden wir nun die beiden Fälle analog zu der If-Bedingung in Zeile 8:

1. Fall ($j \neq s \wedge b_j = 1_{\mathbb{F}_q}$): Jedes beliebige Gruppenelement $(A, \varphi; \alpha) \in \langle E_{[j+1]}\rangle$ lässt sich in der Form $(A, \varphi; \alpha) = E_j^z(B, \psi; \beta)$ mit $z \in \mathbb{N}$ und $(B, \psi; \beta) \in \langle E_{[j]}\rangle$ darstellen.

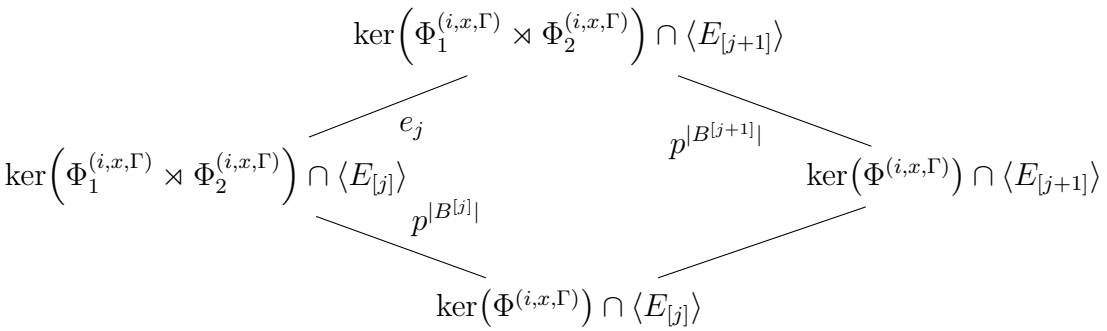
Um Gleichung (5.3) zu beweisen, sei nun $(A, \varphi; \alpha) \in \langle E_{[j]}\rangle \cap \ker\left(\Phi_1^{(i,x,\Gamma)} \rtimes \Phi_2^{(i,x,\Gamma)}\right)$ beliebig. Da auch $E_j \in \ker\left(\Phi_1^{(i,x,\Gamma)} \rtimes \Phi_2^{(i,x,\Gamma)}\right)$ im Kern liegt, ist $(B, \psi; \beta)$ ebenfalls ein Element dieser Untergruppe, und es gilt

$$\Phi_0^{(i,x,\Gamma)}(A, \varphi; \alpha) = \Phi_0^{(i,x,\Gamma)}(E_j^z(B, \psi; \beta)) \in za_j + V^{[j]} \subseteq V^{[j+1]}.$$

Umgekehrt lässt sich aber auch aufgrund der Induktionsvoraussetzung jedes Element $v \in V^{[j+1]}$ als Summe $v = \sum_{b\in B^{[j+1]}} z_b a_b$ mit $z_b \in [p]$ darstellen. Das Gruppenelement $\prod_{b\in B^{[j+1]}} E_b^{z_b}$ liegt im Kern von $\Phi_1^{(i,x,\Gamma)} \rtimes \Phi_2^{(i,x,\Gamma)}$ und es ist $\Phi_1^{(i,x,\Gamma)}(\prod_{b\in B^{[j+1]}} E_b^{z_b}) = v$.

Damit ist die Aussage der Gleichung (5.3) in beiden Teilfällen $a_j \in V^{[j]}$ bzw. $a_j \notin V^{[j]}$ gezeigt. Gleichung (5.4) zeigt man analog.

Die Gültigkeit von Gleichung (5.5) ergibt sich aus dem Untergruppendiagramm



Die angegebenen Indizes der jeweiligen Untergruppen lassen sich leicht über den Homomorphiesatz, also über die Mächtigkeiten der Bilder schließen. Ist nun $a_j \in V^{[j]}$, so ist $|B^{[j+1]}| = |B^{[j]}|$ und daher der Index von $\ker\left(\Phi^{(i,x,\Gamma)}\right) \cap \langle E_{[j]}\rangle$ in $\ker\left(\Phi^{(i,x,\Gamma)}\right) \cap \langle E_{[j+1]}\rangle$ gleich e_j. Man überlegt sich leicht, dass der neu hinzugenommene Erzeuger E'_j in Zeile 11 genau den nötigen Anstieg der Gruppenordnung bewirkt.

Im Fall von $a_j \notin V^{[j]}$ ist aber p ein Teiler von e_j und damit $e_j = p$. Damit sind aber die Gruppen $\ker\left(\Phi^{(i,x,\Gamma)}\right) \cap \langle E_{[j]}\rangle$ und $\ker\left(\Phi^{(i,x,\Gamma)}\right) \cap \langle E_{[j+1]}\rangle$ in diesem Fall gleich.

2.Fall ($j = s \vee b_j \neq 1_{\mathbb{F}_q}$): Wir beginnen wieder mit dem Beweis der Aussage (5.3). Falls $j < s$ gilt, so ist e_j ein Teiler von $(q-1)$. Also sind auch die Quotienten

$$\frac{\left|\ker\left(\Phi_1^{(i,x,\Gamma)} \rtimes \Phi_2^{(i,x,\Gamma)}\right) \cap \langle E_{[j+1]}\rangle\right|}{\left|\ker\left(\Phi_1^{(i,x,\Gamma)} \rtimes \Phi_2^{(i,x,\Gamma)}\right) \cap \langle E_{[j]}\rangle\right|} \quad \text{bzw.} \quad \frac{\left|\Phi_0^{(i,x,\Gamma)}\left(\langle E_{[j+1]}\rangle \cap \ker\left(\Phi_1^{(i,x,\Gamma)} \rtimes \Phi_2^{(i,x,\Gamma)}\right)\right)\right|}{\left|\Phi_0^{(i,x,\Gamma)}\left(\langle E_{[j]}\rangle \cap \ker\left(\Phi_1^{(i,x,\Gamma)} \rtimes \Phi_2^{(i,x,\Gamma)}\right)\right)\right|}$$

Teiler von e_j bzw. $(q-1)$. Hieraus schließen wir, dass

$$\begin{aligned} V^{[j+1]} &= \Phi_0^{(i,x,\Gamma)}\left(\langle E_{[j+1]}\rangle \cap \ker\left(\Phi_1^{(i,x,\Gamma)} \rtimes \Phi_2^{(i,x,\Gamma)}\right)\right) \\ &= \Phi_0^{(i,x,\Gamma)}\left(\langle E_{[j]}\rangle \cap \ker\left(\Phi_1^{(i,x,\Gamma)} \rtimes \Phi_2^{(i,x,\Gamma)}\right)\right) = V^{[j]} \end{aligned}$$

gelten muss. Im Spezialfall $j = s$ können wir nicht über die Teilbarkeit argumentieren, da möglicherweise der Index e_s ein Vielfaches von p ist. Wir nutzen in dieser Situation die Tatsache, dass für beliebiges $(B, \psi; \beta) \in \langle E_{[s+1]}\rangle$ eine Gleichung der Gestalt:

$$\mathrm{id}_{\mathbb{F}_q} = \Phi_2^{(i,x,\Gamma)}\left(E_s^{z'}(B, \psi; \beta)\right) = \alpha_s^{z'} \circ \mathrm{id}_{\mathbb{F}_q} \text{ für ein } z' \in \mathbb{Z}$$

sofort $\alpha_s^{z'} = \mathrm{id}_{\mathbb{F}_q}$ und somit $z' \equiv 0 \mod e_s$ impliziert. Hieraus ergibt sich $V^{[s+1]} = V^{[s]}$, denn es ist

$$\ker\left(\Phi_1^{(i,x,\Gamma)} \rtimes \Phi_2^{(i,x,\Gamma)}\right) \cap \langle E_{[s+1]}\rangle = \ker\left(\Phi_1^{(i,x,\Gamma)} \rtimes \Phi_2^{(i,x,\Gamma)}\right) \cap \langle E_{[s]}\rangle.$$

In beiden Situationen bleibt also die Belegung der Variablen $V^{[j]} = V^{[j+1]}$ begründetermaßen im Schleifendurchlauf unverändert. Somit ist Aussage (5.3) bewiesen.

Innerhalb der While-Schleife, welche in Zeile 19 beginnt, gibt uns die Variable z jeweils den Exponenten für die Gleichung $(\widetilde{a}, \widetilde{b}, \widetilde{\alpha}) = (a_j, b_j, \alpha_j)^z$. Die Schleife terminiert, da $E_j^{e_j}$ in $\langle E_{[j]} \rangle$ liegt und daher $\Phi_0^{(i,x,\Gamma)}(E_j^{e_j}) \in W^{[j]} + V^{[j]}$ das Abbruchkriterium erfüllt.

Der in Zeile 24 hinzugenommene Erzeuger E'_j erfüllt ganz offensichtlich $\Phi_0^{(i,x,\Gamma)}(E'_j) = 0$. Dabei ergibt sich die Existenz einer zulässigen Variablenbelegung $x \in [p]^B$ und $y_{\ell'}$, $\ell' \in B'$ aus dem Abbruchkriterium der While-Schleife.

Für den Beweis der Hilfsaussagen (5.4) und (5.5) zeigen wir zunächst, dass der Vektorraum $V^{[j]}$ unter der Operation $v \mapsto b_j \alpha_j(v)$ abgeschlossen ist. Hierfür nutzt man die Normalteilereigenschaft $\langle E_{[j]} \rangle \trianglelefteq \langle E_{[j+1]} \rangle$ aus:
Es sei $v = \sum_{\ell \in B^{[j]}} x_\ell a_\ell$ beliebig, dann gilt $E_j \big(\prod_{\ell \in B^{[j]}} E_\ell^{x_\ell} \big) E_j^{-1} \in \langle E_{[j]} \rangle$ und somit

$$(a_j, b_j, \alpha_j)(v, 1, \mathrm{id}) \underbrace{(a_j, b_j, \alpha_j)^{-1}}_{=\left(-\alpha_j^{-1}\left(b_j^{-1} a_j\right), \alpha_j^{-1}\left(b_j^{-1}\right), \alpha_j^{-1}\right)} = (b_j \alpha_j(v), 1, \mathrm{id}) \in V^{[j]}.$$

Außerdem lässt sich jedes beliebige $(A, \varphi; \alpha) \in \langle E_{[j+1]} \rangle$ in der Form

$$(A, \varphi; \alpha) = E_j^{z' + z \cdot z''}(B, \psi; \beta) = E_j^{z'}(B', \psi'; \beta') {E'_j}^{z''}$$

mit $z' \in [z]$, $z'' \in \mathbb{N}$ und $(B, \psi; \beta), (B', \psi'; \beta') \in \langle E_{[j]} \rangle$ darstellen. Damit ergibt sich

$$\begin{aligned} \Phi^{(i,x,\Gamma)}(A, \varphi; \alpha) &= (a_j, b_j, \alpha_j)^{z'} \cdot \underbrace{\Phi^{(i,x,\Gamma)}(B', \psi'; \beta')}_{=:(a', b', \alpha')} \cdot \underbrace{\left(\Phi^{(i,x,\Gamma)}(E'_j)\right)^{z''}}_{=:(0, b'', \alpha'')} \\ &= (a_j, b_j, \alpha_j)^{z'} (\underbrace{a'}_{\in V^{[j]} + W^{[j]}}, b', \alpha')(0, b'', \alpha'')^{z''} \end{aligned}$$

und somit $\Phi_0^{(i,x,\Gamma)}(A, \varphi; \alpha) \in W^{[j+1]} + V^{[j]} = W^{[j+1]} + V^{[j+1]}$. Dies beweist sofort Aussage (5.4).

Die dritte Aussage (5.5) zeigt man ähnlich: Wir nehmen an, dass

$$\langle E'_{[j+1]} \rangle \subsetneq {\Phi_0^{(i,x,\Gamma)}}^{-1}(0) \cap \langle E_{[j+1]} \rangle$$

eine echte Teilmenge sei und wollen dies zu einem Widerspruch führen. Dann existiert eine ganze Zahl $0 < z' < z$ und ein Gruppenelement $(B, \psi; \beta) \in \langle E_{[j]} \rangle$, so dass $\Phi^{(i,x,\Gamma)}(E_j^{z'}(B, \psi; \beta)) = (0, b, \alpha)$ für ein $b \in \mathbb{F}_q^*$, $\alpha \in \mathrm{Aut}(\mathbb{F}_q)$ ist. Die Gleichung

$$\begin{aligned} (0, b, \alpha) = \Phi^{(i,x,\Gamma)}\Big(E_j^{z'}(B, \psi; \beta)\Big) &= (a_j, b_j, \alpha_j)^{z'} \cdot \underbrace{\Phi^{(i,x,\Gamma)}(B, \psi; \beta)}_{(a', b', \alpha')} \\ &= (a_j, b_j, \alpha_j)^{z'} \cdot (\underbrace{a'}_{\in W^{[j]} + V^{[j]}}, b', \alpha') \end{aligned}$$

widerspräche sofort der Minimalität des Exponenten z aus dem Abbruchkriterium der While-Schleife in Zeile 19.

Mit der Aussage (5.4) für $j = s$ zeigt man nun leicht, dass das berechnete Gruppenelement T aus Zeile 30 genau zu einer Generatormatrix $T\Gamma =: \Gamma'$ führt, für welche die Projektion auf den Eintrag $\overline{\text{coeff}}^{(x)}(\Gamma'_{i,n-1})$ minimal unter allen Bahnenelementen ist. Genauso garantiert (5.5), dass ein Erzeugendensystem für die $G_{\Gamma',(i,x+1)}$ bestimmt wurde. □

5.1.46 Bemerkung. Ist der neue Erzeuger E'_j aus dem Kern von $\Phi^{(i,x,\Gamma)}$, so ist es nicht nötig die Konjugation mit T in Zeile 31 von Algorithmus 5.1 durchzuführen. Dies beweisen wir über die folgende Rechnung:
Es sei $E, T \in G_{\Gamma,(i,x)}$ beliebig. Weiter sei $\Phi^{(i,x,\Gamma)}(E) =: (a, b, \alpha)$ und $\Phi_0^{(i,x,\Gamma)}(T) = t$. Dann gilt:

$$\begin{aligned}\Phi_0^{(i,x,T\Gamma)}(E) &= \overline{\text{coeff}}^{(x)}((ET\Gamma)_{i,n-1} - (T\Gamma)_{i,n-1}) \\ &= \overline{\text{coeff}}^{(x)}((ET\Gamma)_{i,n-1} - \Gamma_{i,n-1}) - \overline{\text{coeff}}^{(x)}((T\Gamma)_{i,n-1} - \Gamma_{i,n-1}) \\ &= \Phi_0^{(i,x,\Gamma)}(ET) - \Phi_0^{(i,x,\Gamma)}(T) = a + b \cdot \alpha(t) - t\end{aligned}$$

Damit ergibt sich $\Phi_0^{(i,x,T\Gamma)}(E'_j) = 0_{\mathbb{F}_q}$.

5.1.47 Bemerkung. Wir wollen an dieser Stelle noch einige weitere Hinweise zu der Implementierung von Algorithmus 5.1 geben:

- Die Generatoren in Zeile 15 wurden nur für den Beweis hinzugefügt. Ebenso sollte in Zeile 24 nur dann der Erzeuger E'_j hinzugefügt werden, falls $z < e_j$ gilt. Tritt dieser Fall für $j = s$ ein, so ist im Folgenden natürlich dies mit der Aufhebung der Forderung $e_s = \text{ord}(\Phi_2^{(i,x,\Gamma)}(E_s))$ zu berücksichtigen.

- Die Menge W sollte in einer Implementierung außer den Nebenklassenrepräsentanten auch deren Berechnungsweg speichern. Dies ermöglicht eine leichte Bestimmung der Variablenbelegungen $y_{\ell'}$ für $\ell' \in B'$.

- Der Test $a_j \in V$ lässt sich leicht über Methoden der linearen Algebra über $\mathbb{F}_p$ vornehmen, denn V ist ein $\mathbb{F}_p$-Vektorraum. Man transformiert etwa die Matrix $(\overline{\text{coeff}}^{(*,x)}(a_\ell))_{\ell \in B} \in \mathbb{F}_p^{r \times |B|}$ auf Zeilenstufenform. Ebenso bestimmt man die Variablenbelegungen $x \in [p]^B$ in den Zeilen 23 und 28 über einfache Rechnungen in $\mathbb{F}_p^r$.

- Die Bestimmung des Minimums in Zeile 27 erfolgt ebenfalls über einfache Vektoroperationen in $\mathbb{F}_p^r$. Hier erweist sich auch die gewählte Definition 4.1.15 der Totalordnung auf R als sehr nützlich.

- Werden die Erzeugendensysteme E über die Algorithmen 5.2 und 5.3 berechnet, so werden wir später sehen, dass es höchstens drei Indizes $j \in [s]$ gibt mit $\Phi_1^{(i,x,\Gamma)}(E_j) \neq 1_{\mathbb{F}_q}$.
- Der Algorithmus bleibt weiterhin gültig, falls wir bereits im Vorfeld Erzeuger E_j mit $E_j \in \ker\left(\Phi^{(i,x,\Gamma)}\right)$ aus der Liste E entfernen und nach Abschluss wieder an der entsprechenden Stelle einfügen. Dieses Vorgehen werden wir in den Algorithmen 5.2 und 5.3 anwenden.

Der Algorithmus 5.1 bildet nun die Basis um umrisstreue Generatormatrizen $\Gamma \in R^{k\times n,\lambda,\mu}$ unter der Operation von G zu kanonisieren. Wir gehen dabei spaltenweise vor und bezeichnen mit $\Gamma^{(0)} = \Gamma$ die Ausgangsmatrix und mit $\Gamma^{(i+1)} := \mathrm{TR}_G\left(\Gamma^{(i)}_{[i+1]}\right) \cdot \Gamma$ das Zwischenergebnis nach der Kanonisierung der Spalte $\Gamma^{(i)}_i$ für alle $i \in [n]$.

Bei einem solchen Schritt von $\Gamma^{(i)}$ zu $\Gamma^{(i+1)}$ wollen wir mit k' den Rang der Teilmatrix $\Gamma^{(i)}_{*,[i]}$ bezeichnen. Weiter sei $\widetilde{\lambda}$ der Umriss der Teilmatrix $\Gamma^{(i+1)}_{*,[i+1]}$ und $\widetilde{\mu} = \mu_{[i+1]}$. Induktiv ergibt sich dann, dass für die ersten $i+1$ Spalten der Matrix $\Gamma^{(i)}$ stets gilt:

$$\Gamma^{(i)}_{*,[i+1]} = \begin{pmatrix} \Gamma^{(i)}_{[k'],[i]} & \gamma^{(0)} \\ \mathbf{0}_{(\mathbf{k-k'})\times\mathbf{i}} & \gamma^{(1)} \end{pmatrix} \text{ mit } \gamma^{(0)} := \Gamma^{(i)}_{[k'],i} \text{ und } \gamma^{(1)} := \Gamma^{(i)}_{[k]\setminus[k'],i}$$

und einer Generatormatrix $\mathrm{CF}_G\left(\Gamma^{(i)}_{[k'],[i]}\right) = \Gamma^{(i)}_{[k'],[i]}$ in reduzierter Zeilenstufenform.

Ist $\gamma^{(1)} = \mathbf{0}_{(\mathbf{k-k'})\times\mathbf{1}}$, so können wir uns zur Lösung des Problems auf die ersten k' Zeilen zurückziehen und diese Situation als Kanonisierung von $\Gamma^{(i)}_{[k'],[i+1]} = \begin{pmatrix} \Gamma^{(i)}_{[k'],[i]} & \gamma^{(0)} \end{pmatrix}$ unter der Operation von $G^{(\widetilde{\lambda},\widetilde{\mu})}$ auffassen. Diese Situation lösen wir dann mit dem Algorithmus 5.2.

Ist $\gamma^{(1)} \neq \mathbf{0}_{(\mathbf{k-k'})\times\mathbf{1}}$, so können wir, wie im Beweis zum Hilfssatz 5.1.17, die Matrix mit elementaren Zeilenoperationen auf die Gestalt

$$\Gamma^{(i)}_{*,[i+1]} = \begin{pmatrix} \Gamma^{(i)}_{[k'],[i]} & \gamma^{(0)} \\ \mathbf{0_i} & \theta^{m-\lambda_{k'}} \\ \mathbf{0}_{(\mathbf{k-k'-1})\times\mathbf{i}} & \mathbf{0}_{(\mathbf{k-k'-1})\times\mathbf{1}} \end{pmatrix}$$

bringen. Hier können wir uns also auf die ersten $k'+1$ Zeilen zurückziehen und die Kanonisierung der Teilmatrix $\begin{pmatrix} \Gamma^{(i)}_{[k'],[i]} & \gamma^{(0)} \\ \mathbf{0_i} & \theta^{m-\lambda_{k'}} \end{pmatrix}$ unter der Operation von $G^{(\widetilde{\lambda},\widetilde{\mu})}$ mit Algorithmus 5.3 lösen. An dieser Stelle wollen wir noch bemerken, dass die weiteren verfügbaren elementaren Zeilenoperationen dazu führen, dass das Resultat $\mathrm{CF}_{G^{(\widetilde{\lambda},\widetilde{\mu})}}\left(\Gamma^{(i)}_{*,[i+1]}\right)$ eine Matrix in reduzierter Zeilenstufenform ist.

Wir werden daher, die Algorithmen 5.2 und 5.3 nur mit Blick auf die ersten k' bzw. $k'+1$ Zeilen und $i+1$ Spalten formulieren. Die Algorithmen selbst ergeben sich im

Wesentlichen durch die induktive Anwendung des Algorithmus 5.1. Wir wollen mit der Angabe dieser Algorithmen vor allem verdeutlichen, wie man die geforderten Anforderungen an das Erzeugendensystem E garantieren kann. Außerdem dienen sie auch dazu, triviale Aufrufe von Algorithmus 5.1 zu verhindern. Unter einem trivialen Aufruf wollen wir hierbei eine der beiden folgenden Situationen verstehen:

- Alle Erzeuger aus E liegen im Kern von $\Phi^{(i,x,\Gamma)}$. Dann ist nichts zu tun.
- Oder es ist bereits zu Beginn bekannt, dass die Basis B, welche im Algorithmus 5.1 bestimmt wird, die Kardinalität r hat, und dass alle weiteren Erzeuger E_j mit $\Phi_0^{(i,x,\Gamma)}(E_j) \neq 0$ an einer späteren Stelle in der Liste E auftreten.

Im Folgenden wollen wir gegebenenfalls ein Gruppenelemente $(A, \varphi; \alpha) \in G^{(\lambda,\mu_{[n-1]})}$ stillschweigend mit $(A, (\varphi, 1); \alpha) \in G$ identifizieren (wir betten $G^{(\lambda,\mu_{[n-1]})}$ in G ein). Genauso betten wir auch $G^{(\lambda_{[k-1]},\mu_{[n-1]})}$ über $(A, \varphi; \alpha) \mapsto ((\begin{smallmatrix} A & \\ & 1 \end{smallmatrix}), (\varphi, 1); \alpha)$ in G ein.

5.1.48 Satz. *Algorithmus 5.2 ist korrekt.*

Beweis. Zunächst ist klar, dass die Untergruppe H zusammen mit der Folge F ein Erzeugendensystem von $\mathrm{Stab}_G(\Gamma_{*,[n-1]})$ bildet. Des Weiteren kommutieren die Erzeuger aus F mit den Elementen aus H.

Schließlich bemerken wir noch, dass die Erzeuger in F passend zu der Normalreihe der multiplikativen Gruppe aus dem Hilfssatz 4.1.19 gewählt wurden. Damit erfüllt das Erzeugendensystem (F, E) die Bedingungen für den Algorithmus 5.1.

Die Korrektheit des Algorithmus 5.2 ergibt sich im Wesentlichen aus einer Beobachtung, welche wir an vielen weiteren Stellen bereits ausgenutzt haben: Ist $i \in [k]$ und $P \in \mathfrak{p}^{\Gamma_{*,[n-1]}}$ mit $P \cap \{j \in \mathrm{supp}(\Gamma_{*,n-1}) \mid j \geq i\} = \emptyset$, so liegt $\Psi^{(P,\Gamma_{*,[n-1]})}(H) = \langle E^{(P)} \rangle$ im Kern von $\Phi^{(i,x,\Gamma)}$. Bemerkung 5.1.46 liefert damit die Begründung, warum die Funktion MinStep nur mit der Teilfolge $(F, E^{(Q)}, E^{(\mathrm{Aut})})$ beziehungsweise $(F, E^{(\mathrm{Aut})})$ gerufen wird.

Der Beginn der For-Schleife in Zeile 16 mit der Höhe x_0 anstatt der Höhe 0 ergibt sich aus der Tatsache, dass $G{\downarrow_{\mathrm{GL}_\lambda(R)}}$ nur obere Dreiecksmatrizen beinhaltet, d.h. wir wissen hiermit bereits, dass alle Erzeuger im Kern von $\Phi^{(i,x,\Gamma)}$ für $0 \leq x < x_0$ liegen. □

Der Algorithmus 5.3 stellt nun für den Fall $\mathrm{shp}(\Gamma) \neq \mathrm{shp}(\Gamma_{*,[n-1]})$ nur eine geeignete Abwandlung von Algorithmus 5.2 dar, in welchem wir berücksichtigen, dass die Spalteneinträge modulo dem Pivotelement reduziert werden können.

5.1.49 Satz. *Algorithmus 5.3 ist korrekt.*

Beweis. Wir geben in diesem Beweis nur die Begründungen für die Veränderungen gegenüber Algorithmus 5.2. Zunächst ist zu bemerken, dass die Folge F nach ihrer Initialisierung nur einen Bruchteil der zusätzlich notwendigen Erzeuger abdeckt. Für $i \in [k-1]$,

Algorithmus 5.2 MinimizeDependent

Input: Γ umrisstreu mit $\mathrm{CF}_G(\Gamma_{*,[n-1]}) = \Gamma_{*,[n-1]}$ und $\lambda = \mathrm{shp}(\Gamma) = \mathrm{shp}(\Gamma_{*,[n-1]})$
Input: E Erzeugendensystem von $H := \mathrm{Stab}_{G^{(\lambda,\mu_{[n-1]})}}\left(\Gamma_{*,[n-1]}\right) \leq G$ wie in Bemerkung 5.1.42
Output: $(B, \psi; \beta) \in \mathrm{Stab}_G\left(\Gamma_{*,[n-1]}\right)$ sd. $\mathrm{CF}_G(\Gamma) := (B, \psi; \beta)\Gamma$
Output: Erzeugendensystem E' von $\mathrm{Stab}_G(\Gamma)$ gemäß Bemerkung 5.1.42

1: **procedure** MinimizeDependent(Γ, E)
2: $F \leftarrow \left(\left(((I_k, (\mathbf{1_{n-1}}, 1 + \xi^{r-1-j}\theta^i); \mathrm{id}_R))_{j\in[r]}\right)_{i=\mu_{n-1}-1,\ldots,1}, (I_k, (\mathbf{1_{n-1}}, \xi; \mathrm{id}_R))\right)$
3: $P_0 \leftarrow \emptyset$
4: $(B, \psi; \beta) \leftarrow (I_k, \mathbf{1_n}; \mathrm{id}_R)$
5: **for** $i \leftarrow \max\{i \in [k] \mid \Gamma_{i,n-1} \neq 0\}$ **to** 0 **by** -1 **do**
6: **if** $i \notin P_0 \wedge \Gamma_{i,n-1} \neq 0$ **then**
7: Bestimme $Q \in \mathfrak{p}^{\Gamma_{*,[n-1]}}$ mit $i \in Q$
8: $\Gamma, T_0, (F, E^{(Q)}, E^{(\mathrm{Aut})}) \leftarrow$ MinStep$(i, \mathrm{ht}(\Gamma_{i,n-1}), \Gamma, (F, E^{(Q)}, E^{(\mathrm{Aut})}))$
9: $(B, \psi; \beta) \leftarrow T_0(B, \psi; \beta)$
10: $F \leftarrow \left(F, E^{(Q)}\right)$
11: $E \leftarrow E \setminus E^{(Q)}$
12: $P_0 \leftarrow P_0 \cup Q$
13: $x_0 \leftarrow (\mathrm{ht}(\Gamma_{i,n-1}) + 1)$
14: **else**
15: $x_0 \leftarrow \min\{\mathrm{ht}(\Gamma_{i',n-1}) \mid i' \geq i\}$
16: **for** $x \leftarrow x_0$ **to** $m - 1$ **do**
17: $\Gamma, T_0, (F, E^{(\mathrm{Aut})}) \leftarrow$ MinStep$(i, x, \Gamma, (F, E^{(\mathrm{Aut})}))$
18: $(B, \psi; \beta) \leftarrow T_0(B, \psi; \beta)$
19: **return** $((B, \psi; \beta), F \cup E)$

$j \in [r]$, $h \in [\lambda_{k-1}]$ bleiben die Erzeuger der Gestalt $(A^{(i,j,h)}, \mathbf{1_n}; \mathrm{id}_R)$ mit

$$\forall\ \ell, \ell' \in [k] : A^{(i,j,h)}_{\ell,\ell'} := \begin{cases} 1, & \ell = \ell' \\ \xi^j\theta^h, & \ell = k-1 \wedge \ell' = i, \\ 0, & \text{sonst} \end{cases}$$

welche die Addition der letzten Zeile zu einer beliebigen weiteren Zeile modellieren, zunächst außen vor. Sie gehen erst an einer späteren Stelle (ab Zeile 26) ein.

Die If-Bedingung in Zeile 10 überprüft die Höhe des gegenwärtigen Eintrags $\Gamma_{i,n-1}$. Ist sie größer gleich der Höhe des Pivotelements, so können wir den Eintrag modulo dem Pivotelement reduzieren. Es ist also nach Abschluss der Zeile 29 der Eintrag $\Gamma_{i,n-1}$ gleich Null. Wie oben rechnet man leicht nach, dass für diese modifizierte Matrix die Untergruppe $\Psi^{(Q,\Gamma_{*,[n-1]})}(H)$ für alle $x \in [m]$ im Kern der Abbildung $\Phi^{(i,x,\Gamma)}$ liegt. Somit bleiben die Blöcke P_0 und Q von $\mathfrak{p}^{\Gamma_{*,[n-1]}}$ zunächst unvereinigt.

Algorithmus 5.3 MinimizeIndependent

Input: Γ umrisstreu mit $\Gamma_{*,[n-1]} = \theta^{m-\lambda_{k-1}}$, $\mathrm{CF}_G(\Gamma_{*,[n-1]}) = \Gamma_{*,[n-1]}$ und $\lambda = \mathrm{shp}(\Gamma) \neq \mathrm{shp}(\Gamma_{*,[n-1]}) = \widetilde{\lambda}$

Input: E Erzeugendensystem von $H := \mathrm{Stab}_{G^{\left(\widetilde{\lambda},\mu_{[n-1]}\right)}}\left(\Gamma_{*,[n-1]}\right) \leq G$ wie in Bemerkung 5.1.42 beschrieben

Output: $(B,\psi;\beta) \in \mathrm{Stab}_G\left(\Gamma_{*,[n-1]}\right)$ sd. $\mathrm{CF}_G(\Gamma) := (B,\psi;\beta)\Gamma$

Output: Erzeugendensystem E' von $\mathrm{Stab}_G(\Gamma)$ gemäß Bemerkung 5.1.42

1: **procedure** MinimizeIndependent(Γ, E)
2: $F \leftarrow \left(\left(\left(\left(\begin{pmatrix} I_{k-1} & \\ & 1+\tau^{-e(m-\lambda_{k-1})}(\xi^{r-1-j})\theta^i \end{pmatrix}, (\mathbf{1_{n-1}}, 1+\xi^{r-1-j}\theta^i); \mathrm{id}_R\right)\right)_{j\in[r]}\right)_{i=\mu_{n-1}-1,\ldots,1}$
3: $\textsc{Append}\left(F, \left(\begin{pmatrix} I_{k-1} & \\ & \tau^{-e(m-\lambda_{k-1})}(\xi) \end{pmatrix}, (\mathbf{1_{n-1}}, \xi); \mathrm{id}_R\right)\right)$
4: **for** $(A,\varphi;\alpha) \in E^{(\mathrm{Aut})}$ **do**
5: $\varphi_{n-1} \leftarrow a$ mit $a \in R^* : \theta^{m-\lambda_{k-1}}a = \alpha(\theta^{m-\lambda_{k-1}})$
6: $P_0 \leftarrow \{k-1\}$
7: $(B,\psi;\beta) \leftarrow (I_k, \mathbf{1_n}; \mathrm{id}_R)$
8: **for** $i \leftarrow k-2$ **to** 0 **by** -1 **do**
9: Bestimme $Q \in \mathfrak{p}^{\Gamma_{*,[n-1]}}$ mit $i \in Q$
10: **if** $\mathrm{ht}(\Gamma_{i,n-1}) < m - \lambda_{k-1}$ **then**
11: **if** $i \notin P_0 \wedge \Gamma_{i,n-1} \neq 0$ **then**
12: $\Gamma, T_0, (F, E^{(Q)}, E^{(\mathrm{Aut})}) \leftarrow \textsc{MinStep}(i, \mathrm{ht}(\Gamma_{i,n-1}), \Gamma, (F, E^{(Q)}, E^{(\mathrm{Aut})}))$
13: $(B,\psi;\beta) \leftarrow T_0(B,\psi;\beta)$
14: $F \leftarrow \left(F, E^{(Q)}\right)$
15: $E \leftarrow E \setminus E^{(Q)}$
16: $P_0 \leftarrow P_0 \cup Q$
17: $x_0 \leftarrow (\mathrm{ht}(\Gamma_{i,n-1}) + 1)$
18: **else**
19: $x_0 \leftarrow \min\{\mathrm{ht}(\Gamma_{i',n-1}) \mid i' \geq i\}$
20: **for** $x \leftarrow x_0$ **to** $m - \lambda_{k-1} - 1$ **do**
21: $\Gamma, T_0, (F, E^{(\mathrm{Aut})}) \leftarrow \textsc{MinStep}(i, x, \Gamma, (F, E^{(\mathrm{Aut})}))$
22: $(B,\psi;\beta) \leftarrow T_0(B,\psi;\beta)$
23: $b \leftarrow \sum_{j=m-\lambda_{k-1}}^{m-1} \mathrm{coeff}^{(j)}(\Gamma_{i,n-1})\theta^{j-m+\lambda_{k-1}}$
24: $\Gamma_{i,n-1} \leftarrow \Gamma_{i,n-1} - b\theta^{m-\lambda_{k-1}}$ // Addiere das Pivotelement zu dieser Zeile
25: $B_{i,*} \leftarrow B_{i,*} - b \cdot B_{k-1,*}$
26: **if** $Q \subseteq P$ **then**
27: **for** $(A,\varphi;\alpha) \in F \cup E^{(\mathrm{Aut})}$ **do**
28: bestimme $a \in R$ mit $a\theta^{m-\lambda_{k-1}} = A_{i,*}\alpha(\Gamma_{*,k-1})\varphi_{n-1}^{-1} - \Gamma_{i,k-1}$
29: $A_{i,k-1} \leftarrow A_{i,k-1} + a \cdot A_{k-1,k-1}$
30: **return** $((B,\psi;\beta), F \cup E)$

Genauso zeigt man auch, dass die separate Behandlung der Fälle $x \geq m - \lambda_{k-1}$ in Zeile 26 korrekt ist. Nun gehen die weiteren Erzeuger $(A^{(i,j,h)}, \mathbf{1_n}; \mathrm{id}_R)_{j\in[r],h\in[\lambda_{k-1}]}$ ein. Hier tritt die von uns oben beschriebene Situation auf, dass die Bilder dieser Erzeuger eine Basis von $V = \mathbb{F}_p^r$ im Algorithmus MinStep bilden. □

Für die weiteren Beobachtungen nehmen wir an, dass der kanonische Repräsentant $\mathrm{CF}_G(\Gamma) = \Gamma$ induktiv über eine Folge der Algorithmen 5.2 und 5.3 ausgehend von einer beliebigen umrisstreuen Matrix $\Gamma' \in G\Gamma$ berechnet wurde.

5.1.50 Bemerkung. In der Zeile 8 von Algorithmus 5.2 und in Zeile 12 von Algorithmus 5.3 würde es genügen, jeweils das letzte Element von F und $E^{(Q)}$ zu übergeben, da alle weiteren Elemente nachweislich im Kern von $\Phi^{(i,x,\Gamma)}$ liegen.

Wir beenden diesen Abschnitt mit einer Abschätzung der Kardinalität des Erzeugendensystems E. Sie beeinflusst maßgeblich die Laufzeit des Backtrackalgorithmus. Wir zeigen, dass diese Zahl im Wesentlichen von r, λ und der Länge der Normalreihe (4.2) abhängt.

5.1.51 Hilfssatz. *Mit o sei die Länge der Kompositionsreihe von Aut_T gemäß (4.2) bezeichnet. Die Anzahl der Erzeuger der Gruppe $\mathrm{Stab}_G(\Gamma)$, welche in den Algorithmen 5.2 und 5.3 zurückgegeben[4] werden, ist stets durch*

$$\left|\mathfrak{p}^\Gamma\right| + r\sum_{j\in[k]}\lambda_j + o$$

nach oben beschränkt.

Beweis. Alle neuen Erzeuger, die zu Beginn von Algorithmus 5.2 in der Folge F zur Verfügung gestellt werden, werden auch wieder in den Aufrufen von MinStep entfernt. Dabei hängt der Zeitpunkt von der Verteilung der Höhen in der Spalte $\Gamma_{*,n-1}$ ab. Der Eintrag $\Gamma_{i,n-1}$ mit der Höhe $m - \mu_{n-1}$ garantiert aber, dass alle Erzeuger aus F bei dessen Kanonisierung entfernt werden.

Bei einer Vereinigung von Q mit P_0 kann man überdies leicht aus der vorausgegangenen Bemerkung und dem Ablauf von Algorithmus 5.1 schließen, dass sich dort ebenfalls die Länge des Erzeugendensystems um eins verringert.

In Algorithmus 5.3 werden dahingegen $r(\mu_{n-1} - 1) + 1$ weitere Erzeuger in der Folge F zur Verfügung gestellt. Wir unterscheiden zwei Teilfälle: Ist $\mu_{n-1} = \lambda_{k-1}$ so handelt es sich um eine Spalte, bei welcher wir alle weiteren Nichtnulleinträge über eine Addition mit dem Pivotelement zu 0 transformieren können. Damit bleiben alle neu hinzugenommenen Erzeuger erhalten. Dies deckt sich aber mit der Behauptung.

Ist hingegen $\mu_{n-1} > \lambda_{k-1}$, so gibt es mindestens einen Eintrag der Höhe $m - \mu_{n-1}$. Für diesen können wir wie oben schließen, dass alle Spaltenmultiplikationen der Gestalt

[4] Die Aussage basiert auf der Annahme, dass Algorithmus 5.1 gemäß Bemerkung 5.1.47 modifiziert wurde.

$1+a\theta^x$ für $1 \leq x \leq \mu_{n-1}-\lambda_{k-1}-1$ und $a \in R^*$ nicht durch eine Addition des Pivotelement ausgeglichen werden können. Es werden also mindestens $r(\mu_{n-1} - \lambda_{k-1} - 1)$ Erzeuger aus F in den Aufrufen der Subroutine MinStep in den Zeilen 12 und 21 auch wieder entfernt. Somit bleiben höchstens

$$r(\mu_{n-1} - 1) - r(\mu_{n-1} - \lambda_{k-1} - 1) = r\lambda_{k-1}$$

neu hinzugefügte Erzeuger auch nach Abschluss von Algorithmus 5.3 erhalten.
Über eine Induktion erhält man damit die angegebene obere Schranke. □

5.1.52 Folgerung. *Die Mächtigkeit der Gruppe* $\mathrm{Stab}_G(\Gamma)$ *ist höchstens*

$$(q-1)^{\mathfrak{p}^\Gamma} \cdot q^{\sum_{i=0}^{k-1} \lambda_i} \cdot |\mathrm{Aut}_T|.$$

5.2. Ein Kanonisierer

Nun sind alle Grundlagen gelegt, um einen Kanonisierer $\mathrm{Can}_{G \rtimes S_{\mathfrak{P}_0}}^{R^{k\times n,\lambda,\mu}}$ für die Operation der Gruppe $G \rtimes S_{\mathfrak{P}_0}$ mit

$$G := \big(\mathrm{GL}_\lambda(R)/N_\lambda(R)\big) \times \big((R^*)^n/(R^*)^\mu\big) \rtimes \mathrm{Aut}_T$$

auf der Menge aller Generatormatrizen $\Gamma \in R^{k\times n,\lambda,\mu}$ zu entwickeln. Für die zugehörigen linearen Codes können wir das Ergebnis als eine Kanonisierung unter der Operation der Gruppe aller semilinearen Isometrien von R^n interpretieren. Wie wir bereits in Abschnitt 2.3.1 beobachten konnten, ist die Gruppe aller semilinearen Isometrien die maximale Untergruppe der Isometriegruppe, welche lineare Codes auf lineare Codes abbildet. Somit ist der hier betrachtete Äquivalenzbegriff für lineare Codes, der allgemeinst Mögliche, welcher sich noch als eine Gruppenoperation auf der Menge aller linearen Codes beschreiben lässt. Die beiden schwächeren Äquivalenzbegriffe – Permutationsisometrie und lineare Isometrie – ergeben sich dann leicht durch die entsprechenden Einschränkungen auf die jeweiligen Untergruppen

$$\mathrm{GL}_\lambda(R)/N_\lambda(R) \quad \text{bzw.} \quad \big(\mathrm{GL}_\lambda(R)/N_\lambda(R)\big) \times \big((R^*)^n/(R^*)^\mu\big).$$

Mit $\Gamma \in R^{k\times n,\lambda,\mu}$ sei nun bis auf Weiteres immer diejenige Generatormatrix bezeichnet, für welche wir die Kanonisierung durchführen werden. Wir beschreiben, wie wir einen Suchbaum $\overline{T}(\Gamma, G \rtimes S_{\mathfrak{P}_0})$ gemäß Abschnitt 3.3 aufbauen werden. Die Partitionierung ist dort bereits über die Individualisierung von Koordinaten $i \in [n]$ fest vorgeschrieben worden. Die Verfeinerung V setzt sich aus einer iterierten Anwendung der inneren Kanonisierung $V^{(\mathrm{im})}$, siehe auch Abschnitt 3.3.1, und der äußeren Verfeinerung $V^{(a)}$ zusammen. Diese Funktionen sind noch nicht näher spezifiziert und müssen nun von uns bereitgestellt werden. Insbesondere werden wir hierbei auch garantieren, dass nur die Nebenklassenvertreter $\overline{\mathcal{C}}(G \rtimes S_{\mathfrak{P}_0})$ zu den speziellen Untergruppen $\overline{\mathcal{L}}(G \rtimes S_{\mathfrak{P}_0})$ aus Gleichung (3.6) als Bilder auftreten werden.

Im Rahmen der äußeren Verfeinerung $V^{(a)}$ werden wir iteriert verschiedene Familien $\left(f_{H \rtimes S_\mathfrak{P}}\right)_{H \rtimes S_\mathfrak{P} \in \overline{\mathcal{L}}(G \rtimes S_{\mathfrak{P}_0})}$ von $H \rtimes S_\mathfrak{P}$-Homomorphismen zur Anwendung bringen. Für das weitere Vorgehen ist zunächst an dieser Stelle nur entscheidend, dass diese H-invariant sind. Für $\Gamma \in R^{k \times n, \lambda, \mu}$ und $(g;\pi) \in G \rtimes S_{\mathfrak{P}_0}$ sind damit die Bilder $V^{(a)}(\Gamma, H \rtimes S_\mathfrak{P}(g;\pi))$ der äußeren Verfeinerung stets Rechtsnebenklassen von $H \rtimes S_\mathfrak{Q}$ für eine kanonische Partition $\mathfrak{Q} \preceq \mathfrak{P}$.

Wir gehen nun wie folgt vor. Im ersten Abschnitt 5.2.1 werden wir zunächst beschreiben, wie wir aus dem vorangegangenen Abschnitt 5.1.2 eine effiziente innere Kanonisierung $V^{(\mathrm{im})}$ gewinnen werden. Die Definition der Homomorphismen $f_{H \rtimes S_\mathfrak{P}}$ zur Definition der äußeren Verfeinerung folgt dann in Abschnitt 5.2.2.

Die gewählte Ordnung auf R beeinflusst die Definition des kanonischen Repräsentanten $\mathrm{Can}^{R^{k \times n, \lambda, \mu}}_{G \rtimes S_{\mathfrak{P}_0}}(\Gamma)$ maßgeblich. Insbesondere ist die Ordnung von der Wahl von θ und ξ abhängig. In Abschnitt 5.2.3 werden wir daher diskutieren, ob wir die Resultate von verschiedenen Kanonisierungen, etwa zu einem isomorphen Kettenring R' oder bei anderer Wahl von ξ und θ, miteinander vergleichen können.

5.2.1. Innere Kanonisierung

Die innere Kanonisierung wurde in Abschnitt 3.3.1 in Abhängigkeit der Funktionen $\Pi_i : X \to Y^{(i)}$, von Kanonisierern $\mathrm{Can}^{Y^{(i)}}_H$ für alle $H \leq G$ und einer Fixierreihenfolge $F : X \times \overline{\mathcal{L}}(G \rtimes S_{\mathfrak{P}_0}) \to [n]^{\leq n}$ beschrieben. Wir beginnen aus diesem Grund unsere Beschreibung zunächst mit der Definition der Abbildung Π_i für $i \in [n]$

$$\Pi_i : R^{k \times n, \lambda, \mu} \to R^k_R, \qquad \Gamma \mapsto \Gamma_{*,i}$$

als Projektion auf die i-te Spalte der Matrix. Schränken wir die Menge der Spaltenvektoren auf das Bild $Y^{(i)} := \Pi_i(R^{k \times n, \lambda, \mu})$ ein, so können wir auf diesem eine natürliche Operation von $G \rtimes \mathrm{Stab}_{S_{\mathfrak{P}_0}}(i)$ definieren: Für $(A, \varphi; \alpha, \pi) \in G \rtimes \mathrm{Stab}_{S_{\mathfrak{P}_0}}(i)$ und $\gamma \in \Pi_i(R^{k \times n, \lambda, \mu})$ setzen wir $(A, \varphi; \alpha, \pi)\gamma := A\alpha(\gamma)\varphi_i^{-1}$. Mit dieser Definition ist Π_i ganz offensichtlich ein $(G \rtimes \mathrm{Stab}_{S_{\mathfrak{P}_0}}(i))$-Homomorphismus, welcher $\mathrm{Stab}_{S_{\mathfrak{P}_0}}(i)$-invariant ist. Damit erfüllt Π_i die in Abschnitt 3.3.1 beschriebenen Anforderungen.

Als nächsten Schritt stellen wir für die Untergruppen $H \leq G$ die Kanonisierer $\mathrm{Can}^{Y^{(i)}}_H$ für die Operation von H auf $Y^{(i)}$ zur Verfügung. Dies gestaltet sich für beliebige Untergruppen $H \leq G$ und Spaltenvektoren $\gamma \in Y^{(i)}$ als schwierigeres Problem. Jedoch können wir über die Fixierreihenfolge (Definition 3.3.4) und unser Wissen, dass nur gewisse Nebenklassen $H \rtimes S_\mathfrak{P}(g;\pi) \in \overline{\mathcal{C}}(G \rtimes S_{\mathfrak{P}_0})$ auch tatsächlich im Backtracking vorkommen, die Kombinationen $(H, \gamma := \Pi_i((g;\pi)\Gamma))$ steuern.

Aus diesem Grund wollen wir uns noch einmal kurz vergegenwärtigen, in welcher Situation wir im Backtracking zu einem Knoten $(H \rtimes S_\mathfrak{P}(g;\pi), j)$ von $\overline{T}(\Gamma, G \rtimes S_{\mathfrak{P}_0})$ eine Kanonisierung $\mathrm{Can}^{Y^{(i)}}_H(\gamma)$ durchführen werden: Es sei $f = (f_0, \ldots, f_{n'-1})$ dasjenige injektive Wort der Länge $n' < n$ mit Einträgen aus $[n] \setminus \{i\}$, welches die Reihenfolge der bereits fixierten und unter G minimierten Spalten angibt. Dann ist $(g;\pi)\Gamma$ ein

Algorithmus 5.4 INNERCAN

Input: $\mathfrak{P} \preceq \mathfrak{P}_0$ kanonische Partition
Input: $f \in \mathrm{Fix}_{S_\mathfrak{P}}([n])^{'n}$ für ein $n' \in [n+1]$
Input: $\Gamma \in R^{k\times n,\lambda,\mu}$ f-semikanonisch, d.h. $\mathrm{CF}_G(\Pi_f(\Gamma)) = \Pi_f(\Gamma)$
Output: Fixierreihenfolge $F(\Gamma, G^{(f,\Gamma)} \rtimes S_\mathfrak{P})$

```
1: procedure InnerCan(𝔓, f, Γ)
2:     f̃ ← ()
3:     F ← Fix_{S_𝔓}([n]) \ {f_j | j ∈ [n']}
4:     while F ≠ ∅ do
5:         k' ← rg(Π_{f+f̃}(Γ))
6:         for e ∈ F do// in lexikographischer Reihenfolge
7:             x ← per(Γ_{[k]\[k'], e})
8:             if x ∈ {0, λ_{k'}} then
9:                 (A, φ; α) ← TR^{Y(e)}_{G^(f+f̃,Γ)}(Γ_{*,e})// falls x = 0 rufe Algorithmus 5.2, sonst Algorithmus 5.3
10:                Γ ← (A, φ; α)Γ
11:                f̃ ← f̃ + (e)
12:                F ← F \ {e}
13:                if x = λ_{k'} then
14:                    break // verlasse die For-Schleife
15:    return f + f̃
```

f-semikanonischer Repräsentant der Bahn $G\pi\Gamma$, d.h. $\Pi_f((g;\pi)\Gamma) = \mathrm{CF}_G(\Pi_f((g;\pi)\Gamma))$, und $H = G^{(f,\pi\Gamma)} = \mathrm{Stab}_G(\Pi_f((g;\pi)\Gamma))$ dessen Stabilisator. Die Koordinate $i \in [n]$ wurde fixiert und die Spalte $\gamma = \Pi_i((g;\pi)\Gamma)$ soll nun unter H minimiert werden. Ist die Matrix $\Pi_{f+(i)}((g;\pi)\Gamma)$ umrisstreu, so korrespondiert dies aber genau mit den Situationen, welche wir mit den Algorithmen 5.2 und 5.3 behandeln können.

Aus diesem Grund werden wir also nun garantieren, dass die von uns gewählte Fixierreihenfolge stets zu umrisstreuen Matrizen $\Pi_{f+(i)}((g;\pi)\Gamma)$ führt. Wir kombinieren hierzu die Definition der Fixierreihenfolge mit der inneren Kanonisierung, da uns dies die Beschreibung wesentlich vereinfacht. Wir beschreiben unser Vorgehen algorithmisch, siehe Algorithmus 5.4. Die Untergruppe $H = G^{(f,\pi\Gamma)}$ übergeben wir dem Algorithmus über die Angabe von f und $\pi\Gamma$.

5.2.1 Satz. *Der Algorithmus 5.4 definiert eine Fixierreihenfolge, indem wir parallel zum Backtracking den benötigten Funktionswert für den Knoten $((H \rtimes S_\mathfrak{P})(g;\pi), j)$ im Baum $\overline{T}(\Gamma, G \rtimes S_{\mathfrak{P}_0})$ durch*

$$F((g;\pi)\Gamma, (H \rtimes S_\mathfrak{P})) := \textsc{InnerCan}(\mathfrak{P}, \overline{F}(\Gamma, (H \rtimes S_\mathfrak{P})(g;\pi), j), (g;\pi)\Gamma)$$

bestimmen.

Beweis. Zunächst zeigen wir, dass die Funktion wohldefiniert ist. In einem ersten Schritt lösen wir dazu die Abhängigkeit von j auf. Es sei also zunächst j minimal, so dass $((H \rtimes S_{\mathfrak{P}})(g;\pi), j)$ ein Knoten im Baum $\overline{T}(\Gamma, G \rtimes S_{\mathfrak{P}_0})$ ist. Wir bezeichnen mit $f := \overline{F}(\Gamma, (H \rtimes S_{\mathfrak{P}})(g;\pi), j)$ und $\tilde{f} := \textsc{InnerCan}(\mathfrak{P}, f, (g;\pi)\Gamma)$ die getroffene Koordinatenauswahl. Ein Konflikt kann nur dann auftreten, wenn der Sohn nach der inneren Kanonisierung von der Gestalt $((H \rtimes S_{\mathfrak{P}})(g;\pi), j+1)$ ist. Dann ist aber $(g;\pi)\Gamma$ bereits $\tilde{f}$-semikanonisch und $H = G^{(\tilde{f},\pi\Gamma)}$. Diese Tatsache kann man nun nutzen um zu zeigen, dass dann der Algorithmus für die modifizierte Eingabe $\tilde{f} = f + \tilde{f}$ eine identische Ausgabe generiert

$$\tilde{f} = \textsc{InnerCan}(\mathfrak{P}, \tilde{f}, (g;\pi)\Gamma).$$

Da aber für alle weiteren Knoten $((H \rtimes S_{\mathfrak{P}})(g;\pi), j')$ mit $j' > j$ der Funktionsaufruf $\overline{F}(\Gamma, (H \rtimes S_{\mathfrak{P}})(g;\pi), j')$ gerade die Folge $\tilde{f}$ zurück gibt, ist die Unabhängigkeit von dem Iterationszähler j gezeigt.

Als nächstes beweisen wir, dass die Funktion auch von dem aktuell betrachteten Backtrackbaum unabhängig ist. Dazu sei $\tilde{\Gamma} \in R^{k\times n,\lambda,\mu}$ eine weitere Generatormatrix und $(\tilde{g},\tilde{\pi}) \in G \rtimes S_{\mathfrak{P}_0}$ ein Gruppenelement mit $(\tilde{g},\tilde{\pi})\tilde{\Gamma} = (g;\pi)\Gamma$, so dass $((H \rtimes S_{\mathfrak{P}})(\tilde{g},\tilde{\pi}), i)$ ein Knoten im Baum $\overline{T}(\tilde{\Gamma}, G \rtimes S_{\mathfrak{P}_0})$ ist. Somit ist $(\tilde{g},\tilde{\pi})^{-1}(g;\pi)\Gamma = \tilde{\Gamma}$ und beide Bäume isomorph[5]. Damit können wir auch ohne Beschränkung der Allgemeinheit $j = i$ annehmen. Mit der Gleichheit

$$\begin{aligned}\overline{F}(\Gamma, (H \rtimes S_{\mathfrak{P}})(g,\pi), j) &= \overline{F}((\tilde{g},\tilde{\pi})^{-1}(g;\pi)\Gamma, (H \rtimes S_{\mathfrak{P}})(g,\pi)\big((\tilde{g},\tilde{\pi})^{-1}(g;\pi)\big)^{-1}, j)\\ &= \overline{F}(\tilde{\Gamma}, (H \rtimes S_{\mathfrak{P}})(\tilde{g},\tilde{\pi}), j)\end{aligned}$$

ist dann die Wohldefiniertheit der Funktion bewiesen.

Es bleibt die geforderten Eigenschaften einer Fixierreihenfolge zu zeigen. Ohne Beschränkung der Allgemeinheit können wir hierbei auch $(g;\pi) = (1_G;\mathrm{id}_n)$ voraussetzen. Ganz offensichtlich ist die resultierende Folge injektiv und beinhaltet nur Fixpunkte von $S_{\mathfrak{P}}$. Ist nun $(h;\sigma) \in H \rtimes S_{\mathfrak{P}}$ beliebig, so ist

$$\begin{aligned}&F((h;\sigma)\Gamma, (H \rtimes S_{\mathfrak{P}}))\\ &= \textsc{InnerCan}(\mathfrak{P}, \overline{F}(\Gamma, (H \rtimes S_{\mathfrak{P}})(h;\sigma), j), (h;\sigma)\Gamma)\\ &= \textsc{InnerCan}(\mathfrak{P}, \overline{F}(\Gamma, (H \rtimes S_{\mathfrak{P}}), j), (h;\sigma)\Gamma)\\ &= \textsc{InnerCan}(\mathfrak{P}, \overline{F}(\Gamma, (H \rtimes S_{\mathfrak{P}}), j), \Gamma)\end{aligned}$$

Für den Beweis der letzten Gleichheit ist zunächst die Beobachtung entscheidend, dass $\sigma \in S_{\mathfrak{P}}$ die Koordinaten $e \in \mathrm{Fix}_{S_{\mathfrak{P}}}([n])$ unverändert lässt und unser Vorgehen nur auf dieser Spaltenauswahl beruht. Somit können wir ohne Beschränkung der Allgemeinheit auch $\sigma = \mathrm{id}_n$ annehmen. Die H-Invarianz des Resultats zeigt man über eine einfache Induktion, wir überlassen diese dem Leser. □

[5] Genauer gesagt, können wir dies nur induktiv für den Teilbaum bis zu dieser Tiefe zeigen, da in einem derartigen Beweis auch die aktuell zu beweisende Aussage eingehen müsste.

Wir haben nun damit die notwendigen Voraussetzungen geschaffen, um die innere Kanonisierung gemäß Abschnitt 3.3.1 zu implementieren. Insbesondere können wir nun auch im nächsten Abschnitt, welcher die äußere Verfeinerung behandelt, unser sehr umfangreiches Wissen über die Stabilisatoren $G^{(f,\pi\Gamma)}$ einbringen.

5.2.2. Äußere Verfeinerung

Wie bereits angedeutet, wird die äußere Verfeinerung $V^{(a)}$ selbst wieder aus einer iterierten Anwendung des Homomorphieprinzips hervorgehen. In diesem Abschnitt wollen wir nun die Familien von $H \rtimes S_\mathfrak{P}$-Homomorphismen angeben, welche wir zur Definition der Verfeinerung $V^{(a)}$ nutzen werden.

Auf die exakte Beschreibung der Anwendungsreihenfolge der nachfolgenden eingeführten Homomorphismen werden wir aus folgendem Grund nicht mehr weiter eingehen: Die Güte einer Verfeinerung hängt zu stark von der aktuellen Eingabe (d.h. dem aktuellen Knoten im Backtracking) und den bereits durchgeführten Verfeinerungen ab. Eine Untersuchung aller möglichen Verfeinerungen und deren gegenseitige Beeinflussung im Rahmen einer iterierten Verfeinerung gestaltet sich als ein sehr umfangreiches und kaum abschließbares Unterfangen. Die Festlegung einer optimalen Strategie für jede beliebige Eingabe $\Gamma \in R^{k\times n,\lambda,\mu}$ ist also kaum zu gewährleisten.

Wir bemerken hierzu aber auch, dass das eingabespezifische Verhalten der Verfeinerungen bereits bei der Kanonisierung von Graphen zu beobachten ist. Auch hier ist die Festlegung der Reihenfolge, der zur Anwendung zu bringenden Verfeinerungen, immer noch Gegenstand eines über dreißig Jahre währenden und anhaltenden Entwicklungsprozesses.

Eine geeignete Festlegung der Reihenfolge der Verfeinerungen für die Kanonisierung linearer Codes erfolgte bislang nur auf Grundlage der Beobachtung verschiedener Instanzen und ist immer noch Gegenstand aktueller Untersuchungen. Wir geben nun also im Folgenden nur eine Auswahl möglicher Familien von $H \rtimes S_\mathfrak{P}$-Homomorphismen zur möglichen Definition von äußeren Verfeinerungen.

Wir haben bereits festgestellt, dass wir nicht für alle Untergruppen $H \rtimes S_\mathfrak{P} \in \overline{L}(G \rtimes S_{\mathfrak{P}_0})$ einen $H \rtimes S_\mathfrak{P}$-Homomorphismus

$$f_{H\rtimes S_\mathfrak{P}} : R^{k\times n,\lambda,\mu} \to Z^n$$

in eine geeignete, geordnete Menge Z zur Verfügung stellen müssen. In unserem Fall genügt es, sich auf Untergruppen $H \rtimes S_\mathfrak{P}$ zu beschränken, welche als Beschriftung eines Knotens $(H \rtimes S_\mathfrak{P}(g;\pi), j)$ in einem Backtrackbaum $\overline{T}(\Gamma, G \rtimes S_{\mathfrak{P}_0})$ zu einer Matrix $\Gamma \in R^{k\times n,\lambda,\mu}$ auftreten. Zum Beispiel wissen wir stets, dass $H = G^{(f,\pi\Gamma)}$ für $f := \overline{F}(\Gamma, H \rtimes S_\mathfrak{P}(g;\pi), j)$ gilt. Daher können wir den Definitionsbereich der Homomorphismen $f_{H\rtimes S_\mathfrak{P}}$ auf die Menge der Matrizen

$$\bigcup_{\substack{(g;\pi),\Gamma \\ \text{wie oben}}} (H \rtimes S_\mathfrak{P})((g;\pi)\Gamma)$$

einschränken.

Einen ersten Typus von Verfeinerungen erhalten wir aus der Beobachtung der Operation der Gruppe $G^{(f,\pi\Gamma)}$ auf $(g;\pi)\Gamma$. Es gilt:

5.2.2 Hilfssatz. *Für alle $H \rtimes S_\mathfrak{P} \in \overline{\mathcal{L}}(G \rtimes S_{\mathfrak{P}_0})$ definiert die Funktion*

$$f^{(\mathrm{inn})}_{H\rtimes S_\mathfrak{P}} : R^{k\times n,\lambda,\mu} \to (R^k)^n, \quad \Gamma \mapsto (\mathrm{CF}_H(\Gamma_{*,i}))_{i\in[n]}$$

einen $H \rtimes S_\mathfrak{P}$-Homomorphismus, welcher H-invariant ist.

Beweis. Trivial. □

Der Funktionswert $f^{(\mathrm{inn})}_{G^{(f,\pi\Gamma)}\rtimes S_\mathfrak{P}}((g;\pi)\Gamma)$ lässt sich mit den Algorithmen aus dem Abschnitt 5.2.1 berechnen. Jedoch bedeutet diese Berechnung bereits einen erheblichen Aufwand, falls die Anzahl der Erzeuger der Gruppe $G^{(f,\pi\Gamma)}$ für $f := \overline{F}(\Gamma, H \rtimes S_\mathfrak{P}(g;\pi), j)$ groß ist.

Aus den Homomorphismen $f^{(\mathrm{inn})}_{H\rtimes S_\mathfrak{P}}$ lassen sich aber auf vielfältigste Weisen Abschwächungen entwickeln. Formal erhalten wir sie durch Komposition mit einem weiteren $S_\mathfrak{P}$-Homomorphismus. Jedoch haben wir dann im Allgemeinen bessere Methoden zur Verfügung, um das Bild der Komposition zu berechnen. Hierzu mehrere Beispiele:

- Es sei $\kappa := \mathrm{rg}(\Pi_f(\pi\Gamma)) < k$, dann können wir die Matrix $f^{(\mathrm{inn})}_{G^{(f,\pi\Gamma)}\rtimes S_\mathfrak{P}}((g;\pi)\Gamma)$ auf die Zeile mit Index κ projizieren. Diese Funktion definiert also ebenfalls einen $G^{(f,\pi\Gamma)} \rtimes S_\mathfrak{P}$-Homomorphismus $f^{(\mathrm{per},\kappa)}_{G^{(f,\pi\Gamma)}\rtimes S_\mathfrak{P}}$. Andererseits rechnet man leicht nach, dass für $i \in [n]$ und $h \in [m+1]$ der Eintrag $\left(f^{(\mathrm{per},\kappa)}_{G^{(f,\pi\Gamma)}\rtimes S_\mathfrak{P}}((g;\pi)\Gamma)\right)_i$ genau dann gleich θ^h ist, wenn $\mathrm{per}\big(((g;\pi)\Gamma)_{[k]\setminus[\kappa],i}\big) = m - h$ gilt. Mit dieser Äquivalenz lässt sich das Bild des Homomorphismus sehr viel leichter direkt bestimmen.

- Eine zweite Alternative gewinnt man, indem man für jedes $i \in [n]$ den Träger der Spalte $\left(f^{(\mathrm{inn})}_{G^{(f,\pi\Gamma)}\rtimes S_\mathfrak{P}}((g;\pi)\Gamma)\right)_{*,i}$ bestimmt. Man kann sich auch auf den Schnitt mit einer fest vorgegebenen Teilmenge von $[\kappa]$ beschränken.

- Eine weitere Möglichkeit besteht darin, im Anschluss an die Abbildung $f^{(\mathrm{inn})}_{G^{(f,\pi\Gamma)}\rtimes S_\mathfrak{P}}$ eine Reduktion modulo einem Ideal $\mathrm{Rad}(R)^x$ für $x \in [m]$ durchzuführen. Auch hier können wir die Algorithmen aus dem Abschnitt 5.1.2 abwandeln und somit den Funktionswert effizienter berechnen.

Diese Liste von Vorschlägen stellt nur eine kleine Auswahl der Möglichkeiten dar. Wir gehen wegen ihrer Vielzahl auf weitere, so gewonnene Homomorphismen auch nicht mehr weiter ein, zumal sie im Allgemeinen zu Beginn des Backtrackings nicht ausreichend gute Verfeinerungen liefern. Gerade hier ist es aber wichtig, die Anzahl der Kinder eines Knotens möglichst gering zu halten. Aus diesem Grund werden wir auch bei den Knoten geringer Tiefe größere Rechenzeiten zur Berechnung der Verfeinerung aufwenden.

Einen zweiten Typ von Verfeinerungen gewinnen wir über einen alternativen Ansatz. Wir werden hierzu aus der Generatormatrix Γ des R-linearen Codes C einen bipartiten Graphen $(V, E^{(\Gamma)})$ gewinnen und die für Graphen zur Verfügung stehenden Verfeinerungen [55] benutzen. Der Graph wird sich im Wesentlichen aus einer Teilmenge der Codewörter von C ergeben. Dieses Vorgehen hat sich bei linearen Codes über endlichen Körpern bereits als sehr nützlich erwiesen, siehe [24, 51].

Homomorphismen zu Wörtern eines vorgeschriebenen Gewichts Im Folgenden sei mit $\mathcal{B}(V^{(0)}, V^{(1)})$ die Menge aller bipartiten Graphen auf den Punkten $V^{(0)} \dot\cup V^{(1)}$ bezeichnet. Innerhalb der Knotenteilmengen $V^{(0)}$ bzw. $V^{(1)}$ treten also keine Kanten auf.

Wir skizzieren zunächst, wie wir den Graphen $(V, E^{(\Gamma)})$ gewinnen wollen: Zunächst definieren wir $U_\lambda := \bigtimes_{i=0}^{k-1} \left(R/\operatorname{Rad}(R)^{\lambda_i} \right)$. Diese Menge repräsentiert alle möglichen Informationsvektoren[6], welche wir mit der Abbildung

$$\begin{aligned} U_\lambda &\to C \\ (u_0 + \operatorname{Rad}(R)^{\lambda_0}, \ldots, u_{k-1} + \operatorname{Rad}(R)^{\lambda_{k-1}}) &\mapsto (u_0, \ldots, u_{k-1})\Gamma \end{aligned}$$

codieren. Der Übergang zu den Nebenklassen $R/\operatorname{Rad}(R)^{\lambda_i}$ ist notwendig, um eine Bijektion beider Mengen zu erreichen. Informationsvektoren bzw. Codewörter, welche durch Linksmultiplikation mit Ringelementen auseinander hervorgehen, wollen wir im weiteren Verlauf zusammenfassen. Aus diesem Grund beschränken wir uns auch auf diejenigen Vektoren $u \in U'_\lambda := \left\{ u' \in U_\lambda \mid \exists\, i \in [k] : u'_i \in (R/\operatorname{Rad}(R)^{\lambda_i})^* \right\}$, welche mindestens eine Einheit enthalten. Da zwei Vektoren $u, v \in U'_\lambda$ genau dann den gleichen Modul erzeugen, wenn sie durch Multiplikation mit einer Einheit auseinander hervorgehen, werden wir in unserer Formulierung von den Bahnen $R^* \backslash\backslash U'_\lambda$ zu den zyklischen Linksuntermoduln $\overline{U}_\lambda := \{Ru \mid u \in U'_\lambda\}$ von U'_λ übergehen. Mit $n_\lambda := |\overline{U}_\lambda|$ werden wir die entsprechende Kardinalität dieser Menge bezeichnen.

5.2.3 Bemerkung. Wieder werden wir im weiteren Verlauf keine formale Unterscheidung zwischen den Elementen $(u_0 + \operatorname{Rad}(R)^{\lambda_0}, \ldots, u_{k-1} + \operatorname{Rad}(R)^{\lambda_{k-1}}) \in U_\lambda$ und den Vektoren $(u_0, \ldots, u_{k-1}) \in R^k$ vornehmen. Die Wohldefiniertheit der nachfolgenden Definitionen lässt sich jeweils leicht Nachrechnen.

Als Knotenmenge des Graphen $(V, E^{(\Gamma)})$ wählen wir nun $V := [n] \dot\cup \overline{U}_\lambda$. Die Kantenmenge $E^{(\Gamma)} = \bigcup_{j=0}^{m} E^{(\Gamma,j)}$ partitionieren wir in $m+1$ disjunkte Teilmengen

$$E^{(\Gamma,j)} := \{\{i, Ru\} \mid i \in [n], Ru \in \overline{U}_\lambda : \operatorname{per}(u\Gamma_{*,i}) = j\}.$$

Der Graph $(V, E^{(\Gamma)})$ ist vollständig bipartit, d.h. jeder Knoten aus $[n]$ ist mit jedem Knoten aus $\overline{U}_\lambda$ benachbart. Ist $u \in U_\lambda$ beliebig und $w_{\mathrm{sym}}(u\Gamma) = (a_0(u\Gamma), \ldots, a_m(u\Gamma))$,

[6] Bei der Datenübermittlung werden die zu übertragenden Informationen diesen Vektoren zugeordnet, anschließend codiert und übertragen.

so sind für jedes $j \in [m+1]$ genau $a_j(u\Gamma)$ Kanten aus $E^{(\Gamma,j)}$ inzident mit dem Knoten $Ru \in \overline{U}_\lambda$. Die Partitionierung der Kantenmenge können wir auch als eine Färbung mit $m+1$ Farben interpretieren. Operieren wir mit einer Permutation auf diesen kantengefärbten Graphen, so soll die Partitionierung der Kantenmenge hiervon berücksichtigt werden.

5.2.4 Hilfssatz. *Es ist* $(\mathcal{G}, \Psi = (\mathrm{id}_{S_{\mathfrak{P}_0}}, \Psi_1))$ *mit*

$$\begin{aligned} \mathcal{G} : R^{k\times n,\lambda,\mu} &\to \mathcal{B}([n], \overline{U}_\lambda) \qquad \text{und} \qquad \Psi_1 : G \rtimes S_{\mathfrak{P}_0} \to S_{\overline{U}_\lambda} \\ \Gamma &\mapsto (V, E^{(\Gamma)}) \qquad\qquad (A, \varphi; \alpha, \pi) \mapsto \big(Ru \mapsto R\big(\alpha(u)A^{-1}\big)\big) \end{aligned}$$

ein Homomorphismus von Gruppenoperationen.

Beweis. Es seien $\Gamma \in R^{k\times n,\lambda,\mu}$, $(A, \varphi; \alpha, \pi) \in (G \rtimes S_{\mathfrak{P}_0})$, $i \in [n]$ und $Ru \in \overline{U_\lambda}$ beliebig. Weiter sei $\Gamma' := (A, \varphi; \alpha, \pi)\Gamma$. Dann gilt

$$\begin{aligned} \mathrm{per}(u\Gamma_{*,i}) &= \mathrm{per}(\alpha(u)\alpha(\Gamma_{*,i})) \\ &= \mathrm{per}\Big(\alpha(u)A^{-1}A\alpha(\Gamma_{*,i})\varphi^{-1}_{\pi(i)}\Big) = \mathrm{per}\big((\alpha(u)A^{-1})\Gamma'_{*,\pi(i)}\big) \end{aligned}$$

und somit für $j \in [m+1]$

$$\{i, Ru\} \in E^{(\Gamma,j)} \iff \big\{\pi(i), \alpha(u)A^{-1}\big\} = (\Psi(A, \varphi; \alpha, \pi) \cdot \{i, Ru\}) \in E^{(\Gamma',j)}.$$

Es ist also $\mathcal{G}((A, \varphi; \alpha, \pi)\Gamma) = \Psi(A, \varphi; \alpha, \pi) \cdot \mathcal{G}(\Gamma)$. □

Durch den Schritt zu den Graphen $\mathcal{G}(\Gamma)$ haben wir nun ein sehr mächtiges Werkzeug zur Definition weiterer Verfeinerungen an der Hand. Im Backtrackbaum $\overline{T}(\Gamma, G \rtimes S_{\mathfrak{P}_0})$ zu $\Gamma \in R^{k\times n,\lambda,\mu}$ können wir zum Beispiel an einem Knoten $((H \rtimes S_{\mathfrak{P}})(A, \varphi; \alpha, \pi), j)$ den Graphen

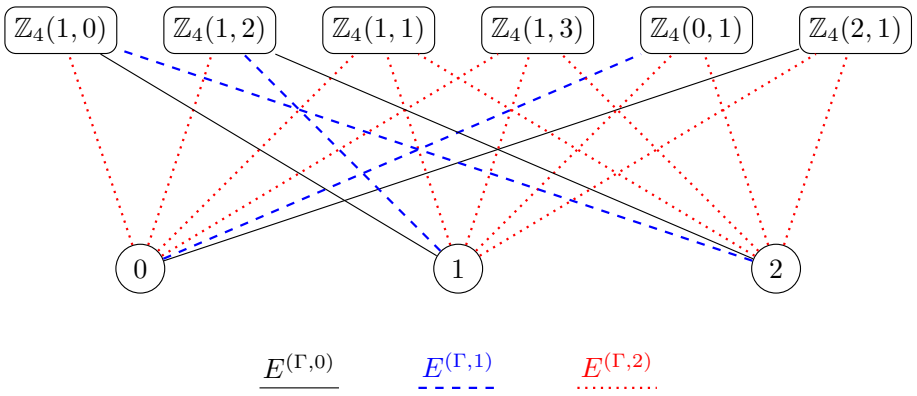
$$\mathcal{G}((A, \varphi; \alpha, \pi)\Gamma) = \Psi(A, \varphi; \alpha, \pi) \cdot \mathcal{G}(\Gamma)$$

bilden und dann über die Komposition mit einem beliebigen $S_{\mathfrak{P}}$-Homomorphismus $f_{\mathfrak{P}} : \mathcal{B}([n], \overline{U}_\lambda) \to Z^n$ Eigenschaften der Koordinaten $i \in [n]$ bestimmen.

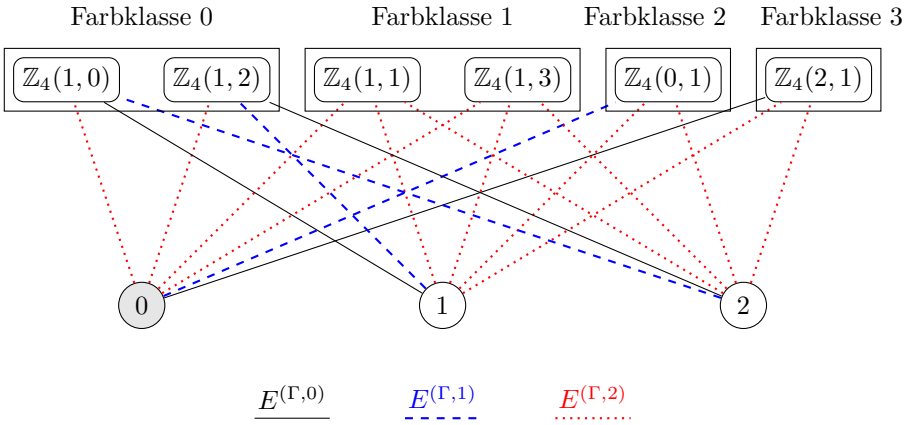
Auch hier haben wir also eine Fülle von Möglichkeiten zur Verfügung, um weitere Verfeinerungen zu gewinnen. Jedoch ist für dieses naive Vorgehen der zu erwartende hohe Aufwand zur Berechnung der Funktionswerte $\mathcal{G}((A, \varphi; \alpha, \pi)\Gamma)$ und $\Psi(A, \varphi; \alpha, \pi)$ sehr problematisch.

Bevor wir uns diesem Problem zuwenden, wollen wir zunächst eine mögliche Definition eines solchen $S_{\mathfrak{P}}$-Homomorphismus $f_{\mathfrak{P}}$ an einem Beispiel vorführen:

5.2.5 Beispiel. Wir betrachten die Generatormatrix $\Gamma = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \end{pmatrix} \in \mathbb{Z}_4^{2\times 3}$. Aus ihr erhalten wir den Graphen $\mathcal{G}(\Gamma)$ in Abbildung 5.1a. In diesem konkreten Beispiel ist jeder Knoten $i \in [3]$ mit jeweils der gleichen Anzahl von gefärbten Kanten aus $E^{(\Gamma,j)}, j \in [3]$ inzident. Wir können also die Knoten $i \in [3]$ zunächst nicht unterscheiden.



(a) ungefärbte Knoten



(b) balancierte Färbung des Graphen

Abbildung 5.1.: Graph $\mathcal{G}(\Gamma)$ zu Beispiel 5.2.5

Jedoch bemerken wir, dass diese Beobachtung für die Knoten aus $\overline{U}_\lambda$ nicht gegeben ist. Partitionieren wir die Knotenmenge $\mathbb{Z}_4 u \in \overline{U_{(2,2)}}$ nach der Färbung der inzidenten Kanten in die angedeuteten Farbklassen, siehe Abbildung 5.1b, so können wir in einem weiteren Schritt die Partition $\{\{0\},\{1,2\}\}$ der Knotenteilmenge $[3]$ gewinnen.

Wir wollen nun beweisen, dass wir stets die balancierte Färbung des Graphen $\mathcal{G}(\Gamma)$, vergleiche Beispiel 5.2.5, zur Definition eines $S_\mathfrak{P}$-Homomorphismus benutzen dürfen. Außerdem werden wir beschreiben, wie wir es vermeiden, Bilder und Urbilder unter dem Homomorphismus Ψ_1 zu berechnen[7].

Dazu wechseln wir unsere Sichtweise auf das Problem: Anstatt der Aufgabe Γ unter der Operation von $G \rtimes S_{\mathfrak{P}_0}$ zu kanonisieren, wollen wir nun eine Kanonisierung des Paars

[7] Dadurch verlieren wir aber gegebenenfalls auch nützliche Informationen für das Backtracking. Dies ist wieder ein Beispiel für das Abwägen zwischen der Komplexität der Verfeinerung und ihrem Nutzen im Backtrackalgorithmus.

$(\Gamma, \mathcal{G}(\Gamma))$ unter der Operation von $(G \rtimes S_{\mathfrak{P}_0}) \times S_{\overline{U}_\lambda}$ anstreben. Dabei wollen wir diese Operation wie folgt definieren:

$$\begin{aligned}\big((G \rtimes S_{\mathfrak{P}_0}) \times S_{\overline{U}_\lambda}\big) \times \big(R^{k\times n,\lambda,\mu} \times \mathcal{B}([n], \overline{U}_\lambda)\big) &\to \big(R^{k\times n,\lambda,\mu} \times \mathcal{B}([n], \overline{U}_\lambda)\big)\\ (((A, \varphi; \alpha, \pi), \sigma), (\Gamma, B)) &\mapsto ((A, \varphi; \alpha, \pi)\Gamma, (\pi, \sigma)B)\end{aligned}$$

5.2.6 Bemerkung. Wir wollen betonen, dass wir diese weitere Sichtweise nur aus einem einzigen Grund einführen: Sie erlaubt eine wesentlich einfachere Beschreibung der Verfeinerungen. Wir können die nachfolgenden Homomorphismen auch auf dem ursprünglichen Problem definieren, jedoch muss hierbei sehr umständlich die aktuelle Datenlage beschrieben werden.

Aus dem Ergebnis der Kanonisierung $\mathrm{Can}_{(G\rtimes S_{\mathfrak{P}_0})\times S_{\overline{U}_\lambda}}(\Gamma, \mathcal{G}(\Gamma))$ können wir leicht einen Kanonisierer für unser Ausgangsproblem gewinnen, denn es gilt dann für $\Gamma, \Gamma' \in R^{k\times n,\lambda,\mu}$:

$$\Gamma' \in (G \rtimes S_{\mathfrak{P}_0})\Gamma \iff (\Gamma', \mathcal{G}(\Gamma')) \in \big((G \rtimes S_{\mathfrak{P}_0}) \times S_{\overline{U}_\lambda}\big)(\Gamma, \mathcal{G}(\Gamma))$$

Wir nehmen also im weiteren Verlauf an, dass

- wir einen Backtrackbaum $V\big((\Gamma, \mathcal{G}\Gamma), (G \rtimes S_{\mathfrak{P}_0}) \times S_{\overline{U}_\lambda}\big)$ aufbauen werden, bei welchem die Knoten eine Beschriftung der Gestalt $((H \rtimes S_{\mathfrak{P}}) \times S_{\mathfrak{Q}})(g, \pi, \sigma)$ tragen, wobei $\mathfrak{P}$ und $\mathfrak{Q}$ kanonische Partitionen[8] von $[n]$ bzw. $\overline{U}_\lambda$ sind.
- Wir wollen die Partitionierung über eine Individualisierung einer Koordinate aus $[n]$ gewinnen,
- und die Verfeinerungen, welche wir bislang kennen gelernt haben – insbesondere die innere Minimierung –, genau in dieser Form auch in diesem Backtracking zur Anwendung bringen.

Wir zeigen jetzt nur noch, wie wir aus dem Graphenanteil weitere Verfeinerungen von $\mathfrak{P}$ und $\mathfrak{Q}$ gewinnen werden. Wir benutzen als Beispiel für einen $(S_{\mathfrak{P}} \times S_{\mathfrak{Q}})$-Homomorphismus wieder das Zählen der Nachbarn nach Farbe:

5.2.7 Definition. Ist $B = (V, E) \in \mathcal{B}(V^{(0)}, V^{(1)})$ ein bipartiter Graph und $\mathfrak{p}$ eine Partition von $V^{(0)}$, so wollen wir für $v_1 \in V_1$ mit

$$N_{\mathfrak{p}}(B, v_1) = (|\{v_0 \in P_j \mid \{v_0, v_1\} \in E\}|)_{P_j \in \mathfrak{p}}$$

das Zählen der Nachbarn von v_1 nach den Farbklassen $P_j \in \mathfrak{p}$ der Knotenmenge $V^{(0)}$ bezeichnen.

[8] Bezüglich einer fest gewählten Anordnung von $\overline{U_\lambda}$ können wir wieder Partitionen von konsekutiven Mengen als kanonische Partitionen bezeichnen.

Da der Graph B bipartit ist, unterscheidet sich diese Definition insofern zu der Definition des Zählens aus Abschnitt 3.2.4, dass die Anzahl der Nachbarn nur für die komplementäre Knotenmenge berechnet wird.

5.2.8 Definition. Für einen kantengefärbten, bipartiten Graphen $B = (V, E)$ mit $V = V^{(0)} \dot\cup V^{(1)}$ und $E = (E^{(0)}, \dots, E^{(m)}) \subseteq (V^{(0)} \times V^{(1)})^{m+1}$ sei dann entsprechend für eine Partition $\mathfrak{p}$ von $V^{(0)}$

$$M_\mathfrak{p}(B) := \Big(\big(N_\mathfrak{p}((V, E^{(j)}), v_1) \big)_{j \in [m+1]} \Big)_{v_1 \in V_1}$$

eine Abzählung der Nachbarn von $v_1 \in V^{(1)}$ nach Knoten- und Kantenfärbung.

Es lässt sich sehr leicht nachrechnen, dass die Funktion $M_\mathfrak{p}$ einerseits einen S_{V_1}-Homomorphismus definiert und andererseits $S_\mathfrak{p}$-invariant ist. Hierauf baut nun der nachfolgende Hilfssatz auf:

5.2.9 Hilfssatz. *Es sei $H \rtimes S_\mathfrak{P} \in \overline{\mathcal{L}}(G \rtimes S_{\mathfrak{P}_\circ})$ und $\mathfrak{Q}$ eine kanonische Partition von $\overline{U}_\lambda$, dann definiert die Abbildung*

$$f^{(\mathrm{neigh})}_{(H \rtimes S_\mathfrak{P}) \times S_\mathfrak{Q}} : R^{k \times n, \lambda, \mu} \times \mathcal{B}([n], \overline{U}_\lambda) \to \left(\mathbb{Z}^{\leq n_\lambda}\right)^{(m+1) \times n} \times \left(\mathbb{Z}^{\leq n}\right)^{(m+1) \times n_\lambda}$$
$$(\Gamma, B) \mapsto (M_\mathfrak{Q}(B), M_\mathfrak{P}(B))$$

einen $(S_\mathfrak{P} \times S_\mathfrak{Q})$-Homomorphismus, welcher H-invariant ist.

Beweis. Es sei $((h; \pi), \sigma) \in (H \rtimes S_\mathfrak{P}) \times S_\mathfrak{Q}$ und $(\Gamma, B) \in R^{k \times n, \lambda, \mu} \times \mathcal{B}([n], \overline{U}_\lambda)$ beliebig. Dann ist

$$\begin{aligned} f^{(\mathrm{neigh})}_{(H \rtimes S_\mathfrak{P}) \times S_\mathfrak{Q}}(((h; \pi), \sigma)(\Gamma, B)) &= (M_\mathfrak{Q}((\pi, \sigma)B), M_\mathfrak{P}((\pi, \sigma)B)) \\ &= (\pi M_\mathfrak{Q}(B), \sigma M_\mathfrak{P}(B)) = (\pi, \sigma) \cdot f^{(\mathrm{neigh})}_{(H \rtimes S_\mathfrak{P}) \times S_\mathfrak{Q}}(\Gamma, B) \end{aligned}$$ □

5.2.10 Bemerkung. Das Zählen der Nachbarn nach Knoten- und Kantenfärbung stellt wieder nur eine Möglichkeit dar, um aus den Graphen B Informationen zum Verfeinern zu gewinnen. Tatsächlich können beliebige $(S_\mathfrak{P} \times S_\mathfrak{Q})$-Homomorphismen zum Einsatz gebracht werden.

5.2.11 Bemerkung. Anstatt zu einer Operation von $(G \rtimes S_{\mathfrak{P}_0}) \times S_{\overline{U}_\lambda}$ auf $R^{k \times n, \lambda, \mu} \times \mathcal{B}([n], \overline{U}_\lambda)$ überzugehen, könnte man auch direkt die Operation mit $(G \rtimes S_{\mathfrak{P}_0})$ auf der gleichen Menge untersuchen. Jedoch muss man dann, Bilder und Urbilder unter Ψ_1 berechnen. Dies führt zu einem erheblichen, zusätzlichen Aufwand.

In dem von uns beschriebenen Vorgehen umgehen wir diese Situation folgendermaßen: An einem Knoten $(((H \rtimes S_\mathfrak{P}) \times S_\mathfrak{Q})(g, \pi, \sigma), j)$ schätzen wir schlichtweg das Bild $\Psi_1(Hg) \subseteq S_\mathfrak{Q}\sigma$ grob über die Nebenklasse $S_\mathfrak{Q}\sigma$ ab. Mit dieser Nebenklasse einer kanonischen Young-Untergruppe lässt sich überdies, im Gegensatz zu dem anderen Vorgehen, viel effizienter arbeiten.

Der Übergang zu den Graphen $\mathcal{G}(\Gamma)$ entspricht dem Übergang von der Generatormatrix Γ zu dem von ihr erzeugten linearen Code C und impliziert somit einen exponentiellen Anstieg der Rechenzeit. Dies ist zwar nach unseren Beobachtungen aus Abschnitt 2.4 prinzipiell gerechtfertigt, da wir ohnehin kein polynomielles Laufzeitverhalten erwarten können, jedoch sollten wir versuchen, den Rechenaufwand so stark wie möglich zu reduzieren:

Wir ziehen uns daher auf kleinere Graphen $\widetilde{\mathcal{G}}(\Gamma) \in \mathcal{B}([n],[\ell])$ zurück. Es ist leicht nachzuvollziehen, dass unsere bisherige Argumentation weiterhin gültig bleibt, solange die Zuordnung auf den Graphen $\widetilde{\mathcal{G}}(\Gamma)$ über einen Homomorphismus von Gruppenoperationen modelliert werden kann, d.h. es existiert ein Gruppenhomomorphismus $\Omega : G \to S_\ell$, so dass $\widetilde{\mathcal{G}}((g;\pi)\Gamma) = (\pi, \Omega(g))\widetilde{\mathcal{G}}(\Gamma)$ für alle $(g;\pi) \in G \rtimes S_{\mathfrak{P}_0}$ und $\Gamma \in R^{k\times n,\lambda,\mu}$ gilt.

Wir beobachten hierzu, dass die Verfeinerung mittels $f^{(\text{neigh})}_{(H\rtimes S_{\mathfrak{P}})\times S_{\mathfrak{Q}}}$ die Knoten in $\overline{U}_\lambda$ des Graphen $\mathcal{G}(\Gamma)$ nach deren symmetrisierten Gewicht unterscheidet. Wir können und werden uns daher zum Beispiel schon von Beginn an auf bestimmte symmetrisierte Gewichte $W \subseteq \{w \in \mathbb{N}^{m+1} \mid \sum_{j=0}^{m+1} w_j = n\}$ festlegen und dann nur denjenigen Teilgraphen $\widetilde{\mathcal{G}}(\Gamma)$ aufbauen, dessen Knotenmenge aus $[n]$ und $\overline{W}_\lambda(\Gamma) := \{Ru \mid u \in U'_\lambda : w_{\text{sym}}(u\Gamma) \in W\}$ besteht. Über eine beliebige Bijektion der Menge $\overline{W}_\lambda(\Gamma)$ nach $[\ell]$, $\ell = |\overline{W}_\lambda(\Gamma)|$ schaffen wir dann die formalen Voraussetzungen.

Bei der Wahl der symmetrisierten Gewichte W gilt es wieder, ein vernünftiges Mittelmaß zwischen der Kardinalität der Menge $\overline{W}_\lambda(\Gamma)$ und deren Nutzen für das Backtracking zu finden. Für lineare Codes V über endlichen Körpern beschreibt J. Leon [51] diese Auswahl folgendermaßen[9]:

> *„The primary requirement for the algorithm to operate efficiently is that there be known in advance a set W of vectors invariant under $\text{Aut}(V)$ which is reasonably small (several thousand elements at most) and yet which contains "enough structure". Roughly speaking, an invariant set W will contain enough structure if $\text{Aut}(V)$ has relatively small index in the group of all monomial permutations of the coordinate set mapping W onto itself. Often it suffices to choose W to be the set of minimal weight vectors or, if this set is exceptionally small, the set of vectors of minimal or next to minimal weight."*

Eine ähnliche Aussage gilt auch für unser Vorgehen, denn beide Verfahren basieren auf der gleichen Grundidee. Bei linearen Codes über endlichen Körpern gehen wir daher wie von J. Leon beschrieben vor und wählen alle Informationsvektoren zu Codewörtern mit minimalem, beziehungsweise nach oben beschränktem Hamming-Gewicht.

Für lineare Codes über endlichen Kettenringe gestaltet sich auch hier die Feststellung des Informationsgehalts der Menge $\overline{W}_\lambda(\Gamma)$ als schwieriger. Wir setzen hier die folgende Heuristik ein:

[9] Das symmetrisierte Gewicht beschreibt im Fall eines endlichen Körpers $\mathbb{F}_q$ die Unterscheidung des Nullelements von allen weiteren Körperelementen, d.h. das Hamming-Gewicht.

Zunächst beinhaltet die Menge W nur das kleinste symmetrisierte Gewicht (bzgl. der lexikographischen Ordnung). Es sei $\ell := |\overline{W}_\lambda(\Gamma)|$ und $\mathfrak{D}$ die diskrete, kanonische Partition von $[\ell]$. Dann berechnen wir den Vektor $m = M_{\mathfrak{D}}(\mathcal{G}(\Gamma)) \in \mathbb{N}^{\ell \times n}$ und bestimmen die Koordinatenmenge $I := \{i \in [n] : |\{j \in [n] : m_i = m_j\}| \leq 2\}$. Hat dann die Teilmatrix $\Gamma_{*,I}$ den Umriss λ, so akzeptieren wir die Menge W. Andernfalls fügen wir das nächstgrößere symmetrisierte Gewicht zu W hinzu und wiederholen das Vorgehen.

5.2.12 Bemerkung. Auch hier wurde noch nicht abschließend untersucht, ob diese Forderung an die Menge W nicht bereits zu stark oder zu schwach sind. Jedoch wurde mit dieser Wahl für die untersuchten Instanzen ein angemessenes Laufzeitverhalten erzielt.

5.2.3. Zur Kanonizität der kanonischen Repräsentanten bei isomorphen Ringen

Nun wollen wir noch die wichtige Frage klären, inwieweit das Resultat der Kanonisierung von dem gewählten Ring R und den Ringelementen $\xi, \theta \in R$ abhängig ist. Hierzu sei R' ein zu R isomorpher Kettenring, insbesondere wollen wir auch den Fall $R = R'$ betrachten. Weiter sei ξ' ein Erzeuger einer Teichmüller-Menge von R' und θ' ein Erzeuger von $\mathrm{Rad}(R')$, so dass $\tau^e(\xi')\theta' = \theta'\xi'$ gilt. Weitere Annahmen haben wir für die Beschreibung des Kanonisierers nicht getroffen.

Die von den fest gewählten Ringelementen $\xi, \theta \in R$ abhängige Kanonisierung der Generatormatrizen über R wollen wir im Folgenden mit $\mathrm{CF}_{G \rtimes S_{\mathfrak{P}_0}} : R^{k \times n, \lambda, \mu} \to R^{k \times n, \lambda, \mu}$ bezeichnen. Analog sei $\mathrm{CF}_{G' \rtimes S_{\mathfrak{P}_0}} : R'^{k \times n, \lambda, \mu} \to R'^{k \times n, \lambda, \mu}$ die analoge Kanonisierung der Generatormatrizen über R', die durch die Wahl von $\xi', \theta' \in R'$ und dem gleichen Ablauf eindeutig bestimmt sei.

5.2.13 Satz. *Definiert $\chi_{\xi'}^{\theta'} : R \to R'$ einen Ringisomorphismus von R nach R', so gilt für die Kanonisierungen* $\mathrm{CF}_{G' \rtimes S_{\mathfrak{P}_0}} \circ \chi_{\xi'}^{\theta'} = \chi_{\xi'}^{\theta'} \circ \mathrm{CF}_{G \rtimes S_{\mathfrak{P}_0}}$.

Beweis. Dieser Satz ist im Wesentlichen eine direkte Konsequenz aus Satz 3.2.13 und der Tatsache, dass $\chi_{\xi'}^{\theta'}$ ordnungserhaltend ist, siehe Folgerung 4.2.3. Dabei definieren wir den notwendigen Gruppenisomorphismus Ψ um von isomorphen Gruppenoperationen sprechen zu können über

$$\begin{aligned} \Psi : G \rtimes S_{\mathfrak{P}_0} &\to G' \rtimes S_{\mathfrak{P}_0} \\ (A, \varphi; \alpha, \pi) &\mapsto (\chi_{\xi'}^{\theta'}(A), \chi_{\xi'}^{\theta'}(\varphi); \chi_{\xi'}^{\theta'} \circ \alpha \circ (\chi_{\xi'}^{\theta'})^{-1}, \pi). \end{aligned}$$

Für den weiteren Beweis der Aussage muss man lediglich zeigen, dass sich die in Abschnitt 5.2.2 definierten $(H \rtimes S_{\mathfrak{P}})$-Homomorphismen $f_{H \rtimes S_{\mathfrak{P}}}$ für $H \rtimes S_{\mathfrak{P}} \in \overline{\mathcal{L}}(G \rtimes S_{\mathfrak{P}_0})$ bzw. $(H' \rtimes S_{\mathfrak{P}})$-Homomorphismen $f_{H' \rtimes S_{\mathfrak{P}}}$ für $H' \rtimes S_{\mathfrak{P}} \in \overline{\mathcal{L}}(G' \rtimes S_{\mathfrak{P}_0})$ auch tatsächlich derart verhalten, dass stets $f_{H \rtimes S_{\mathfrak{P}}} = f_{\Psi(H \rtimes S_{\mathfrak{P}})} \circ \chi_{\xi'}^{\theta'}$ gilt.

Wir wollen diese Aussage exemplarisch am Beispiel der innere Kanonisierung zeigen. Da eine ausführliche Diskussion sich als zu umfangreich erweist, geben wir nur kurz

die Beweisidee an: Es sei $\Gamma \in R^{k\times n,\lambda,\mu}$ eine beliebige Generatormatrix. Wir nutzen die (ξ,θ)-adische Entwicklung der Elemente des Rings. Hierzu sei $\Xi := \begin{pmatrix}\xi^{r-1} & \dots & \xi^0\end{pmatrix}$ und $\Theta := \begin{pmatrix}\theta^0 \\ \vdots \\ \theta^{m-1}\end{pmatrix}$, dann gilt

$$\Gamma = \begin{pmatrix}\Xi & 0 & \dots & 0 \\ 0 & \Xi & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \Xi\end{pmatrix} \underbrace{\begin{pmatrix}\operatorname{coeff}(\Gamma_{0,0}) & \dots & \operatorname{coeff}(\Gamma_{0,n-1}) \\ \vdots & & \vdots \\ \operatorname{coeff}(\Gamma_{k-1,0}) & \dots & \operatorname{coeff}(\Gamma_{k-1,n-1})\end{pmatrix}}_{=:\operatorname{coeff}(\Gamma)\in[p]^{kr\times mn}} \begin{pmatrix}\Theta & 0 & \dots & 0 \\ 0 & \Theta & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \Theta\end{pmatrix}$$

Die innere Kanonisierung zielt nun gerade darauf ab, Einträge der Matrix $\operatorname{coeff}(\Gamma)$ in einer fest vorgegebenen Reihenfolge zu minimieren. Wechseln wir mittels $\chi_{\xi'}^{\theta'}$ in den isomorphen Ring R', so bleibt die Matrix $\operatorname{coeff}(\Gamma) = \operatorname{coeff}(\Gamma')$ hiervon unberührt. □

5.2.14 Folgerung. *Für eine unter Ringisomorphismen $\chi_{\xi'}^{\theta'}$ unabhängige Darstellung des kanonischen Repräsentanten $\Gamma \in R^{k\times n}$ einer Bahn bietet es sich also an, statt Γ die Matrix* $\operatorname{coeff}(\Gamma) \in [p]^{rk\times mn}$ *der Koeffizienten der (ξ,θ)-adischen Entwicklung eines jeden Eintrags von Γ zu speichern bzw. zu vergleichen.*

5.2.15 Beispiel. Wie wir wissen, definiert nicht jede zulässige Wahl von ξ' und θ' auch einen Ringisomorphismus sondern unter Umständen nur eine Bijektion $\chi_{\xi'}^{\theta'}$ von R nach R', siehe Beispiel 4.2.2. Somit ist es auch nicht erstaunlich, dass wir mittels unseres Programms, siehe Kapitel 7, leicht eine Generatormatrix

$$\Gamma := \begin{pmatrix}1 & 0 & 4 & 0 & 8 \\ 0 & 1 & 1 & 8 & 4\end{pmatrix} \in \mathbb{Z}_9^{2\times 5}$$

über $\mathbb{Z}_9$ bestimmen können, für welche dann der vorausgegangene Satz (mit $\theta = 3$ und $\theta' = 6$) nicht gilt:

$$\chi_8^6(\mathrm{CF}_{G\rtimes S_5}(\Gamma)) = \begin{pmatrix}1 & 1 & 0 & 2 & 1 \\ 0 & 0 & 1 & 1 & 1\end{pmatrix} \neq \mathrm{CF}_{G'\rtimes S_5}(\chi_8^6(\Gamma)) = \begin{pmatrix}1 & 1 & 0 & 8 & 1 \\ 0 & 0 & 1 & 1 & 1\end{pmatrix}.$$

Da wir im Allgemeinen eine größere Wahlfreiheit für die Festlegung von θ' und ξ' haben, als wir sie durch die Menge aller Ringisomorphismen abdecken können, müssen wir die ausgewählten Elemente ξ, ξ' bzw. θ, θ' also noch weiter auszeichnen. Bei endlichen Körpern $\mathbb{F}_{p^r}$ haben wir in [24, Abschnitt 6] etwa verlangt, dass ξ eine Nullstelle des eindeutig bestimmten Conway-Polynoms vom Grad r über $\mathbb{F}_p$ ist. Da die Automorphismengruppe von $\mathbb{F}_{p^r}$ transitiv auf der Nullstellenmenge operiert, reicht diese Forderung aus, um die Kanonizität unabhängig von der tatsächlichen Darstellung von $\mathbb{F}_{p^r}$ zu gewährleisten.

Bei Galois-Ringen $\mathrm{GR}(p^m, r)$ ist die Teichmüller-Gruppe T eindeutig. Wir können also auch hier diejenigen Erzeuger ξ von T auszeichnen, für welche $\overline{\xi} \in \mathbb{F}_{p^r}$ eine Nullstelle des

Conway-Polynoms vom Grad r über $\mathbb{F}_p$ ist. Wieder operiert die Automorphismengruppe von $\mathrm{GR}(p^m, r)$ transitiv auf dieser Menge. Die Festlegung eines ausgezeichneten Elements θ ist hier ebenfalls sehr einfach, wir wählen $\theta = p$.

Für beliebige isomorphe Kettenringe R und S führt eine eindeutig bestimmbare Auszeichnung von ξ, θ bzw. ξ', θ' aber auf das derzeit noch ungelöste Problem der Klassifikation der Kettenringe. Sind unsere Wahlen von θ' und ξ' also nicht zufällig derart, dass $\chi_{\xi'}^{\theta'}$ einen Automorphismus definiert, so können wir uns bei der Kanonisierung nur damit behelfen, indem wir einen der beiden Ringe samt den Erzeugern ξ, θ bzw. ξ', θ' auszeichnen und in diesem arbeiten.

6. Modifikationen & Anwendungen

Dieses Kapitel soll nun vor allem aufzeigen, dass der Kanonisierer aus dem Abschnitt 5.2 bzw. dessen Implementierung, siehe Kapitel 7, auch für größere und interessante Codes immer noch praktikabel ist und kanonische Repräsentanten vor allem bei Klassifikationsproblemen den entscheidenden Vorteil bringen. Außerdem wollen wir in diesem Kapitel auch mögliche Modifikationen des Algorithmus diskutieren, welche es erlauben, den Kanonisierer auch in weiteren Teilgebieten der angewandten Mathematik, wie etwa der Kryptographie, einzusetzen. Es zeigt sich, dass der von uns beschriebene Ansatz sehr flexibel ist. Wir werden aber auch in Abschnitt 6.2.4 die Grenzen unseres Ansatzes aufzeigen.

6.1. Lineare Codes über Galois-Ringen der Charakteristik 4

6.1.1. Klassifikation verallgemeinerter Teichmüller-Codes

In [45, Abschnitt 3.1] wurde eine Familie $\mathcal{T}_{q,k,s}$ von linearen Codes über Galois-Ringen $R = \mathrm{GR}(4, r)$ der Charakteristik 4 konstruiert, welche hervorragende Parameter besitzen. Wir gehen später nochmals kurz auf die entscheidenden Schritte der Konstruktion ein. Die Codes $\mathcal{T}_{q,k,s}$ werden *verallgemeinerte Teichmüller-Codes* genannt. Der Parameter $q = 2^r$ beschreibt hierbei, wie bisher, den Restklassenkörper $\mathbb{F}_q \simeq R/\operatorname{Rad}(R)$.

Über die Gray-Abbildung können wir die verallgemeinerten Teichmüller-Codes $\mathcal{T}_{q,k,s}$ der Länge $n := 2^s \frac{q^k-1}{q-1}$ mit den linearen Codes der Länge nq und der gleichen Mächtigkeit q^{2k} über $\mathbb{F}_q$ vergleichen. Die Hamming-Minimaldistanz des Bildes unter der Gray-Abbildung ist in vielen Fällen besser als diejenige, welche alle bekannten linearen Codes mit gleichen Parametern erreichen. Die verallgemeinerten Teichmüller-Codes sind also BTKL-Codes.

Das Konstruktionsprinzip zur Definition dieser Codes nutzt die in Folgerung 3.1.10 gemachte Beobachtung über den Zusammenhang zwischen linearen Codes und den Multimengen der von den Spalten einer Generatormatrix erzeugten, zyklischen R-Rechtsmoduln. Der Code $\mathcal{T}_{q,k,s}$ bestimmt sich hierbei über die Auswahl einer geeigneten Teilmenge $\mathfrak{T}_{q,k,s}$ der Punktmenge (freie Moduln vom Rang 1) der projektiven Rechts-Hjelmslev-Geometrie $\mathrm{PHG}(R_R^k)$, $k \geq 2$.

Wir werden die notwendigen Schritte zur Definition der Punktmenge $\mathfrak{T}_{q,k,s}$ nur kurz skizzieren, für Details verweisen wir auf Originalliteratur [45]:

1. Man betrachtet eine Ringerweiterung $\mathrm{GR}(4,r) \subseteq \mathrm{GR}(4,rk) =: S$ und nutzt die Isomorphie der freien R-Rechtsmoduln S_R und R_R^k .

2. Anschließend wählt man einen Untervektorraum U des $\mathbb{F}_2$-Vektorraums $\mathbb{F}_{q^k}$ der Dimension $\dim(U) = s + r$, welcher $\mathbb{F}_q$ enthält. Dieser definiert eine Untergruppe $R^* \leq \Sigma_U \leq S^*$ der multiplikativen Gruppe und induziert eine Punktmenge $\mathrm{pts}(\Sigma_U) := \{xR \mid x \in \Sigma_U \subset S\}$ der Kardinalität $2^s \frac{q^k-1}{q-1}$.

3. Für gewisse Einschränkungen an s und an die Unterräume U, siehe [45, Satz 3.1.8], setzen wir die Punktmenge $\mathfrak{T}^{(U)}_{q,k,s}$ gleich $\mathrm{pts}(\Sigma_U)$.

In den Fällen $s = 0$ bzw. $s = (k-1)r$ existiert genau ein solcher Unterraum $U = \mathbb{F}_q$ bzw. $U = \mathbb{F}_{q^k}$. Daher ist der Code $\mathcal{T}_{q,k,s}$ bis auf Isomorphie eindeutig bestimmt. In allen weiteren Fällen gibt es jedoch für fest gewählte Parameter q, k, s mehrere Möglichkeiten zur Auswahl des Untervektorraums U.

Mit den Beispielen 3.1.16 und 3.1.17 in [45] wurde nachgewiesen, dass sich aus den verschiedenen Wahlmöglichkeiten für U auch nicht isomorphe $\mathbb{Z}_4$-lineare Codes $\mathcal{T}_{2,5,2}$ bzw. $\mathcal{T}_{2,6,1}$ ergeben können. Die Aussage wurde hierbei über eine Implementierung unseres Kanonisierers in C++, siehe Abschnitt 7.2, bewiesen. Wir greifen diese Aussage zu den verallgemeinerten Teichmüller-Codes $\mathcal{T}_{2,6,1}$ im Beispiel 6.1.2 nochmals auf.

Wir wollen nun diese Untersuchung auch auf die Teichmüller-Codes über dem nächstgrößeren[1] Galois-Ring $\mathrm{GR}(4,2)$ fortsetzen. Bevor wir die Beispiele bearbeiten, bemerken wir an dieser Stelle noch, dass die Automorphismengruppe $\mathrm{Aut}(\mathfrak{T}_{q,k,s}) \leq \Gamma\mathrm{L}_k(R)$ der Punktmenge $\mathfrak{T}_{q,k,s}$ stets die Ordnung $|\mathrm{Aut}(\mathfrak{T}_{q,k,s})| = \frac{|\mathrm{Aut}(\mathcal{T}_{q,k,s})|}{|Z(R)|}$ hat.

6.1.1 Bemerkung. Laut [45, Lemma 3.1.7] operiert die Gruppe Σ_U/R^* scharf transitiv auf $\mathrm{pts}(\Sigma_U)$. Dies erklärt, dass wir im Folgenden stets eine transitive Operation von $\mathrm{Aut}(\mathcal{T}_{q,k,s})\downarrow S_n$ auf der Koordinatenmenge $[n]$ des Codes $\mathcal{T}_{q,k,s}$ bzw. von $\mathrm{Aut}(\mathfrak{T}_{q,k,s})$ auf der Punktmenge $\mathfrak{T}_{q,k,s}$ beobachten können.

6.1.2 Beispiel. Es gibt 16 Unterräume, welche zur Definition eines $\mathbb{Z}_4$-linearen Teichmüller-Codes $\mathcal{T}_{2,6,1}$ herangezogen werden können. Diese Menge partitioniert sich in

- zwei Isomorphieklasse mit je 6 Codes, deren Automorphismengruppen jeweils die Ordnung $126 \cdot |Z(\mathbb{Z}_4)| = 252$ haben,
- eine Isomorphieklasse mit 3 Codes, deren Automorphismengruppen die Ordnung $252 \cdot 2$ haben,
- und einen weiteren, dazu nicht isomorphen Code mit einer Automorphismengruppe der Ordnung $756 \cdot 2$.

[1] Für noch größere Galois-Ringe $\mathrm{GR}(4,r)$, $r \geq 3$ ist die Kardinalität der Codes $\mathcal{T}_{q,k,s}$ bereits zu groß, um sie mit dem Kanonisierer behandeln zu können.

Für die Codes aus den ersten beiden Isomorphieklassen beobachten wir eine Rechenzeit zwischen 150 und 220 Sekunden bei einer durchschnittlichen Laufzeit von 190 Sekunden. Bei der dritten Klasse von Codes steigt die durchschnittliche Rechenzeit auf 220 Sekunden an. Für den Code mit der größten Automorphismengruppe beobachten wir einen weiteren Anstieg der Rechenzeit auf ungefähr 4000 Sekunden. Dieser erhebliche Anstieg erklärt sich dadurch, dass der Backtrackalgorithmus im Gegensatz zu den anderen Instanzen auch Blätter erreicht, welche nicht zu der schlussendlich bestimmten kanonischen Form korrespondieren. Gegebenenfalls ließe sich das Laufzeitverhalten mit einer Überarbeitung der eingesetzten äußeren Verfeinerung also noch weiter verbessern.

6.1.3 Beispiel. Zur Definition einer Punktmenge $\mathfrak{T}^{(U)}_{4,3,2}$ stehen 20 geeignete Unterräume U zur Auswahl. Die Menge der resultierenden linearen Codes $\mathcal{T}^{(U)}_{4,3,2}$ und damit auch die Menge der Punktmengen $\mathfrak{T}^{(U)}_{4,3,2}$ wird durch die Gruppenoperation in 4 Isomorphieklassen partitioniert:

- drei dieser Klassen beinhalten je sechs lineare Codes mit einer Automorphismengruppe der Ordnung $84 \cdot |Z(\mathrm{GR}(4,2))|$,
- eine weitere enthält genau zwei lineare Codes mit einer Automorphismengruppe der Ordnung $252 \cdot |Z(\mathrm{GR}(4,2))|$.

6.1.4 Beispiel. Zur Definition einer Punktmenge $\mathfrak{T}^{(U)}_{4,4,2}$ hat man 256 Möglichkeiten zur Wahl des Unterraums U. Jeder verallgemeinerte Teichmüller-Code $\mathfrak{T}^{(U)}_{4,4,2}$ hat eine Automorphismengruppe der Ordnung $340 \cdot |Z(\mathrm{GR}(4,2))|$. Je acht Unterräume definieren semilinear isometrische Codes.

Ebenso verteilen sich die 320 möglichen Codes $\mathcal{T}^{(U)}_{4,4,4}$ auf 40 Isomorphieklassen mit je 8 linearen Codes. Jeder Code $\mathcal{T}^{(U)}_{4,4,4}$ hat eine Automorphismengruppe der Ordnung $1360 \cdot |Z(\mathrm{GR}(4,2))|$.

Laufzeiten zu den hier behandelten und weiteren Beispielen finden sich in der Tabelle 6.1. Bei dem Programmaufruf handelt es sich in allen Fällen um eine Implementierung des Kanonisierers in Sage [70], siehe Abschnitt 7.1.2. Dabei sei mit $t_{\min}$, $t_{\max}$ und $t_{\varnothing}$ die minimale, maximale und durchschnittliche Laufzeit (in Minuten) auf den Instanzen $\mathcal{T}^{(U)}_{q,k,s}$ bezeichnet. Die letzte Tabellenspalte gibt die Anzahl der Isomorphieklassen in Abhängigkeit von der Automorphismengruppe und der Anzahl zueinander isomorpher Codes an. Ein Eintrag $(a,b)^x$ soll hier die Bedeutung haben, dass es x Isomorphieklassen von linearen Codes gibt, welche b Elemente $\mathcal{T}^{(U)}_{q,k,s}$ enthalten und deren Automorphismengruppen jeweils die Ordnung $\left|\mathrm{Aut}\left(\mathcal{T}^{(U)}_{q,k,s}\right)\right| = a \cdot |Z(R)|$ haben.

6.1.5 Folgerung. *Die Verteilung aller untersuchten linearen Codes $\mathcal{T}^{(U)}_{q,k,s}$, für $U \leq \mathbb{F}_{q^k}$ zulässig, auf Isomorphieklassen hat stets die Eigenschaft, dass die Produkte*

$$\left|\mathrm{Aut}(\mathcal{T}^{(U)}_{q,k,s})\right| \cdot \left|\left\{\mathcal{T}^{(U')}_{q,k,s} \mid \mathcal{T}^{(U')}_{q,k,s} \text{ semilinear isometrisch zu } \mathcal{T}^{(U)}_{q,k,s}\right\}\right|$$

nur von den Parametern q, k, s abhängig sind.

$\mathbf{q}$	$\mathbf{k}$	$\mathbf{s}$	$\mathbf{t}_{\min}$	$\mathbf{t}_{\max}$	$\mathbf{t}_{\varnothing}$	**Anzahl Isomorphieklassen nach** $\lvert\mathbf{Aut}(\mathfrak{T}^{(\mathbf{U})}_{\mathbf{q,k,s}})\rvert$ **und** $\left\lvert\left\{\mathbf{U}' \mid \mathcal{T}^{(\mathbf{U})}_{\mathbf{q,k,s}} \sim \mathcal{T}^{(\mathbf{U}')}_{\mathbf{q,k,s}}\right\}\right\rvert$
2	4	1	0.040	0.050	0.048	$(30,4)^1$
2	5	2	1.8	3.5	3.0	$(124,5)^4$
2	6	1	2.5	67	7.1	$(126,6)^2$, $(252,3)^1$, $(756,1)^1$
4	3	2	0.69	2.0	1.0	$(84,6)^3$, $(252,2)^1$
4	4	2	25	69	46	$(340,8)^{32}$
4	4	4	2500	4500	3500	$(1360,8)^{40}$

Tabelle 6.1.: Laufzeiten der Sage-Implementierung des Kanonisierers für $\mathcal{T}_{q,k,s}$ in Minuten

Da für alle $x \in X$ einer G-Menge X stets die Bahnenformel $|\operatorname{Stab}_G(x)| \cdot |Gx| = |G|$ erfüllt ist, legt die obige Beobachtung die folgende Vermutung nahe:

6.1.6 Vermutung. Es existiert eine Gruppenoperation einer (unbekannten) Gruppe $G \leq (R^*)^n \rtimes (S_n \times \operatorname{Aut}(R))$ auf der Menge der linearen Codes $\{\mathcal{T}^{(U)}_{q,k,s} \mid U \leq \mathbb{F}_{q^k} \text{ zulässig}\}$, welche die auftretenden Isomorphieklassen und alle Automorphismen vollständig beschreibt.

Wir vermuten weiter, dass sich diese Operation aus einer isomorphen Gruppenoperation einer Gruppe G' auf $\{U \mid U \leq \mathbb{F}_{q^k} \text{ zulässig}\}$ mit Hilfe des Homomorphieprinzips, der Bijektion $U \mapsto \mathcal{T}^{(U)}_{q,k,s}$ und einer Einbettung $G' \hookrightarrow G \leq (R^*)^n \rtimes (S_n \times \operatorname{Aut}(R))$ ergibt.

6.1.7 Bemerkung. Diese Vermutung wurde bereits von Michael Kiermaier aus dem ausgewerteten Datenmaterial zu dem Galois-Ring $\mathbb{Z}_4 = \mathrm{GR}(4,1)$ gezogen. Sie konnte in dieser Arbeit mit der Untersuchung des nächstgrößeren Galois-Rings $\mathrm{GR}(4,2)$ nun schließlich untermauert werden.

6.1.2. Automorphismen von verallgemeinerten Kerdock-Codes

In [49] wurde die Konstruktion der $\mathbb{Z}_4$-linearen Kerdock-Codes auf beliebige Galois-Ringe $R = \mathrm{GR}(4,r)$ der Charakteristik 4 verallgemeinert. Wir wollen nun zunächst diese Verallgemeinerung beschreiben und anschließend die Automorphismengruppe der *verallgemeinerten Kerdock-Codes* untersuchen und mit unserem Programm berechnen. Wir können hiermit die Korrektheit der Ausgabe verifizieren und die Einsatzfähigkeit der Implementierung auf interessanten Probleminstanzen aufzeigen.

Es sei $k \geq 3$ und ungerade. Wie oben betrachtet man wieder eine Ringerweiterung $S = \mathrm{GR}(4,rk)$ von R, definiert $q := 2^r$ und nutzt die R-Modul-Isomorphie $\psi : S_R \to R^k_R$.

Dann definiert die Matrix

$$\Gamma_{q,k+1} := \begin{pmatrix} 1_R & 1_R & 1_R & \dots & 1_R \\ \psi(0_S) & \psi(\xi_S^0) & \psi(\xi_S^1) & \dots & \psi(\xi_S^{q^k-1}) \end{pmatrix}$$

einen freien, linearen Code $\mathcal{K}_{q,k+1}$ der Länge q^k vom Rang $k+1$ über dem Kettenring R. Die von den Spalten von $\Gamma_{q,k+1}$ erzeugte Punktmenge in $\mathrm{PHG}(R_R^{k+1})$ wollen wir die *Kerdock-Punktmenge* zu $\mathcal{K}_{q,k+1}$ nennen. Im Folgenden werden wir nun ausnahmsweise mit T und T^* die Teichmüller-Menge bzw. Teichmüller-Gruppe von S bezeichnen.

Nun zu der angekündigten Untersuchung der Automorphismengruppe der verallgemeinerten Kerdock-Codes $\mathcal{K}_{q,k+1}$. Klar ist, dass

- die Multiplikation mit Elementen $a \in T^*$ der Teichmüller-Gruppe eine R-lineare Abbildung auf $S \simeq R_R^k$ definiert. Diese lässt sich somit über eine Matrix $A \in \mathrm{GL}_k(R)$ darstellen. Wir schließen, dass die Matrix $\left(\begin{smallmatrix} 1 & \\ & A \end{smallmatrix}\right) \in \mathrm{GL}_{k+1}(R)$ einen Automorphismus der Kerdock-Punktmenge definiert und somit auch einen Automorphismus des Kerdock-Codes induziert.

- Genauso definiert ein Ringautomorphismus $\alpha \in \mathrm{Aut}(S)$ eine R-semilineare Abbildung auf $S \simeq R_R^k$. Man zeigt leicht, dass hierdurch ebenfalls ein Automorphismus der Kerdock-Punktmenge induziert wird.

- Die multiplikative Gruppe R^* definiert wegen der Linearität der Codes über die triviale Operation $c \mapsto c \cdot a^{-1}$ für alle $a \in R^*$ stets Automorphismen.

Man rechnet leicht nach, dass alle so gewonnenen Automorphismen (mit Ausnahme der Identität) der Kerdock-Punktmenge verschieden sind. Sie erzeugen eine Untergruppe von $\Gamma\mathrm{L}_{k+1}(R)$, welche zu $(R^* \times T^*) \rtimes \mathrm{Aut}(S)$ isomorph ist.

6.1.8 Folgerung. *Es sei $k \geq 3$ ungerade und $r \in \mathbb{N}$ beliebig, sowie $R := \mathrm{GR}(4, r)$, $S := \mathrm{GR}(4, rk)$ und T^* die Teichmüller-Gruppe von S. Dann besitzt die Automorphismengruppe des R-linearen Kerdock-Codes $\mathcal{K}_{q,k+1}$ eine Untergruppe, welche isomorph zu $(R^* \times T^*) \rtimes \mathrm{Aut}(S)$ ist.*

Für die klassischen $\mathbb{Z}_4$-linearen Kerdock-Codes $\mathcal{K}_{2,k+1}$ ist die Automorphismengruppe bekannt:

6.1.9 Fakt ([36])**.** *Für ungerades $k \geq 5$ hat der $\mathbb{Z}_4$-lineare Kerdock-Code $\mathcal{K}_{2,k+1}$ eine Automorphismengruppe der Ordnung $|\mathrm{Aut}(\mathcal{K}_{2,k+1})| = 2 \cdot 2^k(2^k - 1) \cdot k$. Der Kerdock-Code $\mathcal{K}_{2,3+1}$ bildet eine Ausnahme und hat eine um den Faktor 8 größere Automorphismengruppe[2]. Die Ordnung ist also $2688 = 2 \cdot 8 \cdot 2^3(2^3 - 1) \cdot 3$.*
In beiden Fällen operiert die Automorphismengruppe zweifach transitiv auf der Kerdock-Punktmenge.

[2] In [36] wird fälschlicherweise $|\mathrm{Aut}(\mathcal{K}_{2,3+1})| = 1344$ angegeben. Diese fehlerhafte Angabe tritt auch in weiteren Arbeiten auf.

Das Zustandekommen des weiteren Faktors 2^k in der Ordnung der Automorphismengruppe der Kerdock-Codes $\mathcal{K}_{2,k+1}$ wird in [36] beschrieben.

Wir wollen nun für den nächstgrößeren Kettenring $R = \mathrm{GR}(4,2)$ ebenfalls die Automorphismengruppe des verallgemeinerten Kerdock-Codes $\mathcal{K}_{4,k+1}$ berechnen. Aufgrund des exponentiellen Wachstums der Ordnung $|\mathcal{K}_{4,k+1}| = 4^{2(k+1)}$ gelingt uns dies für die ersten beiden Fälle $k = 3$ und $k = 5$:

6.1.10 Hilfssatz. *Die Kerdock-Codes $\mathcal{K}_{4,3+1}$ und $\mathcal{K}_{4,5+1}$ über dem Galois-Ring $\mathrm{GR}(4,2)$ besitzen nur die in Folgerung 6.1.8 beschriebenen Automorphismen.*

Die Kerdock-Punktmengen beider Codes zerfallen unter der Operation der jeweiligen Automorphismengruppe in zwei Bahnen. Die erste besteht jeweils nur aus dem Punkt $(1_R, \psi(0_S))^T$. Die Punkte $(1_R, \psi(\xi_S^i))^T$ der Teichmüller-Gruppe bilden die zweite Bahn.

Beweis. Resultat des Aufrufs unseres Kanonisierers aus Abschnitt 7.1.2. Die Berechnungen wurden in 20 Sekunden bzw. in 18 Stunden abgeschlossen. □

6.1.11 Vermutung. Für alle Kerdock-Codes $\mathcal{K}_{2^r,k+1}$, $r > 1$ und $k \geq 3$ ungerade, ist die Automorphismengruppe $\mathrm{Aut}(\mathcal{K}_{2^r,k+1})$ isomorph zu $(R^* \times T^*) \rtimes \mathrm{Aut}(S)$.

Durch Punktieren an der Spalte $(1_R, \psi(0_S))^T$ definieren wir zu $\mathcal{K}_{q,k+1}$ den *punktierten Kerdock-Code* $\dot{\mathcal{K}}_{q,k+1}$. Bei den klassischen $\mathbb{Z}_4$-linearen Kerdock-Codes ist diese besondere Auszeichnung einer Spalte – wegen der transitiven Operation der Automorphismengruppe auf der Kerdock-Punktmenge – willkürlich. In den Beispielen $\mathcal{K}_{4,3+1}$ und $\mathcal{K}_{4,5+1}$ ist dies aufgrund unserer Beobachtungen zur Automorphismengruppe aber nicht mehr der Fall. In der Tat ergeben sich hier zwei Isomorphieklassen von Codes mit unterschiedlichen Eigenschaften:

- Das Gray-Bild des punktierten Kerdock-Codes $\dot{\mathcal{K}}_{4,3+1}$ definiert einen nichtlinearen Code der Länge 252 über $\mathbb{F}_4$ der Kardinalität 4^8 und Minimaldistanz 177. Der Vergleich mit [31] zeigt, dass $\mathcal{K}_{4,3+1}$ ein BTKL-Code ist.
- Das Punktieren an einer beliebigen anderen Spalte führt dahingegen zu einem Code mit Minimaldistanz 176 und sonst identischen Parametern, siehe [45, Bemerkung 2.4.3 (e)].

Hier spielt also die Auswahl der punktierten Spalte eine gewichtige Rolle. Nur bei der Punktierung an der Spalte $(1_R, 0_S)^T$ ergibt sich auch ein interessanter BTKL-Code über $\mathrm{GR}(4,2)$.

6.2. Klassifikationsprobleme

Dieses Kapitel beinhaltet eine Sammlung von Ergebnissen, welche mit Hilfe einer C++ Implementierung des Kanonisierers, siehe Abschnitt 7.2, im Spezialfall $R = \mathbb{F}_q$, $R = \mathbb{Z}_4$ und $R = \mathbb{F}_2[X]/(X^2)$ erzielt werden konnten. Diese Ergebnisse fanden auch bereits Eingang in die Arbeiten [21, 25, 26].

6.2.1. Lineare Codes über endlichen Körpern

Dieser Abschnitt bildet eine Zusammenfassung der Resultate aus dem Artikel [21]. Er behandelt die vollständige Klassifikation aller linearen Codes mit fest gewählten Parametern bis auf semilineare Isometrie.

6.2.1 Definition. Es sei $C \leq \mathbb{F}_q^n$ ein linearer Code der Länge n und Dimension k. Für eine Koordinate $i \in [n]$ definiert man den

- in *i punktierten Code* $P_i(C) := \{(c_0, \ldots, c_{i-1}, c_{i+1}, \ldots c_{n-1}) \mid c \in C\}$ durch Streichen der i-ten Koordinaten, und den
- in *i verkürzten Code* $S_i(C) := P_i(S_i'(C))$, indem man sich zunächst auf den linearen Teilcode $S_i'(C) := \{c \in C \mid c_i = 0\}$ zurückzieht und anschließend die i-te Koordinate punktiert.

Weiter ist zu einem Codewort $c \in C$ der *residuelle Code* $\mathrm{Res}(C, c)$ definiert durch das Punktieren des Codes C auf der Trägermenge[3] $\mathrm{supp}(c)$ von c.

6.2.2 Fakt ([39], Corollary 2.7.2)**.** *Es sei $C \leq \mathbb{F}_q$ ein linearer Code der Dimension $k \in [n+1]$, mit Minimaldistanz d und dualer Minimaldistanz $d^\perp$, kurz: ein $[n, k, d]_q^{d^\perp}$-Code. Für jedes beliebige $c \in C$ mit $\mathrm{w_H}(c) = d$ ist $\mathrm{Res}(C, c)$ ein $\left[n - d, k - 1, \geq \left\lceil \frac{d}{q} \right\rceil\right]_q^{\geq d^\perp}$-Code.*

6.2.3 Fakt (Konstruktion Y_1)**.** *Wendet man die obige Konstruktion auf ein Codewort $c \in C^\perp$ mit $\mathrm{w_H}(c) = d^\perp$ an und dualisiert anschließend wieder, so nennt man dieses Verfahren* Konstruktion Y_1. *Der Code $\left(\mathrm{Res}(C^\perp, c)\right)^\perp$ hat dann die Parameter*

$$\left[n - d^\perp, k - d^\perp + 1, \geq d\right]_q^{\geq \left\lceil \frac{d^\perp}{q} \right\rceil}.$$

Das Ziel ist es nun, für festgelegte Parameter n, k, q und Minimaldistanzen d sowie $d^\perp$ alle linearen $[n, k, \geq d]_q^{d^\perp}$-Codes bis auf semilineare Isometrie zu klassifizieren. Diese können wir eindeutig durch eine Transversale $T(n, k, d, d^\perp, q) \subseteq \mathbb{F}_q^{(n-k)\times n}$ von Kontrollmatrizen beschreiben. Das Verfahren beruht nun auf einem Vorgehen, welches versucht, über eine Invertierung der Konstruktion Y_1 einen iterativen Algorithmus zu entwickeln. Ausgangspunkt bildet also eine Transversale

$$S := \bigcup_{d' \geq \left\lceil \frac{d^\perp}{q} \right\rceil} T(n - d^\perp, k - d^\perp + 1, d, d', q)$$

zu allen Codes mit den möglichen Parametern, welche wir aus Konstruktion Y_1 gewinnen könnten. Anschließend werden sukzessiv weitere $d^\perp$ Spalten an die Kontrollmatrix angefügt.

[3] Man streicht alle Koordinaten $i \in \mathrm{supp}(c)$ simultan.

	q = 2	**q = 3**			**q = 4**						**q = 5**			**q = 7**		**q = 8**
n	35	22	24	28	19	21	22	27	30	39	16	16	17	15	26	30
k	10	8	14	21	8	14	16	17	21	27	5	6	8	8	20	23
d	13	10	7	5	9	6	5	8	7	9	10	9	8	7	6	7

Tabelle 6.2.: Parameter, für welche keine linearen Codes existieren

Das Anfügen von Spalten kann wieder mit einem Suchbaum modelliert werden. Die Kinder eines Knotens $\Delta \in T(n-(d^\perp-i), k-(d^\perp-i), d, i, q)$ auf Tiefe $i < d^\perp$ werden von allen zulässigen Erweiterungen von Δ um eine weitere Spalte gebildet. Eine Erweiterung ist zulässig, falls die Bedingungen an die Minimaldistanzen eingehalten werden. Weiter können wir auch vorschreiben, dass die Spalten aufsteigend sortiert in der Kontrollmatrix vorliegen. Der Suchbaum wird nun im Rahmen einer Tiefensuche durchlaufen.

In der Arbeit [21] wurde gezeigt, dass es genügt, sich in dieser Suche auf jeder Tiefe $i \leq d^\perp$ auf eine Transversale $T(n-(d^\perp-i), k-(d^\perp-i), d, i, q)$ zurückzuziehen[4]. Um dies zu gewährleisten, wird nun der Kanonisierer für lineare Codes benötigt: Zu jeder neu hinzugenommenen Spalte wird zunächst der kanonischer Repräsentant der resultierenden Kontrollmatrix berechnet. Trat dieser bislang noch nicht auf, so muss das Anfügen von Spalten fortgeführt werden. Andernfalls kann man die Suche an dieser Stelle abbrechen und mit dem nächsten Kandidaten fortfahren.

Ergebnisse Zu 16 verschiedenen Parametertupeln $[n, k, d]_q$, siehe Tabelle 6.2, mit $q = 2, 3, 4, 5, 7, 8$ konnte die Existenz eines linearen $[n, k, d]_q$-Codes ausgeschlossen werden. Über die in Definition 6.2.1 aufgezeigten Standardkonstruktionen lassen sich damit sogar 217 Verbesserungen der oberen Schranken $\mathrm{ub}(n, k, q)$ an die Minimaldistanz d eines linearen $[n, k, d]_q$-Codes gewinnen. Dabei haben wir uns mit den verfügbaren Tabellen [31] und [65] über die Minimaldistanz eines optimalen linearen Codes verglichen. Leider sind die Daten in beiden Tabellen nicht vollständig. Sofern bekannt wurden weitere Quellen, etwa [2], ebenfalls berücksichtigt. In 109 Fällen konnte anhand dieser Verbesserungen sogar die Lücke zwischen oberer und unterer Schranke geschlossen werden und damit die Minimaldistanz eines optimalen linearen Codes exakt bestimmt werden.

Die erzielten Resultate sind im Unterverzeichnis `results` auf der beigefügten CD abrufbar. Die Datei mit dem Namen `ub<q>.html` für $q = 2, 3, 4, 5, 7, 8, 9$ enthält die oberen Schranken an die Minimaldistanz eines linearen $[n, k]_q$-Codes. Die Verbesserungen gegenüber der Tabelle [31] sind farbig hervorgehoben. Führt man den Mauszeiger über einen farbig hinterlegten Tabelleneintrag, so wird ein Beweis oder die Referenz angegeben.

[4] Dieses Vorgehen ist allgemein als *isomorph rejection* bekannt.

n	**n − k = 6**	**n − k = 7**		
10	$1^0 \ldots 3^0\ 4^2$			
11	$1^0 \ldots 4^0\ 5^1$			1^2
12	$1^0 \ldots 5^0\ 6^1$		1^1	2^{51}
13		1^1	2^6	3^{1219}
14		2^2	3^{30}	4^{7431}
15		3^7	4^{88}	5^{3797}
16		4^{13}	5^{64}	6^{261}
17		5^9	6^{17}	7^4
18		6^5	7^1	8^0
19		7^1	8^0	9^0
20		8^1	9^0	10^0
21		9^0	10^0	11^0

Tabelle 6.3.: Anzahl nicht isomorpher $[n, k, d]_4^{d^\perp}$-Codes für $d \geq 6$ mit Unterscheidung nach $d^\perp$

Wir wollen nun exemplarisch das Beispiel eines $[21, 14, 6]_4$-Codes behandeln. Die Existenz eines solchen Codes konnte über die folgenden Berechnungen ausgeschlossen werden: Zunächst beobachtet man, dass die duale Distanz $d^\perp$ eines derartigen linearen Codes nur die Werte 9, 10 oder 11 annehmen kann. Die obere Schranke liest man direkt aus [31] ab, die untere gewinnt man über eine Argumentation mit dem residuellen Code.

Die Tabelle 6.3 gibt nun die Mächtigkeiten derjenigen Transversalen $T(n, k, d, d^\perp, q)$, welche im Zuge dieser Klassifikation bestimmt wurden. Der Eintrag $d^{\perp^x}$ sagt hierbei aus, dass es x Isomorphieklassen von $[n, k, \geq d]_q^{d^\perp}$-Codes gibt. In der Spalte $n - k = 6$ findet man die Mächtigkeiten der Ausgangspunkte S für das iterative Vorgehen. Die Spalte $n - k = 7$ beschreibt die Anzahlen der zu betrachtenden, nicht isomorphen Codes für jeden Zwischenschritt.

Die Einträge mit $d^\perp = 1$ werden also aus den diagonal darüberliegenden Einträgen der Spalte $n - k = 6$ berechnet. Alle weiteren Einträge ergeben sich aus den Einträgen, welche direkt oberhalb notiert sind. Insgesamt konnten das Ergebnis in einer Gesamtzeit von 2.5 Minuten bestimmt werden.

6.2.2. Nichtexistenz eines extremalen, selbstdualen Codes der Länge 72 mit vorgeschriebenen Automorphismen

Es sei $C \leq \mathbb{F}_2^n$ ein selbstdualer, linearer Code. Dann haben alle Codewörter gerades Hamming-Gewicht. Sind alle auftretenden Hamming-Gewichte sogar durch 4 teilbar, so nennen wir C *doppelt-gerade*. Einen doppelt-geraden, selbstdualen Code nennt man

auch vom *Typ II*. Es ist wohlbekannt [39], dass die Minimaldistanz eines linearen Codes vom Typ II durch $4 + 4\left\lfloor \frac{n}{24} \right\rfloor$ nach oben beschränkt ist. Codes vom Typ II, welche diese Schranke erreichen, heißen *extremal*. Die Frage nach der Existenz eines extremalen Codes der Länge 72 ist ein ungelöstes, sehr bekanntes Problem der Codierungstheorie [39, Research Problem 9.3.6], [42, Research Problem 7.53] oder [46].

In mehreren Arbeiten wurde versucht, einen extremalen, selbstdualen Code der Länge 72 über das Vorschreiben von Automorphismen zu konstruieren. Alle Versuche schlugen bislang fehl. Dies führt im Umkehrschluss zu der folgenden Aussage über die Automorphismengruppe eines derartigen Codes:

6.2.4 Fakt ([5])**.** *Die Automorphismengruppe eines extremalen Codes der Länge* 72 *hat höchstens die Ordnung* 5*.*

Zu dieser Aussage liefert die Arbeit [25] den folgenden Beitrag: Es wurde gezeigt, dass die Gruppentypen $\mathbb{Z}_3 \times \mathbb{Z}_3$, $\mathbb{Z}_7$, D_{10} nicht als Untergruppen der Automorphismengruppe eines extremalen, selbstdualen Codes der Länge 72 auftreten können.

Wir wollen die notwendigen Berechnungen des Computerbeweises kurz skizzieren. Für eine genaue Beschreibung verweisen wir wieder auf den Originalartikel [25]. Wir wollen an dieser Stelle vor allem den Einsatz des Kanonisierers und der Klassifikationsmethoden aus dem Abschnitt 6.2.1 aufzeigen. Diese gehen bei den Fällen $\mathbb{Z}_7$ und D_{10} ein.

Automorphismen der Ordnung 7 Ein Automorphismus $\pi \in S_{72}$ der Ordnung 7 eines extremalen, selbstdualen $[72, 36, 16]_2$-Codes besteht aus zehn 7-Zyklen und zwei Fixpunkten [13]. Verkürzt man den Code an den beiden Fixpunkten, so erhält man einen linearen $[70, 34, \geq 16]$-Code D. Mit Mitteln der Darstellungstheorie zeigt man, dass sich D bis auf Isomorphie als eine direkte Summe

$$D \simeq \varphi_0(D_0) \perp \varphi_1(C_1) \oplus \varphi_2(C_1^\perp)$$

beschreiben lässt. Dabei ist $D_0 \leq D_0^\perp \leq \mathbb{F}_2^{10}$ ein selbstorthogonaler, binärer linearer Code und C_1 ein linearer $[10, k, \geq 4]_8^{\geq 4}$-Code sowie $\varphi_0 : \mathbb{F}_2^{10} \to \mathbb{F}_2^{70}$ und $\varphi_1, \varphi_2 : \mathbb{F}_8^{10} \to \mathbb{F}_2^{70}$ geeignete Isomorphismen von $\mathbb{F}_2$-Vektorräumen.

Um nun einen extremalen, selbstdualen $[72, 36, 16]_2$-Code mit einem Automorphismus $\pi \in S_{72}$ der Ordnung 7 zu gewinnen, wurde nun versucht, den Teilcode D über die obige Zerlegung zu konstruieren. Hierzu beobachtet man zunächst, dass die Abbildungen φ_1 und φ_2 jeweils das Hamming-Gewicht eines beliebigen Vektors vervierfachen. Da die Teilcodes $\varphi_1(C_1)$ und $\varphi_2(C_1^\perp)$ mindestens die Minimaldistanz 16 aufweisen müssen, schließen wir hieraus, dass die Codes C_1 und $C_1^\perp$ mindestens die Minimaldistanz 4 haben.

Mit dem in Abschnitt 6.2.1 beschriebenem Verfahren werden nun zunächst alle zulässigen Codes C_1 bis auf Isomorphie klassifiziert. Dabei genügt es wegen der Isomorphie von $\varphi_1(C_1) \oplus \varphi_2(C_1^\perp)$ und $\varphi_1(C_1^\perp) \oplus \varphi_2(C_1)$, nur jeweils eines der Paare $(C_1, C_1^\perp)$ und $(C_1^\perp, C_1)$ zu untersuchen. In Tabelle 6.4 finden sich die Mächtigkeiten der Transversalen $T(10, k, d, d^\perp, 8)$. Eine erhebliche Einschränkung ergibt sich, wenn man zusätzlich

k	d	$d^\perp$	Anzahl der nicht isomorphen Kandidaten für C_1	für C_1 mit $d(\varphi_1(C_1) \oplus \varphi_2(C_1^\perp)) \geq 16$
3	8	4	1	1
4	4	4	81 717	657
4	5	4	1 854 753	8 657
4	6	4	490 382	2 632
5	4	4	61 487 808	145 918
5	5	4	3 742 898	10 769
5	5	5	3 014 997	9 216
Summe			70 672 556	177 850

Tabelle 6.4.: Resultate im Fall $\mathbb{Z}_7$

berücksichtigt, dass auch die Minimaldistanz des Teilcodes $\varphi_1(C_1) \oplus \varphi_2(C_1^\perp)$ mindestens gleich 16 sein muss. Anschließend ergänzt man die verbliebenen Fälle um den Code $\varphi_0(D_0)$, für welchen wiederum 945 Möglichkeiten zur Auswahl stehen. Man stellt schließlich fest, dass man aus diesen Summen keinen linearen Code D erhält, dessen Minimaldistanz größer oder gleich 16 ist.

6.2.5 Folgerung. *Ein extremaler, selbstdualer Code vom Typ II der Länge* 72 *besitzt keinen Automorphismus der Ordnung* 7*.*

Die Diedergruppe D_{10} Zum Ausschluss der Diedergruppe D_{10} kann man wie oben zunächst den linearen Code C bis auf Isomorphie als eine direkte Summe $C \cong C_0 \oplus \Pi^{-1}(\Psi(X))$ für ein $C_0 \in \mathcal{C}_0$ und $X \in \tilde{\mathcal{X}}$ ausdrücken. Dabei wollen wir nicht näher auf die Abbildungen Π^{-1}, Ψ und die Menge $\mathcal{C}_0$ eingehen.

Die Menge $\tilde{\mathcal{X}}$ stellt eine Teilmenge aller $\mathbb{F}_4$-linearen Codes $X \leq \mathbb{F}_{16}^7$ dar, welche unter dem spur-hermiteschen[5] Skalarprodukt selbstdual sind und weitere Bedingungen an das in der Arbeit [25] eingeführte 5-Gewicht erfüllen. Dieses Gewicht induziert eine Isometriegruppe, welche zu $D_{10}^7 \rtimes S_7$ isomorph ist.

Zur Konstruktion des Codes C bis auf Isomorphie genügt es dann wieder, eine Transversale $\mathcal{X} \subseteq \tilde{\mathcal{X}}$ unter der Gruppenoperation von $D_{10}^7 \rtimes S_7$ zu berechnen. Durch eine Anpassung der inneren Kanonisierung an die geänderte Gruppenstruktur konnte auch hier ein Kanonisierer für diese Gruppenoperation auf der Menge aller linearen Codes realisiert werden. Mit dessen Hilfe und einem iterativen Aufbau der Generatormatrizen ließ sich schließlich eine Transversale $\mathcal{X}$ berechnen. Berücksichtigt man bei den Zwischenschritten auch, dass eine Ergänzung mit einem geeigneten Code $C_0 \in \mathcal{C}_0$ immer

[5] $\langle x, y \rangle := \sum_{i=0}^{6} \mathrm{trace}_{\mathbb{F}_{16}/\mathbb{F}_4}(x_i y_i^4)$.

noch möglich sein muss, so bleiben am Ende nur 4 Kandidaten für die Codes $X \in \mathcal{X}$. Keiner lässt sich zu einem selbstdualen $[72, 36, 16]_2$-Code ergänzen.

6.2.6 Folgerung. *Die Diedergruppe D_{10} tritt nicht als Automorphismengruppe bzw. Untergruppe der Automorphismengruppe eines extremalen, selbstdualen Codes vom Typ II der Länge 72 auf.*

6.2.3. Lineare Codes über endlichen Kettenringen der Ordnung 4

Im Rahmen der Arbeit [20] haben wir einen Kanonisierer für lineare Codes über einem Kettenring R der Ordnung 4 und Kettenlänge 2 implementiert, siehe Abschnitt 7.2. Der Ring R ist dann entweder gleich $\mathbb{Z}_4$ oder $\mathbb{F}_2[X]/(X^2)$. Da für beide Ringe sowohl die multiplikativen Gruppen $(R/\operatorname{Rad}(R))^* = \mathbb{F}_2^* = \{1_R\}$ als auch die Automorphismengruppen $\operatorname{Aut}(R) = \{\mathrm{id}_R\}$ trivial sind, konnte in diesem Fall die innere Kanonisierung mit erheblich einfacheren Mitteln behandelt werden. Dennoch musste auch dort bereits mit einem Erzeugendensystem gearbeitet werden. Für Details verweisen wir wieder auf die Originalliteratur. Jedoch sei angemerkt, dass dort im Wesentlichen mit den gleichen Methoden vorgegangen wird, wie sie hier auch ausgenutzt wurden.

Die entworfene C++ Implementierung des Algorithmus für diese Kettenringe wurde anschließend zur Klassifikation aller linearen Codes vom Rang $k \leq 4$ und Länge $n \leq 10$ eingesetzt. Dabei wurde, wie in Abschnitt 6.2.1, eine Obermenge aller nicht isomorphen Generatormatrizen durch sukzessives Verlängern um eine weitere Spalte gewonnen. Jedoch wurde hierbei diese Obermenge weder über die duale Distanz noch durch einen vorgeschriebenen residuellen Code eingeschränkt.

Tabelle 6.5 zeigt einen Auszug[6] ($5 \leq n \leq 8$) aller erzielten Ergebnisse [19]. Die erste Spalte gibt dabei den Umriss λ der Codes an. Ein Eintrag d^x in dieser Tabelle bedeutet wieder, dass es genau x paarweise nicht semilinear isometrische Codes der Länge n und vom Umriss λ gibt, welche eine minimale homogene Distanz d besitzen.

In Tabelle 6.6 haben wir die Minimaldistanz der Bilder $\iota(C)$ unter der Gray-Abbildung

$$\begin{aligned}\iota : R &\to \mathbb{F}_2^2 \\ 0 \mapsto (0,0), 1 &\mapsto (1,0), \theta \mapsto (1,1), 1+\theta \mapsto (0,1)\end{aligned}$$

der linearen Codes C aus Tabelle 6.5 in Relation zu der Minimaldistanz eines optimalen linearen, binären Code mit gleichen Parametern gesetzt. Dieser Wert ist in Klammern notiert. Für den Kettenring $\mathbb{F}_2[X]/(X^2)$ sind die Bilder unter der Gray-Abbildung stets linear, dort werden wir also keine BTKL-Codes beobachten können.

Sehr erstaunlich ist, dass die Anzahlen der Isomorphieklassen für die beiden nicht isomorphen Ringe nur sehr geringfügig voneinander abweichen. Eine sehr wichtige Ausnahme bildet der bis auf Isomorphie eindeutige Code $C \leq \mathbb{Z}_4^8$ vom Umriss $\lambda = (2,2,2,2)$ mit minimaler homogener Distanz $d(C) = 6$. Ein Parametervergleich beweist, dass dieser Code isomorph zu dem Kerdock-Code $\mathcal{K}_{2,4}$ ist.

[6] Der Auszug ist auch bereits im Rahmen der Arbeit [20, Table 2] erschienen.

$\boldsymbol{\lambda}$	$\mathbf{n=5}$	$\mathbf{n=6}$	$\mathbf{n=7}$	$\mathbf{n=8}$
$(2,2)$	$1^4\ 2^{30}\ 3^5\ 4^{10}$	$1^5\ 2^{58}\ 3^{10}\ 4^{43}\ 5^4\ 6^1$	$1^6\ 2^{100}\ 3^{14}\ 4^{101}\ 5^{23}\ 6^{12}$	$1^7\ 2^{161}\ 3^{18}\ 4^{196}\ 5^{50}\ 6^{72}\ 7^4\ 8^4$
$(2,1)$	$1^1\ 2^{16}\ 3^2\ 4^{13}\ 5^1$	$1^1\ 2^{21}\ 3^2\ 4^{25}\ 5^4\ 6^5$	$1^1\ 2^{27}\ 3^2\ 4^{36}\ 5^6\ 6^{19}\ 7^1\ 8^1$	$1^1\ 2^{33}\ 3^2\ 4^{48}\ 5^7\ 6^{35}\ 7^4\ 8^{12}$
$(1,1)$	$2^1\ 4^2\ 6^1$	$2^1\ 4^2\ 6^2\ 8^1$	$2^1\ 4^2\ 6^2\ 8^2$	$2^1\ 4^2\ 6^2\ 8^3\ 10^1$
$(2,2,2)$	$1^{18}\ 2^{44}\ 3^1$	$1^{49}\ 2^{283}\ 3^{27}\ 4^{22}$	$1^{121}\ 2^{1275}\ 3^{184}\ 4^{370}\ 5^4\ \boxed{6^1}$	$1^{256}\ 2^{4705}\ 3^{699}\ 4^{2973}\ 5^{238}\ 6^{23}$
$(2,2,1)$	$1^{17}\ 2^{96}\ 3^4\ 4^4$	$1^{33}\ 2^{349}\ 3^{32}\ 4^{85}$	$1^{58}\ 2^{985}\ 3^{108}\ 4^{553}\ 5^{19}\ 6^5$	$1^{93}\ 2^{2382}\ 3^{246}\ 4^{2222}\ 5^{242}\ 6^{133}\ 8^1$
$(2,1,1)$	$1^3\ 2^{40}\ 3^2\ 4^9$	$1^4\ 2^{86}\ 3^6\ 4^{52}\ 5^1$	$1^6\ 2^{162}\ 3^{11}\ 4^{160}\ 5^9\ 6^{11}$	$1^7\ 2^{276}\ 3^{16}\ 4^{375}\ 5^{31}\ 6^{76}\ 7^1\ 8^4$
$(1,1,1)$	$2^3\ 4^3$	$2^4\ 4^7\ 6^1$	$2^6\ 4^{11}\ 6^3\ 8^1$	$2^7\ 4^{17}\ 6^7\ 8^3$
$(2,2,2,2)$	$1^9\ 2^5$	$1^{63}\ 2^{115}\ 3^1$	$1^{381}\ 2^{1718}\ 3^{88}\ 4^{28}$	$1^{1955}\ 2^{19292}\ 3^{2340}\ 4^{2302}\ 5^2\ \boxed{6^1}$
$(2,2,2,1)$	$1^{23}\ 2^{32}$	$1^{121}\ 2^{454}\ 3^8\ 4^4$	$1^{499}\ 2^{4125}\ 3^{273}\ 4^{287}$	$1^{1728}\ 2^{27552}\ 3^{2939}\ 4^{7376}\ 5^{35}\ 6^4$
$(2,2,1,1)$	$1^{16}\ 2^{51}$	$1^{54}\ 2^{412}\ 3^{12}\ 4^{22}$	$1^{149}\ 2^{2168}\ 3^{140}\ 4^{439}$	$1^{359}\ 2^{8839}\ 3^{717}\ 4^{4426}\ 5^{64}\ 6^{14}$
$(2,1,1,1)$	$1^3\ 2^{25}\ 4^1$	$1^6\ 2^{104}\ 3^3\ 4^{21}$	$1^{12}\ 2^{321}\ 3^{15}\ 4^{153}\ 5^1\ 6^1$	$1^{21}\ 2^{834}\ 3^{41}\ 4^{735}\ 5^{17}\ 6^{19}\ 8^1$
$(1,1,1,1)$	$2^3\ 4^1$	$2^6\ 4^5$	$2^{12}\ 4^{14}\ 6^1$	$2^{21}\ 4^{37}\ 6^4\ 8^1$

(a) Anzahl Isomorphieklassen für lineare Codes über $\mathbb{Z}_4$

$\boldsymbol{\lambda}$	$\mathbf{n=5}$	$\mathbf{n=6}$	$\mathbf{n=7}$	$\mathbf{n=8}$
$(2,2)$	$1^4\ 2^{30}\ 3^5\ 4^{10}$	$1^5\ 2^{58}\ 3^{10}\ 4^{43}\ 5^4\ 6^1$	$1^6\ 2^{100}\ 3^{14}\ 4^{101}\ 5^{23}\ 6^{12}$	$1^7\ 2^{161}\ 3^{18}\ 4^{196}\ 5^{50}\ 6^{72}\ 7^4\ 8^4$
$(2,1)$	$1^1\ 2^{16}\ 3^2\ 4^{13}\ 5^1$	$1^1\ 2^{21}\ 3^2\ 4^{25}\ 5^4\ 6^5$	$1^1\ 2^{27}\ 3^2\ 4^{36}\ 5^6\ 6^{19}\ 7^1\ 8^1$	$1^1\ 2^{33}\ 3^2\ 4^{48}\ 5^7\ 6^{35}\ 7^4\ 8^{12}$
$(1,1)$	$2^1\ 4^2\ 6^1$	$2^1\ 4^2\ 6^2\ 8^1$	$2^1\ 4^2\ 6^2\ 8^2$	$2^1\ 4^2\ 6^2\ 8^3\ 10^1$
$(2,2,2)$	$1^{18}\ 2^{44}\ 3^1$	$1^{49}\ 2^{283}\ 3^{27}\ 4^{22}$	$1^{121}\ 2^{1275}\ 3^{184}\ 4^{371}\ 5^4$	$1^{256}\ 2^{4705}\ 3^{699}\ 4^{2975}\ 5^{238}\ 6^{21}$
$(2,2,1)$	$1^{17}\ 2^{96}\ 3^4\ 4^4$	$1^{33}\ 2^{349}\ 3^{32}\ 4^{85}$	$1^{58}\ 2^{985}\ 3^{108}\ 4^{553}\ 5^{19}\ 6^5$	$1^{93}\ 2^{2382}\ 3^{246}\ 4^{2222}\ 5^{242}\ 6^{133}\ 8^1$
$(2,1,1)$	$1^3\ 2^{40}\ 3^2\ 4^9$	$1^4\ 2^{86}\ 3^6\ 4^{52}\ 5^1$	$1^6\ 2^{162}\ 3^{11}\ 4^{160}\ 5^9\ 6^{11}$	$1^7\ 2^{276}\ 3^{16}\ 4^{375}\ 5^{31}\ 6^{76}\ 7^1\ 8^4$
$(1,1,1)$	$2^3\ 4^3$	$2^4\ 4^7\ 6^1$	$2^6\ 4^{11}\ 6^3\ 8^1$	$2^7\ 4^{17}\ 6^7\ 8^3$
$(2,2,2,2)$	$1^9\ 2^5$	$1^{63}\ 2^{115}\ 3^1$	$1^{381}\ 2^{1718}\ 3^{89}\ 4^{27}$	$1^{1955}\ 2^{19292}\ 3^{2344}\ 4^{2304}\ 5^1$
$(2,2,2,1)$	$1^{23}\ 2^{32}$	$1^{121}\ 2^{454}\ 3^8\ 4^4$	$1^{499}\ 2^{4125}\ 3^{273}\ 4^{287}$	$1^{1728}\ 2^{27552}\ 3^{2939}\ 4^{7379}\ 5^{34}\ 6^2$
$(2,2,1,1)$	$1^{16}\ 2^{51}$	$1^{54}\ 2^{412}\ 3^{12}\ 4^{22}$	$1^{149}\ 2^{2168}\ 3^{140}\ 4^{439}$	$1^{359}\ 2^{8839}\ 3^{717}\ 4^{4426}\ 5^{64}\ 6^{14}$
$(2,1,1,1)$	$1^3\ 2^{25}\ 4^1$	$1^6\ 2^{104}\ 3^3\ 4^{21}$	$1^{12}\ 2^{321}\ 3^{15}\ 4^{153}\ 5^1\ 6^1$	$1^{21}\ 2^{834}\ 3^{41}\ 4^{735}\ 5^{17}\ 6^{19}\ 8^1$
$(1,1,1,1)$	$2^3\ 4^1$	$2^6\ 4^5$	$2^{12}\ 4^{14}\ 6^1$	$2^{21}\ 4^{37}\ 6^4\ 8^1$

(b) Anzahl Isomorphieklassen für lineare Codes über $\mathbb{F}_2[X]/(X^2)$

Tabelle 6.5.: Klassifikationsergebnisse für Kettenringe der Kardinalität 4

$\lvert\mathbf{C}\rvert$	$\mathbf{n'=6}$	$\mathbf{n'=8}$	$\mathbf{n'=10}$	$\mathbf{n'=12}$	$\mathbf{n'=14}$	$\mathbf{n'=16}$	$\mathbf{n'=18}$	$\mathbf{n'=20}$
2^2	4(4)	5(5)	6(6)	8(8)	9(9)	10(10)	12(12)	13(13)
2^3	2(3)	4(4)	5(5)	6(6)	8(8)	8(8)	10(10)	10(11)
2^4	2(2)	4(4)	4(4)	6(6)	6(7)	8(8)	8(8)	10(10)
2^5	2(2)	2(2)	4(4)	4(4)	6(6)	8(8)	8(8)	8(9)
2^6	1(1)	2(2)	3(3)	4(4)	6(5)	6(6)	8(8)	8(8)
2^7		2(2)	2(2)	4(4)	4(4)	6(6)	6(7)	8(8)
2^8		1(1)	2(2)	3(3)	4(4)	6(5)	6(6)	8(8)

(a) für lineare Codes über $\mathbb{Z}_4$

$\lvert\mathbf{C}\rvert$	$\mathbf{n'=6}$	$\mathbf{n'=8}$	$\mathbf{n'=10}$	$\mathbf{n'=12}$	$\mathbf{n'=14}$	$\mathbf{n'=16}$	$\mathbf{n'=18}$	$\mathbf{n'=20}$
2^2	4(4)	5(5)	6(6)	8(8)	9(9)	10(10)	12(12)	13(13)
2^3	2(3)	4(4)	5(5)	6(6)	8(8)	8(8)	10(10)	10(11)
2^4	2(2)	4(4)	4(4)	6(6)	6(7)	8(8)	8(8)	10(10)
2^5	2(2)	2(2)	4(4)	4(4)	6(6)	8(8)	8(8)	8(9)
2^6	1(1)	2(2)	3(3)	4(4)	5(5)	6(6)	8(8)	8(8)
2^7		2(2)	2(2)	4(4)	4(4)	6(6)	6(7)	8(8)
2^8		1(1)	2(2)	3(3)	4(4)	5(5)	6(6)	8(8)

(b) für lineare Codes über $\mathbb{F}_2[X]/(X^2)$

Tabelle 6.6.: Minimaldistanz der Gray-Bilder der Codes aus Tabelle 6.5

Unter der Gray-Abbildung erhält man als Bild von $\mathcal{K}_{2,4}$ einen nichtlinearen, binären Code (den Nordstrom-Robinson-Code) der doppelten Länge. Da ι eine Isometrie ist, hat dieser Code die Hamming-Minimaldistanz 6. Ein Vergleich mit der oberen Schranke $\mathrm{ub}(16, 8, 2) = 5$ an die Hamming-Minimaldistanz eines linearen $[16, 8]_2$-Codes zeigt, dass der Kerdock-Code $\mathcal{K}_{2,4}$ ein BTL-Code ist. Der zweite, ausgezeichnete BTL-Code in der Tabelle 6.5a mit Umriss $(2, 2, 2)$ der Länge 6 ergibt sich durch die Verkürzung des Kerdock-Codes $\mathcal{K}_{2,4}$.

Es zeigen sich neben den beobachteten BTL-Codes aber auch Parameter, etwa $|C| = 2^4$ und $n' = 14$, bei welchen ein binärer, linearer Code eine bessere Minimaldistanz erreicht als das Gray-Bild eines optimalen $\mathbb{Z}_4$-linearen Codes.

6.2.4. Kryptographie

In vielen kryptographischen Verfahren kommen Boolesche Funktionen $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ als sogenannte S-Boxen zum Einsatz. Für eine Einführung in dieses Themengebiet verweisen wir auf den Übersichtsartikel [10]. Die Güte der erzielten Verschlüsselung hängt hierbei

stark von der Auswahl dieser Funktionen ab. Wie in der Codierungstheorie auch, können wir Äquivalenzklassen von Funktionen bilden, welche sich beim Einsatz als S-Box gleich (gut oder schlecht) verhalten.

In [16] wurden in dieser Hinsicht die am häufigsten genutzten Äquivalenzbegriffe untersucht und der Zusammenhang dieser Begriffe mit der Isomorphie linearer Codes hergestellt. Wir möchten an dieser Stelle diese Resultate wiederholen um aufzuzeigen, dass der von uns entworfene Kanonisierer auch in der Kryptographie Anwendung finden kann.

Zunächst führen wir die am meisten genutzten Äquivalenzbegriffe für Boolesche Funktionen ein. Wir nennen zwei Funktionen $F_0, F_1 : \mathbb{F}_2^n \to \mathbb{F}_2^m$ *CCZ-äquivalent*[7], falls eine invertierbare Blockmatrix $A = \begin{pmatrix} A^{(0,0)} & A^{(0,1)} \\ A^{(1,0)} & A^{(1,1)} \end{pmatrix} \in \mathrm{GL}_{n+m}(\mathbb{F}_2)$ und Vektoren $a \in \mathbb{F}_2^n$, $b \in \mathbb{F}_2^m$ existieren mit

$$\left\{ \begin{pmatrix} A^{(0,0)} & A^{(0,1)} \\ A^{(1,0)} & A^{(1,1)} \end{pmatrix} \cdot \begin{pmatrix} x \\ F_0(x) \end{pmatrix} + \begin{pmatrix} a \\ b \end{pmatrix} \mid x \in \mathbb{F}_2^n \right\} = \left\{ \begin{pmatrix} x \\ F_1(x) \end{pmatrix} \mid x \in \mathbb{F}_2^n \right\}. \tag{6.1}$$

Können wir in Gleichung (6.1) die Matrix A sogar derart wählen, dass die Untermatrix $A^{(0,1)} = \mathbf{0_{n \times m}}$ gleich der Nullmatrix ist, so nennen wir beide Funktionen *EA-äquivalent*[8]. Gibt es eine Matrix A, welche Gleichung (6.1) erfüllt und für die sowohl $A^{(0,1)}$ als auch $A^{(1,0)}$ Nullmatrizen sind, so nennen wir die Booleschen Funktionen F_0 und F_1 *affin äquivalent*.

Indem wir nun den Vektoren eine zusätzliche Koordinate voranstellen und diese gleich 1 setzen, können wir die Menge $\left\{(1, x, F_0(x))^T \mid x \in \mathbb{F}_2^n\right\}$ als Punktmenge in der projektiven Geometrie $\mathrm{PG}(\mathbb{F}_2^{n+m+1})$ auffassen. Die Gleichung

$$\begin{pmatrix} 1 & 0 & 0 \\ a & A^{(0,0)} & A^{(0,1)} \\ b & A^{(1,0)} & A^{(1,1)} \end{pmatrix} \cdot \left\{ \begin{pmatrix} 1 \\ x \\ F_0(x) \end{pmatrix} \mid x \in \mathbb{F}_2^n \right\} = \left\{ \begin{pmatrix} 1 \\ x \\ F_1(x) \end{pmatrix} \right\}$$

ist nun aber eine äquivalente Umformulierung von Gleichung (6.1). CCZ-äquivalente Boolesche Funktionen definieren also isomorphe Punktkonfigurationen der projektiven Geometrie. Fassen wir die Punktmenge als Spaltenmenge einer Kontrollmatrix[9] $\Gamma^{(F_0)}$, $\Gamma^{(F_1)}$ auf, so definieren sie aber auch semilinear isometrische Codes $C^{(F_0)}$, $C^{(F_1)}$.

Indem man nun noch zeigt, dass isomorphe Codes $C^{(F_0)}$, $C^{(F_1)}$ auch CCZ-äquivalente Boolesche Funktionen definieren, erhält man das nachfolgende Theorem:

6.2.7 Fakt ([16], Theorem 9)**.** *Es seien $F_0, F_1 : \mathbb{F}_2^n \to \mathbb{F}_2^m$ Boolesche Funktionen. Dann ist F_0 genau dann CCZ-äquivalent zu F_1, falls die linearen Codes $C^{(F_0)}$ und $C^{(F_1)}$ isomorph sind.*

[7] Nach den Autoren Carlet, Charpin und Zinoviev, welche diese zum ersten Mal definierten.

[8] Die Abkürzung EA steht für *extended affin*.

[9] Wir können die Punkte genauso gut als die Spalten einer Generatormatrix auffassen. Wir folgen hierbei aber der Definition aus der Kryptographie.

Man macht sich leicht klar, dass man mit unserem Ansatz zur Kanonisierung von Generator- bzw. Kontrollmatrizen linearer Codes auch sofort die affine wie auch die EA-Äquivalenz Boolescher Funktionen behandeln kann. Man zieht sich schlicht auf die Untergruppen G^{aff} bzw. G^{EA} von $\mathrm{GL}_{n+m+1}(\mathbb{F}_2)$, d.h. auf Blockmatrizen der Gestalt

$$\begin{pmatrix} 1 & 0 & 0 \\ a & A^{(0,0)} & 0 \\ b & 0 & A^{(1,1)} \end{pmatrix} \qquad \text{bzw.} \qquad \begin{pmatrix} 1 & 0 & 0 \\ a & A^{(0,0)} & 0 \\ b & A^{(1,0)} & A^{(1,1)} \end{pmatrix},$$

zurück. Da wir ausschließlich den binären Fall behandeln, entspricht die innere Kanonisierung zu $G = \mathrm{GL}_{n+m+1}(\mathbb{F}_2)$ den elementaren Zeilenumformungen aus dem Gaußschen Eliminationsverfahren. Dies erklärt nun auch sofort, wie man die innere Kanonisierung für die Untergruppen G^{aff} bzw. G^{EA} effizient implementieren kann.

Da zum Erscheinungszeitpunkt von [16] unser Vorgehen aber in dieser Form noch nicht bekannt war, erreichten die Autoren die Einschränkungen auf die Gruppen G^{aff} bzw. G^{EA} durch das Anfügen weiterer $2^n - 1$ bzw. $2(2^n - 1)$ Spalten an die Kontrollmatrix $\Gamma^{(F)}$, d.h. durch eine Verdopplung bzw. Verdreifachung der Längen der Codes.

Klassifikation selbstdualer Bent-Funktionen in acht Variablen Als Anwendung des Kanonisierers beschreiben wir nun die Klassifikation selbstdualer Bent-Funktionen in acht Variablen, wie sie in der Arbeit [26] erschienen ist. Wir geben wieder eine kurze Zusammenfassung.

Es sei nun $m = 1$, d.h. wir betrachten Boolesche Funktionen $f : \mathbb{F}_2^n \to \mathbb{F}_2$. Eine solche Funktion lässt sich stets eineindeutig als Auswertungsfunktion eines Polynoms

$$p_f = \sum_{i=(i_0,\ldots,i_{n-1})\in[2]^n} p_i x_0^{i_0} \cdots x_{n-1}^{i_{n-1}} \in \mathbb{F}_2[x_0, \ldots, x_{n-1}]$$

in n Unbestimmten beschreiben. Wir definieren den Grad einer Booleschen Funktion f über den Grad des zugehörigen Polynoms p_f.

Zu einer ganzen Zahl $i \in [2^n]$ sei der Vektor $(i_0, \ldots, i_{n-1}) \in \mathbb{F}_2^n$ definiert über die Binärdarstellung $i = \sum_{j=0}^{n-1} i_j 2^j$. Dann wollen wir mit dem Vektor $c_f := (f_0, \ldots, f_{2^n-1}) \in \mathbb{F}_2^{2^n}$ den Wahrheitswerteverlauf der Funktion f beschreiben.

Eine Boolesche Funktion heißt affin, falls der Grad des Polynoms p_f eins ist und linear, falls zusätzlich der konstante Term des Polynoms p_f null ist. Die Nichtlinearität einer Booleschen Funktion f wollen wir nun über den minimalen Hamming-Abstand von c_f zu den Vektoren c_g aller affinen Funktion g messen, d.h. wir definieren die Nichtlinearität von f als $\mathrm{nl}(f) := \min\{d_{\mathrm{H}}(c_f, c_g) \mid g : \mathbb{F}_2^n \to \mathbb{F}_2 \text{ affin}\}$.

Für kryptographische Verfahren ist man nun an möglichst stark nichtlinearen Funktionen interessiert. Man kann zeigen, dass die Nichtlinearität einer Booleschen Funktion f stets durch $2^{n-1} - 2^{n/2-1}$ nach oben beschränkt ist. Diejenigen Funktionen, welche diesen Wert annehmen, nennt man daher auch *krumm* oder *Bent*-Funktionen. Es ist klar, dass Bent-Funktionen nur für gerades n existieren können.

Die Definition einer Bent-Funktion lässt sich auf beliebige Boolesche Funktionen $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$ übertragen, indem man die Nichtlinearität für jede der m Komponentenfunktionen untersucht. Für Bent-Funktionen fallen die Begriffe der EA-Äquivalenz und CCZ-Äquivalenz stets zusammen, siehe [10].

Wie in der Codierungstheorie auch, lässt sich auf der Menge aller Booleschen Funktionen $f : \mathbb{F}_2^n \to \mathbb{F}_2$ ein Dualitätsbegriff[10] definieren. Jedoch wird die Selbstdualität einer Booleschen Funktion im Allgemeinen nicht unter EA-Äquivalenz erhalten, d.h. in der Äquivalenzklasse einer selbstdualen Booleschen Funktion gibt es auch nicht selbstduale Vertreter.

Um nun wieder eine Gruppenoperation auf der Menge der selbstdualen Booleschen Funktionen $f : \mathbb{F}_2^n \to \mathbb{F}_2$ zur Verfügung zu haben, schränken wir uns auf die Operation der Untergruppe

$$\overline{\mathcal{O}}_n := \left\{ \begin{pmatrix} 1 & 0 & 0 \\ Lb^T & L & 0 \\ c & b & 1 \end{pmatrix} \mid \begin{array}{l} L \in \mathrm{GL}_n(\mathbb{F}_2) : L^T L = I_n, c \in \mathbb{F}_2, \\ b \in \mathbb{F}_2^n : \mathrm{w_H}(b) \equiv 0 \mod 2 \end{array} \right\}$$

ein. Wir sagen zwei Boolesche Funktionen $f, g : \mathbb{F}_2^n \to \mathbb{F}_2$ sind *EO-äquivalent*[11], falls die Kontrollmatrizen $\Gamma^{(f)}$, $\Gamma^{(g)}$ unter der Gruppenoperation mit $\overline{\mathcal{O}}_n \times S_{2^n}$ isomorph sind.

Die Multiplikation eines Vektor $x \in \mathbb{F}_2^n$ mit $L \in \mathcal{O}_n$ und auch die Addition mit einem Vektor $b \in \mathbb{F}_2^n$ von geradem Gewicht erhält die Parität des Hamming-Gewichts von x. Wir können also die Spalten $(1, x, F(x)), x \in \mathbb{F}_2^n$ nach geradem und ungeradem Hamming-Gewicht von x mit dem Homomorphieprinzip partitionieren, d.h. uns auf die Gruppenoperation von $\overline{\mathcal{O}}_n \times S_{2^{n-1}} \times S_{2^{n-1}}$ auf entsprechend sortierten Generatormatrizen zurückziehen.

Für die Gruppe $\overline{\mathcal{O}}_n$ zeigt sich nun das Problem, dass wir für die innere Minimierung nicht stets einen effizienten Kanonisierer für alle Untergruppen zur Verfügung stellen können. Dies liegt daran, dass sich die Stabilisatoren, welche in der inneren Minimierung auftreten, wesentlich schwerer beschreiben lassen.

Das Problem umgehen wir, indem wir auf der Matrixkomponente wieder zur Definition von EA-Äquivalenz zurückkehren, d.h. zur Operation von $G^{\mathrm{EA}} \times S_{2^{n-1}} \times S_{2^{n-1}}$. Führen nun zwei selbstduale Bent-Funktionen zu isomorphen Codes unter dieser Operation, so lässt sich über die Transporterelemente und die Automorphismengruppe sehr leicht feststellen, ob die zugehörigen linearen Codes unter der Operation von $\overline{\mathcal{O}}_n \times S_{2^{n-1}} \times S_{2^{n-1}}$ ebenfalls isomorph sind, siehe [26, Lemma 1].

Über die Lösung von mehreren diophantischen Gleichungssystemen [26] wurde in dieser Arbeit nun eine Obermenge einer Transversalen der selbstdualen Bent-Funktionen vom Grad 3 in acht Variablen unter EO-Äquivalenz bestimmt. Sie umfasst 1 912 496

[10] Eine Involution $f \mapsto \tilde{f}$ auf der Menge aller Booleschen Funktionen. Sie wird aus der sogenannten Walsh–Hadamard Transformation gewonnen.

[11] Die Abkürzung EO steht für *extended orthogonal*, da $\mathcal{O}_n := \{L \in \mathrm{GL}_n(\mathbb{F}_2) \mid L^T L = I_n\}$ auch als orthogonale Gruppe bezeichnet wird.

Funktionen. Mit dem oben beschriebenen Vorgehen konnte hieraus eine Transversale der EO-Äquivalenzklassen berechnet werden:

6.2.8 Fakt ([26], Theorem 5)**.** *Es existieren 1 162 420 992 selbstduale, kubische Bent-Funktionen in acht Variablen. Diese bilden 45 EO-Äquivalenzklassen.*

6.3. Network- und $\mathbb{F}_q$-lineare $\mathbb{F}_{q^r}$-Codes

In diesem Abschnitt wollen wir nun noch kurz aufzeigen, dass das von uns beschriebene Verfahren zur Kanonisierung linearer Codes bereits alle notwendigen Ideen beinhaltet, um auch Network-Codes über $\mathbb{F}_q$ und $\mathbb{F}_q$-lineare Codes über dem Alphabet $\mathbb{F}_{q^r}$ zu behandeln. Eine detaillierte Beschreibung der Kanonisierung dieser Strukturen wird in [18] gegeben.

Wir wollen nun zunächst beweisen, dass sich die Äquivalenzbegriffe in den beiden Strukturen auf die gleiche Gruppenoperation zurückführen lassen[12]. Für diese werden wir anschließend die grundlegenden Ideen zur Definition eines Kanonisierers beschreiben.

6.3.1. Network-Codes

Die Menge $\mathcal{P}_q(k) := \{\mathcal{U} \mid \mathcal{U} \leq \mathbb{F}_q^k\}$ aller Unterräume von $\mathbb{F}_q^k$ wird über die Definition der sogenannten Unterraum-Distanz

$$d_S(\mathcal{U}, \mathcal{V}) := \dim(\mathcal{U}) + \dim(\mathcal{V}) - 2\dim(\mathcal{U} \cap \mathcal{V})$$

zu einem metrischen Raum[13]. Die Vektoren in $\mathbb{F}_q^k$ werden wir ausnahmsweise an dieser Stelle als Spaltenvektoren auffassen. Diese Sichtweise wird sich später als nützlich erweisen. Als *Network-Code* bezeichnen wir nun jede beliebige Teilmenge $C := \{\mathcal{U}_0, \ldots, \mathcal{U}_{n-1}\} \subseteq \mathcal{P}_q(k)$ von $\mathcal{P}_q(k)$.

Die Operation der Gruppe $\Gamma\mathrm{L}_k(\mathbb{F}_q)$ von links auf der Menge $\mathcal{P}_q(k)$ erhält, wie man sich leicht überlegt, die Unterraum-Distanz. Aus dem Hauptsatz der projektiven Geometrie folgt andererseits, dass für $k \geq 3$ die Isometriegruppe (bezüglich der Unterraum-Distanz) von $\mathcal{P}_q(k)$ durch $\mathrm{P}\Gamma\mathrm{L}_k(\mathbb{F}_q)$ gegeben ist. Die Äquivalenzklassen von Network-Codes definiert man folglich über die Bahnen der Operation mit $\Gamma\mathrm{L}_k(\mathbb{F}_q)$.

Zur Vereinfachung der nachfolgenden Formulierungen gehen wir nun davon aus, dass alle Unterräume $\mathcal{U} \in C$ des zu kanonisierenden Network-Codes C eine maximale Dimension $\dim(\mathcal{U}) < r$, für ein $0 < r < k$, besitzen. Dann lässt sich jedes Element $\mathcal{U} \in C$ über eine Matrix $U \in \mathbb{F}_q^{k \times r}$, deren Spalten ein Erzeugendensystem von $\mathcal{U}$ bilden, beschreiben. Wir schreiben hierfür wieder kurz $\langle U \rangle = \mathcal{U}$.

[12] Wir danken Y. Edel, Ghent University, an dieser Stelle für den Hinweis.

[13] Wir können diese Distanz auch über den graphentheoretischen Abstand der Knoten $\mathcal{U}$, $\mathcal{V}$ im Hasse-Diagramm des Unterraumverbands von $\mathbb{F}_q^k$ erklären.

Wir wollen nun die Äquivalenz von Network-Codes über eine Gruppenoperation auf der Menge der Vektoren (bzw. Blockmatrizen) $(U_0, \ldots, U_{n-1}) \in \left(\mathbb{F}_q^{k\times r}\right)^n$ ausdrücken. Wir gehen hierzu analog zu Folgerung 3.1.10 vor. Es gilt dann der folgende Satz:

6.3.1 Satz. *Zwei Network-Codes*

$$C := \{\langle U_i\rangle \mid U_i \in \mathbb{F}_q^{k\times r}, i \in [n]\} \text{ und } C' := \{\langle V_i\rangle \mid V_i \in \mathbb{F}_q^{k\times r}, i \in [n]\}$$

sind genau dann äquivalent, falls es ein Gruppenelement

$$(A, (B_0, \ldots, B_{n-1}); \alpha, \pi) \in (\mathrm{GL}_k(\mathbb{F}_q) \times (\mathrm{GL}_r(\mathbb{F}_q))^n) \rtimes (\mathrm{Aut}(\mathbb{F}_q) \times S_n)$$

gibt mit $A\alpha(U_{\pi(i)})B^{-1} = V_i$ *für alle* $i \in [n]$.

Beweis. Trivial. □

6.3.2. $\mathbb{F}_q$-lineare $\mathbb{F}_{q^r}$-Codes

Es sei auch hier weiter $r \geq 1$. Wir nennen einen $\mathbb{F}_q$-lineare Teilmenge von $\mathbb{F}_{q^r}^n$ einen *$\mathbb{F}_q$-linearen $\mathbb{F}_{q^r}$-Code*[14]. Bekannte Eigenschaften linearer Codes werden etwa in [38] auf $\mathbb{F}_q$-lineare $\mathbb{F}_{q^r}$-Codes übertragen. Wie für die klassischen linearen Codes definieren wir nun wieder den Äquivalenzbegriff $\mathbb{F}_q$-linearer $\mathbb{F}_{q^r}$-Codes über diejenige Untergruppe der Hamming-Isometrien von $\mathbb{F}_{q^r}^n$, welche $\mathbb{F}_q$-lineare $\mathbb{F}_{q^r}$-Codes auf $\mathbb{F}_q$-lineare $\mathbb{F}_{q^r}$-Codes abbilden. Um diese Gruppe besser beschreiben zu können, definieren wir zunächst für $\mathbb{F}_q^r$ die Distanzfunktion

$$d_{\mathrm{H},r}(u,v) := \begin{cases} 0, & \text{falls } u = v \\ 1, & \text{sonst} \end{cases} \text{ für alle } u, v \in \mathbb{F}_q^r$$

und setzen diese durch Summenbildung auf $\mathbb{F}_q^{rn}$ fort.

Eine $\mathbb{F}_q$-lineare Abbildung $T : \mathbb{F}_{q^r} \to \mathbb{F}_q^r$ definiert somit auch eine Isometrie der metrischen Räume $(\mathbb{F}_{q^r}, d_{\mathrm{H}})$ und $(\mathbb{F}_q^r, d_{\mathrm{H},r})$. Wir geben uns nun T fest vor und setzen T komponentenweise zu einer $\mathbb{F}_q$-linearen Bijektion $\mathbb{F}_{q^r}^n \to \mathbb{F}_q^{rn}$ fort. Hiermit haben wir die Möglichkeit, uns den $\mathbb{F}_q$-linearen $\mathbb{F}_{q^r}$-Code C als Untervektorraum (linearen Code) $T(C)$ der $\mathbb{F}_q$-Dimension k des metrischen Raums $\left(\mathbb{F}_q^{rn}, d_{\mathrm{H},r}\right)$ vorzustellen. Den Äquivalenzbegriff untersuchen wir nun unter diesen Gesichtspunkt:

- Eine beliebige Permutation $\pi \in S_n$ der Koordinatenblöcke der Länge r definiert eine $\mathbb{F}_q$-lineare $d_{\mathrm{H},r}$-Isometrie auf $\mathbb{F}_q^{rn}$. Wir nennen diese $\iota^{(\pi)}$.
- Die Operation mit einer Blockdiagonalmatrix $B = \mathrm{diag}(B_0, \ldots, B_{n-1})$, für $B_i \in \mathrm{GL}_r(\mathbb{F}_q), i \in [n]$, auf $\mathbb{F}_q^{rn}$ definiert ebenso eine $\mathbb{F}_q$-lineare $d_{\mathrm{H},r}$-Isometrie.

[14] Insbesondere – aber nicht ausschließlich – falls q prim ist, wird in der Literatur auch der Name *additiver Code* verwendet.

- Für jeden Körperautomorphismus $\alpha \in \mathrm{Aut}(\mathbb{F}_q)$ ist die komponentenweise Anwendung von α auf $\mathbb{F}_q^{rn}$ eine $\mathbb{F}_q$-semilineare Abbildung und bildet daher Untervektorräume auf Untervektorräume ab. Außerdem respektiert sie ganz offensichtlich die Metrik $d_{\mathrm{H},r}$.

Umgekehrt überlegt man sich leicht, dass es für $rn \geq 3$ keine weiteren $d_{\mathrm{H},r}$-Isometrien von $\mathbb{F}_q^{rn}$ von $\mathbb{F}_q^{rn}$ gibt, welche Untervektorräume auf Untervektorräume abbilden:

6.3.2 Hilfssatz. *Ist $rn \geq 3$ und ι eine $d_{\mathrm{H},r}$-Isometrie von $\mathbb{F}_q^{rn}$, welche Untervektorräume auf Untervektorräume abbildet, so lässt sich ι bereits über ein Element der Gruppe $(\mathrm{GL}_r(\mathbb{F}_q))^n \rtimes (\mathrm{Aut}(\mathbb{F}_q) \times S_n)$ eindeutig beschreiben.*

Beweis. Da $rn \geq 3$ vorausgesetzt wurde, ist eine beliebige Abbildung, welche Untervektorräume auf Untervektorräume abbildet, nach dem Hauptsatz der projektiven Geometrie bereits $\mathbb{F}_q$-semilinear.

Es sei nun ι eine $d_{\mathrm{H},r}$-Isometrie von $\mathbb{F}_q^{rn}$, welche Untervektorräume auf Untervektorräume abbildet. Dann gibt es also $\alpha \in \mathrm{Aut}(R)$ und $B \in \mathrm{GL}_{rn}(\mathbb{F}_q)$, so dass $\iota(v) = \alpha(v)B^{-1}$ für alle $v \in \mathbb{F}_q^{rn}$ gilt. Sind nun $x \in [n]$ und $i \in [r]$ beliebig, so schließt man aus

$$d_{\mathrm{H},r}(\iota(e_{xr+i}), \iota(\mathbf{0_{rn}})) = d_{\mathrm{H},r}(e_{xr+i}, \mathbf{0_{rn}}) = 1,$$

dass $\mathrm{supp}(\iota(e_{xr+i})) \subseteq [(y+1)r] \setminus [yr]$ Teilmenge eines einzigen Koordinatenblocks für ein $y \in [n]$ ist. Nehmen wir nun an, es gäbe ein $j \in [r]$ mit $\mathrm{supp}(\iota(e_{xr+j})) \subseteq [(y'+1)r] \setminus [y'r]$ und $y' \neq y$, so folgt hieraus der Widerspruch:

$$\begin{aligned} 2 &= d_{\mathrm{H},r}(\iota(e_{xr+i}) + \iota(e_{xr+j}), \iota(\mathbf{0_{rn}})) = d_{\mathrm{H},r}(\iota(e_{xr+i} + e_{xr+j}), \iota(\mathbf{0_{rn}})) \\ &= d_{\mathrm{H},r}(e_{xr+i} + e_{xr+j}, \mathbf{0_{rn}}) = 1 \end{aligned}$$

Diese eindeutige Zuordnung $[n] \to [n],\ x \mapsto y(x)$ definiert eine Bijektion π auf $[n]$. Die Abbildung $\iota^{(\pi^{-1})} \circ \iota$ lässt sich dann aber über eine Blockdiagonalmatrix $B' \in (\mathrm{GL}_r(\mathbb{F}_q))^n$ und den Körperautomorphismus α ausdrücken. □

Insgesamt gewinnen wir den folgenden Satz:

6.3.3 Satz. *Es sei $rn \geq 3$. Zwei $\mathbb{F}_q$-lineare $\mathbb{F}_{q^r}$-Codes C, C' in $\mathbb{F}_{q^r}^n$ sind genau dann äquivalent, falls es für beliebige Generatormatrizen Γ von $T(C)$ und Γ' von $T(C')$ ein Gruppenelement*

$$(A, (B_0, \ldots, B_{n-1}); \alpha, \pi) \in (\mathrm{GL}_k(\mathbb{F}_q) \times (\mathrm{GL}_r(\mathbb{F}_q))^n) \rtimes (\mathrm{Aut}(\mathbb{F}_q) \times S_n)$$

gibt mit $A\alpha(\Gamma)\,\mathrm{diag}(B_0, \ldots, B_{n-1})^{-1} = \Gamma'$.

Beweis. Folgt aus der vorangegangenen Diskussion. □

6.3.3. Ein Kanonisierer

Wir haben gesehen, dass wir die Äquivalenz von Network-Codes über $\mathbb{F}_q$ und auch von $\mathbb{F}_q$-linearen $\mathbb{F}_{q^r}$-Codes durch eine Gruppenoperation der Gruppe $G \rtimes S_n$ mit

$$G := (\mathrm{GL}_k(\mathbb{F}_q) \times (\mathrm{GL}_r(\mathbb{F}_q))^n) \rtimes \mathrm{Aut}(\mathbb{F}_q)$$

auf $\mathbb{F}_q^{k \times rn}$ beschreiben können.

Das Kapitel 3.3 bietet uns wieder eine Möglichkeit, das Kanonisierungsproblem zu lösen. Da sich für diese Gruppenoperation die innere Kanonisierung aber wesentlich komplexer gestaltet, wird in [18] das nachfolgend beschriebene Vorgehen entwickelt.

Zuallererst wollen wir dieses in Zusammenhang mit dem zuvor entworfenen Kanonisierer für lineare Codes zu bringen. Wir gehen daher zunächst nochmals auf die äußere Verfeinerung aus dem Abschnitt 5.2.2 ein: Für lineare Codes über einem endlichen Körper $\mathbb{F}_q$ können wir die Definition der Menge $\overline{W}_\lambda(\Gamma)$ als eine Auswahl einer Teilmenge aller Hyperebenen in $\mathrm{PG}(\mathbb{F}_q^k)$ auffassen. Die ausgewählten Hyperebenen erfüllen hierbei gewisse Inklusionsbedingungen mit den durch die Spalten der Generatormatrix Γ ausgezeichneten Punkten.

In gleicher Weise gehen wir nun für die Operation von $G \rtimes S_n$ auf $\mathbb{F}_q^{k \times rn}$ vor. Im weiteren Verlauf sei $U = (U_0, \ldots, U_{n-1}) \in (\mathbb{F}_q^{k \times r})^n$ fest vorgegeben und $C = \{\langle U_i \rangle \mid i \in [n]\}$. Wir zeichnen wieder eine Teilmenge

$$W(C) := \left\{\langle h_0 \rangle^\perp, \ldots, \langle h_{m-1} \rangle^\perp\right\} \text{ mit } h_i \in \mathbb{F}_{q^k} \setminus \{\mathbf{0_k}\}, i \in [m] \text{ und } m \in \mathbb{N}$$

von Hyperebenen aus, welche eine fest vorgegebene Schnitteigenschaft mit den Unterräumen $\langle U_i \rangle \in C$ des Network-Codes C haben. Für den Kanonisierer ist es entscheidend, dass diese Zuordnung wieder einen $\Gamma\mathrm{L}_k(\mathbb{F}_q)$-Homomorphismus definiert, d.h. für ein beliebiges $(A; \alpha) \in \Gamma\mathrm{L}_k(\mathbb{F}_q)$ muss

$$\begin{aligned} W((A,\alpha)C) &= (A,\alpha) \cdot \left\{\langle h_0 \rangle^\perp, \ldots, \langle h_{m-1} \rangle^\perp\right\} \\ &= \left\{\left\langle (A^T)^{-1}\alpha(h_0)\right\rangle^\perp, \ldots, \left\langle (A^T)^{-1}\alpha(h_{m-1})\right\rangle^\perp\right\} \end{aligned}$$

gelten. Auch hier werden wir nun über diese Wahl einen bipartiten Graphen $\mathcal{G}(C) \in B([n],[m])$ zur Definition einer äußeren Verfeinerung gewinnen. Entscheidend ist jedoch, dass wir nicht nur den Graphen zur Definition des Backtrackalgorithmus einbringen, sondern auch die Menge $W(C)$ selbst.

Um eine kanonische Form von U zu berechnen, bilden wir einen Backtrackalgorithmus zu der Gruppenoperation von

$$\left(\mathrm{GL}_k(\mathbb{F}_q) \times (\mathrm{GL}_r(\mathbb{F}_q))^n \times \left(\mathbb{F}_q^*\right)^m\right) \rtimes (\mathrm{Aut}(\mathbb{F}_q) \times S_n \times S_m)$$

auf dem Vektor $(U_0, \ldots, U_{n-1}, h_0, \ldots, h_{m-1})$. Dies bietet den folgenden Vorteil:
Über die Fixierreihenfolge steuern wir die innere Kanonisierung zunächst derart, dass wir nur eine Auswahl der Hyperebenen $h_0, \ldots, h_{m-1}$ über die Operation von

$$\left(\mathrm{GL}_k(\mathbb{F}_q) \times \left(\mathbb{F}_q^*\right)^m\right) \rtimes \mathrm{Aut}(R)$$

minimieren. Dies erfolgt gerade über die Methoden, welche in [24] bzw. im Abschnitt 5.1.2 entwickelt wurden, und kann damit sehr effektiv implementiert werden[15]. Hierdurch erreichen wir sehr frühzeitig eine Einschränkung der Komponente $\mathrm{GL}_k(\mathbb{F}_q)$ auf eine Untergruppe von Blockmatrizen der folgenden Gestalt

$$\left\{ \begin{pmatrix} D & \mathbf{0}_{\mathbf{k'}\times(\mathbf{k}-\mathbf{k'})} \\ A^{(0)} & A^{(1)} \end{pmatrix} \mid \begin{array}{l} D \in \mathrm{GL}_{k'}(\mathbb{F}_q) \text{ Diagonalmatrix,} \\ A^{(1)} \in \mathrm{GL}_{k-k'}(\mathbb{F}_q) \text{ und } A^{(0)} \in \mathbb{F}_q^{(k-k')\times k'} \end{array} \right\}.$$

Ist dann $U_i = \begin{pmatrix} U_i^{(0)} \\ U_i^{(1)} \end{pmatrix}$ mit $U_i^{(0)} \in \mathbb{F}_q^{k'\times r}$ und $U_i^{(1)} \in \mathbb{F}_q^{(k-k')\times r}$, so ist

$$\begin{pmatrix} D & \mathbf{0}_{\mathbf{k'}\times(\mathbf{k}-\mathbf{k'})} \\ A^{(0)} & A^{(1)} \end{pmatrix} \begin{pmatrix} U_i^{(0)} \\ U_i^{(1)} \end{pmatrix} B_i^{-1} = \begin{pmatrix} DU_i^{(0)} B_i^{-1} \\ \left(A^{(0)} U_i^{(1)} + A^{(1)} U_i^{(1)} \right) B_i^{-1} \end{pmatrix}.$$

Mit dieser Beobachtung schließen wir, dass wir auch für den oberen $(k' \times r)$-Teilblock der Matrix U_i mit einer Transformation auf reduzierte Spaltenstufenform beginnen können. Hierdurch erzielen wir auch eine gute Einschränkung der Komponente $\mathrm{GL}_r(\mathbb{F}_q)$ an der Position $i \in [m]$ der operierenden Gruppe.

Auf die äußere Verfeinerung soll nun an dieser Stelle nicht mehr näher eingegangen werden; Details finden sich in [18]. Mit dieser Kurzbeschreibung wollen wir vor allem aufzeigen, dass für die neuesten Entwicklungen [17] in der Codierungstheorie, welche auch Network-Codes über beliebigen Kettenringen betrachten, mit unserer Arbeit möglicherweise ebenfalls der Grundstein gelegt wurde, um auch für diese Probleme einen effizienten Kanonisierer zur Verfügung zu stellen.

[15] Es muss lediglich beachtet werden, dass wir die Operation mit $A \in \mathrm{GL}_k(\mathbb{F}_q)$ auf den Vektoren h_i über $(A, h_i) \mapsto (A^{-1})^T \cdot h_i$ definieren.

7. Entwickelte Programme

Der Kanonisierer für lineare Codes über beliebigen Kettenringen wurde als Sage [70] Paket implementiert. Es trägt den Namen `codecan-1.1.spkg`. Wir beschreiben die Installation und die Funktionsweise in Abschnitt 7.1.2. Das Paket befindet sich auf der beiliegenden CD im Verzeichnis `sage`. Für lineare Codes über endlichen Körpern ist der Algorithmus, beginnend ab Version 6.1 von Sage, bereits in der Standardinstallation enthalten. Wir gehen in Abschnitt 7.1.1 im Rahmen einer Beispielsitzung hierauf ein.

Für lineare Codes über endlichen Körpern und Kettenringen der Kardinalität 4 existiert auch eine Implementierung in der Programmiersprache C++. Diese ist eine Fortentwicklung der Programme, welche im Zusammenhang mit den Arbeiten [20, 22] bereits erschienen sind. Auf das C++ Programm wird in Abschnitt 7.2 näher eingegangen. Es ist ebenfalls im Umfang der beiliegenden CD enthalten.

7.1. Sage

Wir gehen im Folgenden davon aus, dass Sage bereits installiert wurde[1].

7.1.1. Lineare Codes über endlichen Körpern

Lineare Codes über endlichen Körpern und viele weitere Methoden aus der Codierungstheorie über endlichen Körpern werden in Sage bereits standardmäßig bereitgestellt. Für weitergehende Informationen verweisen wir auf das entsprechende Kapitel[2] der Dokumentation [70]. Den binären Hamming-Code der Länge 7 erhält man etwa über den Aufruf[3]

```
sage: C = codes.HammingCode(3, GF(4, 'a'))
```

Alle Methoden, welche für das Objekt `C` zur Verfügung stehen, kann man sich innerhalb einer Sitzung von Sage über die sogenannte Tabulator-Vervollständigung anzeigen lassen. Dies erreicht man durch Drücken der Tabulator-Taste nach der Eingabe von

[1] Die notwendigen Schritte zur Installation werden in der offizielle Programmdokumentation `http://www.sagemath.org/doc/installation/` beschrieben.

[2] Siehe `http://www.sagemath.org/doc/reference/coding/index.html`.

[3] Zur Verdeutlichung, dass wir einen Befehl innerhalb einer Sitzung von Sage eingeben, beginnen wir diesen immer mit „`sage:`“. Diese Konvention übernehmen wir aus der offiziellen Programmdokumentation.

```
sage: C.[Tabulator]
```

Ebenso besteht die Möglichkeit, innerhalb einer Sitzung von Sage auf die gesamte Dokumentation interaktiv zuzugreifen. Dazu fügt man an das Objekt bzw. die Funktion ein Fragezeichen an. So wird etwa über die Befehle

```
sage: C?
sage: C.minimum_distance?
```

die Dokumentation der Klasse `LinearCode` bzw. der Methode `minimum_distance` angezeigt. In den meisten Fällen umfasst die Dokumentation auch Beispiele, welche die Benutzung illustrieren. Der Aufruf der Methode `canonical_representative`

```
sage: C_can, t = C.canonical_representative()
```

startet dann die Kanonisierung des linearen Codes `C`. Die Methode gibt ein Tupel, bestehend aus der kanonischen Form und einem Transporterelement, zurück. Über die Zuweisung speichern wir das Resultat in den Variablen `C_can` und `t`. Ein Erzeugendensystem `E` und die Ordnung `ord` der Automorphismengruppe des linearen Codes `C` erhält man schließlich über den Aufruf

```
sage: E, ord = C.automorphism_group_gens()
```

Beide Methoden erlauben die Eingabe eines optionalen Parameters `equivalence` zur Steuerung des genutzten Äquivalenzbegriffs. Standardmäßig ist dieser auf `'semilinear'` gesetzt. Es sind aber auch die Optionen `'linear'` und `'permutational'` zulässig. Entsprechend lauten die Funktionsaufrufe zur Kanonisierung und zur Bestimmung der Automorphismengruppe:

```
sage: C_can, t = C.canonical_representative(equivalence='linear')
sage: E, ord = C.automorphism_group_gens(equivalence='linear')
```

beziehungsweise

```
sage: C_can, t = C.canonical_representative(equivalence='permutational')
sage: E, ord = C.automorphism_group_gens(equivalence='permutational')
```

7.1.2. Lineare Codes über endlichen Kettenringen

Das Verzeichnis `sage`, welches sich auf der beiliegenden CD befindet, enthält neben dem Programmpaket `codecan-1.1.spkg` auch einen Unterverzeichnis `sage/doc`. Dieses beinhaltet die vollständige Dokumentation aller Klassen und Funktionen, welche durch das Paket `codecan-1.1.spkg` zur Verfügung gestellt werden. Sie beschreibt auch – in der Datei `sage/doc/html/index.html` – die zur Installation des Pakets notwendigen Schritte.

Das Paket ist kompatibel zu der zum Erscheinungszeitpunkt dieser Dissertation aktuellsten Version 6.1 von Sage [70].

Installation Zur Installation des Pakets `codecan-1.1.spkg` genügt es, Sage einmalig mit der Option

```
./sage -i codecan-1.1.spkg
```

zu starten. Dabei gehen wir davon aus, dass das Programmpaket `codecan-1.1.spkg` in das Installationsverzeichnis von Sage kopiert wurde. Damit stehen nun alle Funktionen des Pakets `codecan-1.1.spkg` bereits beim Start der nächsten Sitzung von Sage zur Verfügung.

Benutzung Da beliebige Kettenringe bislang nicht von Sage unterstützt werden, stellt das Paket `codecan-1.1.spkg` neben dem Kanonisierer auch eine Auswahl von 66 Kettenringen in einer Datenbank zur Verfügung. Wir laden die Datenbank und initialisieren zum Beispiel $R = \mathbb{Z}_4$ durch:

```
sage: from codecan.chain_ring import ChainRings
sage: R = ChainRings['Z4']
```

Eine vollständige Liste der bereitgestellten Kettenringe kann man sich über den Aufruf

```
sage: ChainRings.list_names()
```

anzeigen lassen.

Lineare Codes für beliebige endliche Kettenringe stehen gegenwärtig in Sage nicht zur Verfügung. Eine Problematik besteht hier zum Beispiel auch darin, dass Sage nicht an allen Stellen Moduln über nicht kommutativen Ringen[4] unterstützt. Daher und aus zeitlichen Gründen wurde auf eine Implementierung einer Klasse für ringlineare Codes im Rahmen des Pakets `codecan-1.1.spkg` verzichtet. In der Kanonisierung umgehen wir diesen Sachverhalt, indem wir den Kanonisierer ausschließlich für Generatormatrizen zur Verfügung stellen. Die genutzten Skalarmultiplikationen sind hierbei darauf getestet worden, dass sie die Multiplikationsreihenfolge tatsächlich respektieren. Gegebenenfalls wurden die notwendigen Operationen neu implementiert.

Auch hier wollen wir nun wieder die Benutzung des Pakets `codecan-1.1.spkg` an einer Beispielsitzung besprechen. Die Klasse, welche den Kanonisierer für Generatormatrizen linearer Codes über endlichen Kettenringen implementiert, laden wir über den Aufruf

```
sage: from codecan import RingLinearCode_AutGroupCanLabel
```

Nun legen wir zum Beispiel eine Generatormatrix `Gamma` des Kerdock-Codes $\mathcal{K}_{2,4}$ über den Befehl

[4] Wir werden gegebenenfalls durch die Warnung „*UserWarning: You are constructing a free module over a noncommutative ring. Sage does not have a concept of left/right and both sided modules, so be careful. It's also not guaranteed that all multiplications are done from the right side.*“ auch durch Sage davon in Kenntnis gesetzt.

```
sage: Gamma = matrix(R, 4, 8,
                     [[1,1,1,1,1,1,1,1],
                      [0,1,0,0,1,2,3,1],
                      [0,0,1,0,3,3,3,2],
                      [0,0,0,1,2,3,1,1]])
```

an und starten die Kanonisierung über

```
sage: CanZ4 = RingLinearCode_AutGroupCanLabel(Gamma)
```

Über weitere optionale Parameter lässt sich wiederum eine Einschränkung der operierenden Gruppe erreichen. Für Details verweisen wir an dieser Stelle auf die Dokumentation der Klasse `RingLinearCode_AutGroupCanLabel`. Über die bereitgestellten Methoden

```
sage: CanZ4.get_canonical_form()
sage: CanZ4.get_transporter()
sage: CanZ4.get_autom_gens()
```

erlangen wir schließlich Zugriff auf die berechneten Daten.

Weiteres Datenmaterial Im Verzeichnis `sage/data` kann der Leser auf die Generatormatrizen der linearen Codes zugreifen, welche in dem Kapitel 6.1 besprochen wurden. Die Unterverzeichnisse der Gestalt `T_q_k_s` beinhalten die Generatormatrizen der Teichmüller-Codes $\mathcal{T}_{q,k,s}$. Das Unterverzeichnis `kerdock` umfasst die Kerdock-Codes $\mathcal{K}_{4,4}$, $\mathcal{K}_{4,6}$ und $\mathcal{K}_{8,4}$. Über den Befehl

```
sage: Gamma = load(<file>)
```

lädt man den Inhalt der Datei `<file>` und speichert diesen in der Variablen `Gamma`. Anschließend kann man zum Beispiel die Kanonisierung der Generatormatrix `Gamma`, wie oben beschrieben, starten.

7.2. C++ Implementierung

Eine C++ Implementierung des Kanonisierers für lineare Codes über endlichen Körpern und den beiden echten Kettenringen der Kardinalität 4 befindet sich ebenfalls auf der beiliegenden CD im Verzeichnis `c++`.

7.2.1. Installation

Das Programm lässt sich über den Aufruf von `make` im Unterverzeichnis `c++/codecan` unter Linux compilieren. Folgende Vorarbeiten bzw. Voraussetzungen sind hierbei für einen fehlerfreien Übersetzungslauf zu gewährleisten:

- Es muss ein Compiler für *C++* auf dem Betriebssystem installiert sein. Der Befehl `make` geht hierbei davon aus, dass der Compiler über den Befehl `gcc` (für *GNU Compiler Collection*) aufgerufen wird. Ist dies nicht der Fall, so ist die Datei `makefile` entsprechend zu modifizieren.

- Des Weiteren werden die *Boost C++ Libraries* [4] und die *GNU Multiple Precision Arithmetic Library* [29] benötigt. Wir gehen davon aus, dass deren Verzeichnisse für den Compiliervorgang in die Suchpfade für die Include-Anweisungen und für das Verlinken aufgenommen wurden.

- Schließlich geht das Programm auch davon aus, dass auf mindestens eines der Computeralgebrasysteme GAP [28], Sage [70] oder Magma [53] zugegriffen werden kann[5]. An dieser Stelle ist es auch nötig, den entsprechenden Pfad zum Aufruf dieser Systeme dem Programm `codecan`, über das Setzen der Präprozessorvariablen `CAS_CALL` in der Datei `makefile`, bereits beim Compilieren mitzuteilen. Die getroffene Wahl des Computeralgebrasystems muss ebenfalls über das Setzen einer Präprozessorvariablen mit dem Namen `CAS_SYSTEM` in der Datei `makefile` dokumentiert werden. Sie nimmt zwingend einen der folgenden Werte `USE_GAP`, `USE_SAGE` oder `USE_MAGMA` an.

Das Programm sollte sich nun mit diesen Voraussetzungen fehlerfrei über den Aufruf von `make` compilieren lassen. Die ausführbare Datei trägt den Namen `codecan`.

7.2.2. Benutzung

Eine ausführliche Dokumentation über die Benutzung des C++ Programms `codecan` wird in der Datei `c++/codecan/Readme` gegeben. Wir behandeln hier nur den Standardaufruf des Kanonisierers mit der Option `-canonize` und unterscheiden diesen zunächst nach der Art des Alphabets R

$R = \mathbb{Z}_p$, $p = 4$ oder p prim:

```
./codecan <algorithm_type> Z<p> -canonize <matrix_file>
```

$R = \mathbb{F}_q$, q Primzahlpotenz:

```
./codecan <algorithm_type> F <field_filename> -canonize <matrix_file>
```

$R = \mathbb{F}_2[X]/(X^2)$:

```
./codecan <algorithm_type> F2_F2 -canonize <matrix_file>
```

Über den Parameter `<algorithm_type>` steuert der Programmbenutzer die Wahl der operierenden Gruppe. Es stehen die folgenden Wahlmöglichkeiten zur Verfügung:

[5] Da das Computeralgebrasystem GAP bereits in der Installation des Computeralgebrasystems Sage eingebunden ist, wird über Sage nur auf diese Installation von GAP zugegriffen.

`permutational`: Kanonisierung der Generatormatrix unter der Gruppenoperation von $\mathrm{GL}_k(R) \times S_n$

`plusminusmonomial`: Kanonisierung der Generatormatrix unter der Gruppenoperation von $(\mathrm{GL}_k(R) \times \{1, -1\}^n) \rtimes S_n$

`linear`: Kanonisierung der Generatormatrix unter der Gruppenoperation von $(\mathrm{GL}_k(R) \times R^{*n}) \rtimes S_n$

`semilinear`: Kanonisierung der Generatormatrix unter der Gruppenoperation von $(\mathrm{GL}_k(R) \times R^{*n}) \rtimes (\mathrm{Aut}(R) \times S_n)$

Die Elemente des endlichen Körpers $\mathbb{F}_q$ werden zur Ein- und Ausgabe mit den Zahlen $\{0, 1, \ldots, q-1\}$ identifiziert. Einzige Voraussetzung, welche an diese Bijektion gestellt wird, ist, dass das Nullelement des Körpers auf 0 und das Einselement auf 1 abgebildet werden. Die Arithmetik des Körpers wird über Tabellen verwaltet. Sie werden dem Programm über die Datei `<field_filename>` übergeben. Deren Aufbau soll an dieser Stelle nicht weiter behandelt werden. Verschiedene endliche Körper werden im Unterverzeichnis `c++/codecan/galoisfields/` bereitgestellt.

Die zu kanonisierende Generatormatrix $\Gamma \in R^{k \times n}$ wird schließlich über die Datei `<matrix_file>` dem Programm übergeben. Diese Datei muss hierbei die folgende Struktur aufweisen:

$$\begin{array}{cccc} n & k & & \\ \Gamma_{0,0} & \Gamma_{0,1} & \ldots & \Gamma_{0,n-1} \\ \vdots & \vdots & \ldots & \vdots \\ \Gamma_{k-1,0} & \Gamma_{k-1,1} & \ldots & \Gamma_{k-1,n-1} \end{array}$$

Im Unterverzeichnis `c++/codecan/input_matrices/` werden zum Beispiel Generatormatrizen der Kerdock-Codes $\mathcal{K}_{2,k+1}$ zur Verfügung gestellt. Für den Kerdock-Code $\mathcal{K}_{2,4}$ lautet somit die Eingabe zum Start des Kanonisierers:

```
./codecan linear Z4 -canonize input_matrices/kerdock4.txt
```

8. Zusammenfassung & Ausblick

In dieser Arbeit wurde ein Kanonisierer für lineare Codes über endlichen Kettenringen entworfen. Er ist eine konsequente Fortentwicklung des Kanonisierers für lineare Codes über endlichen Körpern, welcher in den Arbeiten [22, 24] angegeben wurde.

Zur Beschreibung des Kanonisierers wurde zunächst allgemein das Kanonisierungsproblem für eine beliebige Gruppe G und eine G-Menge X untersucht. Häufig bildet das bekannte Verfahren, über Partitionen und Verfeinerungen einen Backtrackalgorithmus zu definieren, das Mittel der Wahl, um einen kanonischen Repräsentanten für $x \in X$ zu definieren. Dieses Verfahren ist seit langem, etwa im Zusammenhang mit der Kanonisierung von Graphen, bekannt. Andererseits tritt gerade dort aber die symmetrische Gruppe S_n häufig zugunsten der Datenstruktur bei der Beschreibung der Algorithmen in den Hintergrund, siehe etwa [41, 55, 56]. Dies erschwert die Übertragung der maßgeblichen Ideen auf allgemeinere Gruppenoperationen. In dieser Dissertation wurde daher sehr viel Wert darauf gelegt, eine detaillierte Beschreibung eines Kanonisierers für eine beliebige Gruppenoperation von G auf X anzugeben. Die Beschreibung beruht zu großen Teilen auf der Kombination der Arbeiten [35] und [42] sowie eigenen Ideen.

Sowohl die theoretischen Grundlagen als auch die notwendigen Ideen zur Gewinnung praxistauglicher Kanonisierer wurden in einer Form beschrieben, welche es erlaubt, auch weitere Gruppenoperation einer beliebigen Gruppe G auf einer Menge X zu untersuchen. Insbesondere bei der Operation einer Gruppe G auf der Menge aller n-elementigen Teilmengen von X sollte der Übergang zu einer Operation von $G \times S_n$ auf X^n als alternativer Lösungsansatz für die Kanonisierung berücksichtigt werden.

Bei der Kanonisierung linearer Codes über einem Kettenring R mit diesem Verfahren bietet das Homomorphieprinzip eine unüberschaubare Fülle möglicher Verfeinerungen an. Es wurde in dieser Arbeit nur eine kleine Auswahl aller Möglichkeiten angegeben. Hier liegt wohl – für die Zukunft – der beste Ansatzpunkt, um weitere Verbesserungen des Algorithmus zu erreichen. Auch kann nicht ausgeschlossen werden, dass die strikte Aufteilung der Verfeinerungsschritte in eine innere Kanonisierung und eine äußere Verfeinerung sich für einen linearen Code über einem beliebigen, endlichen Kettenring nicht nachteilig auswirkt. Wir haben uns an dieser Stelle von der herausragenden Performance des Kanonisierers für lineare Codes über endlichen Körpern leiten lassen. Insbesondere wurde das Vertrauen dadurch bestärkt, dass dieser Algorithmus gerade bei ansteigender Kardinalität des Körpers $\mathbb{F}_q$, im Vergleich zur Implementierung von Leons Algorithmus ([51]) in Magma [53], besser abschneidet. Die Laufzeiten der entwickelten Programme – auf vielen getesteten Probleminstanzen – deuten an, dass der vorgestellte Ansatz wohl auch für R-lineare Codes eine effiziente Kanonisierung ermöglicht.

Ein weiterer, möglicher Ansatz zur Definition des Kanonisierers bestünde etwa auch darin, die Generatormatrix Γ sukzessiv für $i \in [m]$ über die Abbildung $R \to R/\operatorname{Rad}(R)^i$ als Generatormatrix eines Codes über dem Kettenring $R/\operatorname{Rad}(R)^i$ der Kettenlänge i aufzufassen. Auch hier bietet das Homomorphieprinzip die Möglichkeit, die Ergebnisse induktiv auf die Kettenringe mit größerer Kettenlänge zu übertragen. Dies wurde bislang noch nicht untersucht.

In Abschnitt 2.4 wurde bewiesen, dass die Entscheidungsprobleme SCE_R, LCE_R und PCE_R für jeden beliebigen Kettenring R mindestens so schwer wie das Graphenisomorphieproblem GI sind. Umgekehrt ist bekannt, dass für endliche Körper $\mathbb{F}_q$ die Probleme $\mathrm{PCE}_{\mathbb{F}_q}$ vermutlich nicht **NP-vollständig** sind. Hier drängt sich sofort die Frage auf, ob diese Aussage auch auf beliebige Kettenringe R verallgemeinerbar ist, und wie sich die Probleme SCE_R und LCE_R verhalten.

In der Anwendung des Kanonisierers für lineare Codes über Galois-Ringe der Charakteristik 4 sind Ergebnisse aufgetreten, die sich wahrscheinlich nicht zufällig ergeben haben. Aus ihnen wurden die Vermutungen 6.1.6 und 6.1.11 gewonnen, welche durch weitere Rechnungen, auf größeren Instanzen, untermauert werden sollten. Gegebenenfalls sollte dann auch ein Beweis der Aussagen erfolgen.

Über die Umkehrung der Konstruktion Y_1 konnte ein Klassifikationsalgorithmus für lineare $[n, k, \geq d]_q^{d^\perp}$-Codes über endlichen Körpern entwickelt werden, welcher in der Lage ist, sehr tief vorzudringen. Dies liegt daran, dass die Transversalen $T(n-(d^\perp-i), k-(d^\perp-i), d, i, q)$ der Zwischenschritte $i \in \left[d^\perp\right]$ vergleichsweise moderate Mächtigkeiten aufweisen. Eine Verallgemeinerung dieses Vorgehens sollte auch für beliebige Kettenringe R angestrebt werden. Diese Klassifikation könnte dann zum Beispiel für (optimale Codes) wesentlich weiter vordringen als das Verfahren, welches in Abschnitt 6.2.3 zum Einsatz kam. Wir versprechen uns hiervon, tiefer liegende Erkenntnisse über R-lineare Codes zu gewinnen.

Schließlich ist dem Autor bewusst, dass der Übergang von einer C++ Implementierung des Kanonisierers zu der Realisierung in Sage zu einem nicht zu vernachlässigenden Anstieg in der Rechenzeit geführt hat. Sage besteht zu großen Teilen aus Programmcode, welcher zur Laufzeit ausgewertet wird. Nur zeitkritische Operationen liegen teilweise in compilierter Form vor. Aus Sicht der Implementierung wurde versucht, möglichst große Anteile des Pakets `codecan-1.1.spkg` über compilierten Quellcode umzusetzen. Jedoch gibt es an bestimmten Stellen der von Sage zur Verfügung gestellten Methoden Flaschenhälse, welche die schnelle Programmausführung stark behindern. Es besteht aber die Hoffnung, dass diese – durch eine stetige Weiterentwicklung von Sage – bereits in naher Zukunft beseitigt werden. Hier profitiert das Paket von den gemeinschaftlichen Anstrengungen eines weltweit verbreiteten Open-Source Projekts.

Andererseits bietet ein frei zugängliches Computeralgebrasystem auch die Chance, einen größeren Benutzerkreis zu erreichen und komplizierte Installationen und Abhängigkeiten von bestimmten Betriebssystemen zu umgehen. Es besteht also die Aussicht, mit dem Sage Paket `codecan-1.1.spkg` einen weitaus größeren Nutzerkreis zu erreichen.

Schließlich wird es auch potentiellen Programmierern erleichtert – über die genutzten mathematischen Objekte in Sage – den Quellcode zu verstehen, Verbesserungen an dem Kanonisierer vorzunehmen und Teile des Quellcodes in andere Projekte einzubringen. Insbesondere hinsichtlich des letztgenannten Punkts wurde die Implementierung derart vorgenommen, dass ein Kanonisierer für eine weitere Gruppenoperationen (der Gestalt $G \rtimes S_{\mathfrak{P}_0}$ operiert auf einer Menge X^n) sehr leicht über die Ableitung einer abstrakten Basisklasse realisiert werden kann.

Ist es aus Effizienzgründen dennoch nötig, eine Implementierung des Kanonisierers für R-lineare Codes in C++ vorzunehmen, so wurden auch dort – über den Einsatz von Templates und einem sehr allgemein gehaltenem Layout der Klassen – bereits optimale Voraussetzungen geschaffen, um die vorhandene Implementierung um diese Fälle zu ergänzen. Ausgehend von beiden Implementierungen in C++ und Sage ließe sich etwa auch eine Kanonisierung von Network-Codes über einem beliebigen Kettenring R leicht realisieren.

A. Erzeugendensysteme von Untergruppen der Automorphismengruppe eines Kettenrings

Wir werden in diesem Anhang einen Algorithmus A.1 entwickeln, welcher ein Erzeugendensystem der Automorphismengruppe Aut_T des Kettenrings R berechnet. Des Weiteren ist dieses Erzeugendensystem an die Normalreihe (4.1) angepasst. Der Beweis der Korrektheit erfolgt über mehrere Hilfssätze. Abschließend geben wir dann in Bemerkung A.5 die notwendigen Modifikationen an dem Algorithmus um ein Erzeugendensystem der Gruppe $\mathrm{Out}(R)$ zu bestimmen.

A.1 Hilfssatz. *In Zeile 10 von Algorithmus A.1 ist E ein Erzeugendensystem für* $\mathrm{Aut}_\xi^{(i,h)} := \{\alpha \in \mathrm{Aut}_\xi \mid (\alpha(\theta) - \theta) \in R^{(i,h)}\}$.

Beweis. Wir beweisen die Aussage induktiv, indem wir für den Fall $\mathrm{Aut}_\xi^{(i,h)} \setminus \mathrm{Aut}_\xi^{(i+1,h)} \neq \emptyset$ einen Nebenklassenvertreter der Gestalt χ_ξ^ω mit $\omega = \theta + \xi^{r-1-i}\theta^h + \sum_{\ell=0}^{|B|-1} x_\ell B_\ell$ für ein $x \in [p]^B$ angeben:
Wir wählen zunächst $\alpha \in \mathrm{Aut}_\xi^{(i,h)} \setminus \mathrm{Aut}_\xi^{(i+1,h)}$ beliebig. Dann ist die (ξ,θ)-adische Entwicklung der Differenz $\alpha(\theta) - \theta$ gleich

$$\alpha(\theta) - \theta =: \sum_{\ell=i}^{r-1} y_{\ell,h}\xi^{r-1-\ell}\theta^h + \sum_{\ell=0}^{r-1}\sum_{j=h+1}^{m-1} y_{\ell,j}\xi^{r-1-\ell}\theta^j$$

für die eindeutig bestimmte Matrix $y = (y_{l,j}) = \mathrm{coeff}(\alpha(\theta) - \theta) \in [p]^{r\times m}$. Außerdem ist ohne Beschränkung der Allgemeinheit $y_{i,h} = 1$, ansonsten betrachten wir eine geeignete Potenz von α mit dieser Eigenschaft.

Es sei nun $(\ell, j) \in [r] \times [m]$ mit $j > h$ oder $j = h$ und $\ell > i$ gegeben. Weiter sei $\xi^{r-1-\ell}\theta^j \notin B$ beliebig und $\beta \in E$ der Nebenklassenrepräsentant von $\mathrm{Aut}_\xi^{(\ell+1,j)}$ in $\mathrm{Aut}_\xi^{(\ell,j)}$. Es ist $\alpha \in \langle E\rangle$, genau dann wenn sich auch der Automorphismus $\alpha \circ \beta^{p-y_{\ell,j}}$ über die Erzeuger in E erzeugen lässt. Nutzen wir diese Äquivalenzen nun entlang der Kette

$$\mathrm{Aut}_\xi^{(i+1,h)} \trianglerighteq \ldots \trianglerighteq \mathrm{Aut}_\xi^{(r,h)} = \mathrm{Aut}_\xi^{(0,h+1)} \trianglerighteq \ldots \trianglerighteq \mathrm{Aut}_\xi^{(r,m-1)} = \{\mathrm{id}_R\},$$

Algorithmus A.1 Berechnung eines Erzeugendensystems für Aut_T

Input: R Kettenring mit $\theta\xi = \tau^e(\xi)\theta$, $R/\operatorname{Rad}(R) \simeq \mathbb{F}_{p^r}$ und $s := \operatorname{ggT}(e, r)$
Output: E Erzeugendensystem von Aut_T gemäß der Normalreihe (4.1)

1: **procedure** AUTOMORPHISM_GENS(R)
2: $E \leftarrow ()$
3: $B \leftarrow ()$
4: **for** $h \leftarrow m-1$ **to** 2 **by** -1 **do**
5: **for** $i \in [r]$ **do**
6: **if** $\exists\, x \in [p]^B : \chi_\xi^{\theta+\xi^{r-1-i}\theta^h+\sum_{\ell=0}^{|B|-1} x_\ell B_\ell} \in \mathrm{Aut}(R)$ **then**
7: APPEND($E, \chi_\xi^{\theta+\xi^{r-1-i}\theta^h+\sum_{\ell=0}^{|B|-1} x_\ell B_\ell}$)
8: **else**
9: APPEND($B, \xi^{r-1-i}\theta^h$)
10: // E ist nun ein Erzeugendensystem für $\mathrm{Aut}_\xi^{(i,h)}$
11: **for** $i \in \{1, \ldots, p^s-1\} : i \mid (p^s-1)$ **do**
12: **for** $x \in [p]^B$ **do**
13: $\omega \leftarrow \xi^i\theta + \sum_{\ell=0}^{|B|-1} x_\ell B_\ell \cdot \overline{\mathrm{coeff}}^{(\mathrm{ht}(B_\ell))}\Big((\xi^i\theta)^{\mathrm{ht}(B_\ell)}\Big)$
14: **if** $\chi_\xi^\omega \in \mathrm{Aut}(R)$ **then**
15: APPEND(E, χ_ξ^ω)
16: **break** i // Verlasse beide Schleifen
17: // E ist nun ein Erzeugendensystem von Aut_ξ
18: **for** $t \in \{1, \ldots, r-1\} : t \mid r$ **do**
19: **for** $j \in [i]$ **do**
20: **for** $x \in [p]^B$ **do**
21: $\omega \leftarrow \xi^j\theta + \sum_{\ell=0}^{|B|-1} x_\ell B_\ell^{p^t} \cdot \overline{\mathrm{coeff}}^{(\mathrm{ht}(B_\ell))}\Big((\xi^j\theta)^{\mathrm{ht}(B_\ell)}\Big)$
22: **if** $\chi_{\xi^{p^t}}^\omega \in \mathrm{Aut}(R)$ **then**
23: APPEND($E, \chi_{\xi^{p^t}}^\omega$)
24: **break** t // Verlasse alle drei Schleifen
25: **return** E

so erhalten wir mit Hilfssatz 4.2.7 einen Automorphismus $\alpha \in \mathrm{Aut}_\xi^{(i,h)} \setminus \mathrm{Aut}_\xi^{(i+1,h)}$ mit

$$\alpha(\theta) - \theta = \xi^{r-1-\ell}\theta^j + \sum_{\ell=0}^{|B|-1} x_\ell B_\ell \text{ mit } x \in [p]^B.$$

Dieser ist ein Nebenklassenrepräsentant der vorgeschriebenen Gestalt. □

A.2 Folgerung. *Es sei B der Vektor aus Algorithmus A.1 und $2 \leq h < m$ beliebig. Dann gilt*

$$\mathbb{F}_q = \Big\langle \overline{\mathrm{coeff}}^{(h)}(\alpha(\theta) - \theta) \mid \alpha \in \mathrm{Aut}_\xi^{(0,h)} \Big\rangle_{\mathbb{F}_p} \oplus \Big\langle \overline{\mathrm{coeff}}^{(h)}(b) \mid b \in B : \mathrm{ht}(b) = h \Big\rangle_{\mathbb{F}_p} \quad \text{(A.1)}$$

als $\mathbb{F}_p$-Vektorraum.

A.3 Hilfssatz. *In Zeile 17 von Algorithmus A.1 ist E ein Erzeugendensystem für Aut_ξ.*

Beweis. Wie im vorangegangenen Beweis, wollen wir wieder aus einem gegebenen Erzeuger $\alpha \circ \mathrm{Aut}_\xi^{(0,2)}$ von $\mathrm{Aut}_\xi / \mathrm{Aut}_\xi^{(0,2)}$ einen weiteren Nebenklassenrepräsentanten der in Zeile 15 angegebenen Gestalt herleiten. In Hilfssatz 4.2.8 haben wir die Abbildung $\mathrm{Aut}_\xi \to \mathbb{F}_q^*, \quad \alpha' \mapsto \overline{\mathrm{coeff}}^{(1)}(\alpha'(\theta))$ als Homomorphismus identifiziert. Ohne Beschränkung der Allgemeinheit können wir also davon ausgehen, dass der Exponent $i > 0$ in $\overline{\mathrm{coeff}}^{(1)}(\alpha(\theta)) = \overline{\xi}^i$ des vorliegenden Automorphismus minimal und damit ein Teiler von $p^s - 1$ ist.

Wir werden nun die Koeffizienten der θ-adischen Entwicklung von $\alpha(\theta) = \sum_{\ell=1}^{m-1} a_\ell \theta^\ell$ induktiv auf die notwendige Form bringen:
Es sei $2 \leq h < m$ und $\beta \in \mathrm{Aut}_\xi^{(0,h)} \setminus \mathrm{Aut}_\xi^{(0,h+1)}$ mit θ-adischer Entwicklung $\beta(\theta) = \theta + \sum_{\ell=h}^{m-1} b_\ell \theta^\ell$ beliebig. Dann gilt:

$$\begin{aligned}\alpha \circ \beta(\theta) &= \alpha\left(\theta + \sum_{\ell=h}^{m-1} b_\ell \theta^\ell\right) = \alpha(\theta) + \sum_{\ell=h}^{m-1} b_\ell \cdot \alpha(\theta)^\ell \\ &\equiv \sum_{\ell=1}^{h-1} a_\ell \theta^\ell + \Big(a_h + b_h \cdot \overline{\mathrm{coeff}}^{(h)}\big((\xi^i\theta)^h\big)\Big)\theta^h \mod \mathrm{Rad}(R)^{h+1}\end{aligned}$$

Da die Rechtsmultiplikation mit $c = \overline{\mathrm{coeff}}^{(h)}\big((\xi^i\theta)^h\big)$ eine $\mathbb{F}_p$-lineare, bijektive Abbildung definiert, ist auch

$$\mathbb{F}_q = \Big\langle \overline{\mathrm{coeff}}^{(h)}(\beta(\theta) - \theta) \cdot c \mid \beta \in \mathrm{Aut}_\xi^{(0,h)} \Big\rangle_{\mathbb{F}_p} \oplus \langle \overline{\mathrm{coeff}}^{(h)}(b) \cdot c \mid b \in B, \mathrm{ht}(b) = h \rangle_{\mathbb{F}_p}.$$

Somit können wir ohne Beschränkung der Allgemeinheit annehmen, dass $\overline{\mathrm{coeff}}^{(h)}(\alpha(\theta))$ in $\langle \overline{\mathrm{coeff}}^{(h)}(b) \cdot c \mid b \in B : \mathrm{ht}(b) = h \rangle_{\mathbb{F}_p}$ liegt. Damit zeigt man induktiv die Existenz eines

Vektors $x \in [p]^B$ mit

$$\omega = \xi^i\theta + \sum_{\ell=0}^{|B|-1} x_\ell \cdot \overline{\text{coeff}}^{(\text{ht}(B_\ell))}\Big(\big(\xi^i\theta\big)^{\text{ht}(B_\ell)}\Big) \cdot B_\ell$$

und $\alpha \circ \text{Aut}_\xi^{(0,2)} = \chi_\xi^\omega \circ \text{Aut}_\xi^{(0,2)}$. □

A.4 Satz. *Algorithmus A.1 ist korrekt.*

Beweis. Nach den beiden vorausgegangenen Hilfssätzen bleibt nur noch zu zeigen, dass der letzte Erzeuger passend gewählt wurde. Wir nehmen an, es sei $t = \frac{r}{r'}$ für die Zahl r' aus Hilfssatz 4.2.6 und $\alpha \in \text{Aut}(R)$ mit $\alpha(\xi) = \xi^{p^t}$.

Liegt der Exponent $j \in [q-1]$ von $\overline{\xi}^j := \overline{\text{coeff}}^{(1)}(\alpha(\theta))$ nicht bereits in der Menge $[i]$, so setzen wir $\ell = -p^{r-t}\lfloor \frac{j}{i} \rfloor$ und ersetzen α durch $\alpha \circ \big(\chi_\xi^\omega\big)^\ell$, wobei χ_ξ^ω der Automorphismus aus Zeile 15 ist. Dann ist

$$\begin{aligned}\alpha \circ \big(\chi_\xi^\omega\big)^\ell(\theta) &\equiv \alpha(\xi^{i\ell}\theta) \equiv \alpha(\xi)^{i\ell}\alpha(\theta) \equiv \Big(\xi^{p^t}\Big)^{i\ell}\xi^j\theta \equiv \big(\xi^{-p^r}\big)^{i\lfloor \frac{j}{i} \rfloor}\xi^j\theta \\ &\equiv \xi^{-i\lfloor \frac{j}{i} \rfloor + j}\theta \equiv \xi^{j-i\lfloor \frac{j}{i} \rfloor}\theta \qquad \mod \text{Rad}(R)^2\end{aligned}$$

Wir können also ohne Beschränkung der Allgemeinheit annehmen, dass $j \in [i]$ gilt. Die weiteren Koeffizienten der θ-adischen Entwicklung von $\alpha(\theta) = \sum_{\ell=1}^{m-1} a_\ell\theta^\ell$ werden wir, wie im vorangegangenen Hilfssatz, auf die notwendige Form bringen:
Wieder sei $2 \le h < m$ und $\beta \in \text{Aut}_\xi^{(0,h)} \setminus \text{Aut}_\xi^{(0,h+1)}$ mit θ-adischer Entwicklung $\beta(\theta) = \theta + \sum_{\ell=h}^{m-1} b_\ell\theta^\ell$ beliebig. Dann gilt

$$\begin{aligned}\alpha \circ \beta(\theta) &= \alpha\left(\theta + \sum_{\ell=h}^{m-1} b_\ell\theta^\ell\right) = \alpha(\theta) + \sum_{\ell=h}^{m-1} \alpha(b_\ell) \cdot \alpha(\theta)^\ell \\ &\equiv \sum_{\ell=1}^{h-1} a_\ell\theta^\ell + \Big(a_h + b_h^{p^t} \cdot \overline{\text{coeff}}^{(h)}\big(\xi^j\theta\big)\Big)\theta^h \mod \text{Rad}(R)^{h+1}\end{aligned}$$

Da auch die Anwendung des Frobenius-Automorphismus eine $\mathbb{F}_p$-lineare, bijektive Abbildung definiert, können wir wie oben argumentieren. Es genügt also in diesem Fall wieder

$$\left\langle \Big(\overline{\text{coeff}}^{(h)}(b)\Big)^{p^t} \cdot \overline{\text{coeff}}^{(h)}\big(\xi^j\theta\big) \mid b \in B, \text{ht}(b) = h \right\rangle_{\mathbb{F}_p}$$

als ein Vertretersystem möglicher Koeffizienten von $\overline{\text{coeff}}^{(h)}(\alpha(\theta))$ zu wählen. □

A.5 Bemerkung. Die nachfolgende Änderung an Algorithmus A.1 führt zu einer Berechnung eines Erzeugendensystems von $\mathrm{Out}(R)$ gemäß der Normalreihe (4.2) aus Satz 4.2.17. Das Vorgehen begründet sich sofort mit der in Satz 4.2.15 entwickelten Normalreihe von Inn_ξ:
In Zeile 5 werden wir zusätzlich unterscheiden, ob $h \equiv 1 \mod s$ gilt. Nur in diesen Fällen treten innere Automorphismen auf und es sind somit Modifikationen nötig. Ist in diesem Fall die Schleifenvariable $i \in [r]$ derart, dass $\mathrm{coeff}^{(i,0)}(t - \tau^e(t)) \neq 0$ und $\mathrm{coeff}^{(i',0)}(t - \tau^e(t)) = 0$ für alle $i' \in [i]$ gilt, so können wir die Bearbeitung dieses Blocks überspringen und mit dem nächsten i fortfahren.

Literatur

[1] Y. Alkhamees. „The group of automorphisms of finite chain rings.“ *Arab Gulf Journal for Scientific Research*, Bd. 8(3) (1990), S. 17–28.

[2] M. Barnabei, D. Searby und C. Zucchini. „On small {k; q}-arcs in planes of order q^2“. *Journal of Combinatorial Theory, Series A*, Bd. 24(2) (1978), S. 241 –246.

[3] A. Betten, M. Braun, H. Fripertinger, A. Kerber, A. Kohnert und A. Wassermann. *Error-correcting linear codes. Classification by isometry and applications.* Bd. 18. Algorithms and Computation in Mathematics. `http://linearcodes.uni-bayreuth.de`. Berlin: Springer-Verlag, 2006. 798 S.

[4] *Boost C++ Libraries*. Boost Community. URL: `http://www.boost.org/` (Stand 12. 07. 2013).

[5] M. Borello. *The automorphism group of a self-dual [72,36,16] code is not an elementary abelian group of order 8*. 2013. arXiv:`1304.7162v2 [cs.IT]`.

[6] W. Bosma, J. Cannon und C. Playoust. „The Magma algebra system. I: The user language.“ *Journal of Symbolic Computation*, Bd. 24(3-4) (1997), S. 235–265.

[7] I. G. Bouyukliev. „About the code equivalence“. *Advances in coding theory and cryptography*. Bd. 3. Ser. Coding Theory Cryptol. World Sci. Publ., Hackensack, NJ, 2007, S. 126–151.

[8] A. Brouwer und L. Tolhuizen. „A sharpening of the Johnson bound for binary linear codes and the nonexistence of linear codes with preparata parameters“. *Designs, Codes and Cryptography*, Bd. 3(2) (1993), S. 95–98.

[9] A. R. Calderbank, E. M. Rains, P. W. Shor und N. J. A. Sloane. „Quantum error correction via codes over GF(4)“. *IEEE Transactions on Information Theory*, Bd. 44(4) (1998), S. 1369–1387.

[10] C. Carlet. „Vectorial Boolean functions for cryptography“. *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*. Hrsg. von Y. Crama und P. L. Hammer. Bd. 134. Encyclopedia of Mathematics and its Applications. Cambridge Univ. Press, 2010, S. 398–469.

[11] W. E. Clark und D. A. Drake. „Finite chain rings“. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, Bd. 39 (1 1973), S. 147–153.

[12] W. E. Clark. „A coefficient ring for finite non-commutative rings“. *Proceedings of the American Mathematical Society*, Bd. 33 (1972), S. 25–28.

[13] J. Conway und V. Pless. „On primes dividing the group order of a doubly-even (72, 36, 16) code and the group order of a quaternary (24, 12, 10) code“. *Discrete Mathematics*, Bd. 38(2–3) (1982), S. 157 –162.

[14] S. Cook. „The P versus NP problem“. *The millennium prize problems*. Clay Math. Inst., Cambridge, MA, 2006, S. 87–104.

[15] J. Cramwinckel, E. Roijackers, R. Baart, E. Minkes, L. Ruscio und D. Joyner. *GUAVA, a GAP package for computing with error-correcting codes*. URL: `http://www.gap-system.org/Packages/guava.html`.

[16] Y. Edel und A. Pott. „On the equivalence of nonlinear functions“. *Enhancing cryptographic primitives with techniques from error correcting codes*. Hrsg. von B. Preneel, S. Dodunekov, V. Rijmen und S. Nikova. Bd. 23. NATO Sci. Peace Secur. Ser. D Inf. Commun. Secur. Amsterdam: IOS, 2009, S. 87–103.

[17] C. Feng, R. W. Nobrega, F. R. Kschischang und D. Silva. *Communication over Finite-Chain-Ring Matrix Channels*. 2013. arXiv:`1304.2523 [cs.IT]`.

[18] T. Feulner. *Canonical Forms and Automorphisms in the Projective Space*. 2013. arXiv:`1305.1193 [cs.IT]`.

[19] T. Feulner. *Canonization of linear codes over $\mathbb{Z}_4$*. URL: `http://codes.uni-bayreuth.de/CanonicalForm/Classification/index.html` (Stand 12. 07. 2013). (siehe auch [20]).

[20] T. Feulner. „Canonization of linear codes over $\mathbb{Z}_4$“. *Advances in Mathematics of Communications*, Bd. 5(2) (2011), S. 245–266.

[21] T. Feulner. „Classification and nonexistence results for linear codes with prescribed minimum distances“. *Designs, Codes and Cryptography*, Bd. 70(1-2) (2014), S. 127–138.

[22] T. Feulner. „Computergestützte Berechnung eines eindeutigen Repräsentanten der semilinearen Isometrieklasse eines fehlerkorrigierenden, linearen Codes und Bestimmung der Automorphismengruppe“. Diplomarbeit. Universität Bayreuth, 2008. 121 S.

[23] T. Feulner. „On canonical forms of ring-linear codes“. *Pre-Proceedings International Workshop on Coding and Cryptography WCC 2013* (Apr. 2013). Hrsg. von L. Budaghyan, T. Helleseth und M. G. Parker, S. 385–396.

[24] T. Feulner. „The automorphism groups of linear codes and canonical representatives of their semilinear isometry classes“. *Advances in Mathematics of Communications*, Bd. 3(4) (2009), S. 363–383.

[25] T. Feulner und G. Nebe. „The automorphism group of an extremal [72, 36, 16] code does not contain $\mathbb{Z}_7$, $\mathbb{Z}_3 \times \mathbb{Z}_3$, or D_{10}“. *IEEE Transactions on Information Theory*, Bd. 58(11) (2012), S. 6916–6924.

[26] T. Feulner, L. Sok, P. Solé und A. Wassermann. „Towards the classification of self-dual bent functions in eight variables“. *Designs, Codes and Cryptography*, Bd. 68(1-3) (2013), S. 395–406.

[27] L. E. Fuller. „A canonical set for matrices over a principal ideal ring modulo m“. *Canadian Journal of Mathematics*, Bd. 7 (1955), S. 54–59.

[28] *GAP – Groups, Algorithms, and Programming, Version 4.6.2*. The GAP Group. 2013. URL: `http://www.gap-system.org`.

[29] *GNU Multiple Precision Arithmetic Library*. Free Software Foundation. URL: `http://gmplib.org/` (Stand 12. 07. 2013).

[30] *GNU Licenses*. GNU Project. URL: `http://www.gnu.org/licenses/` (Stand 02. 01. 2014).

[31] M. Grassl. *Bounds on the minimum distance of linear codes*. URL: `http://www.codetables.de` (Stand 12. 07. 2013).

[32] M. Greferath. „An introduction to ring-linear coding theory“. *Gröbner Bases, Coding, and Cryptography*. Hrsg. von M. Sala, T. Mora, L. Perret, S. Sakata und C. Traverso. Springer, 2009, S. 219–238.

[33] M. Greferath und S. E. Schmidt. „Gray isometries for finite chain rings and a nonlinear ternary $(36, 3^{12}, 15)$ code.“ *IEEE Transactions on Information Theory*, Bd. 45(7) (1999), S. 2522–2524.

[34] J. Grochow. „Matrix Isomorphism of Matrix Lie Algebras“. *Computational Complexity (CCC), 2012 IEEE 27th Annual Conference on*. Juni 2012, S. 203–213.

[35] R. Gugisch. „Konstruktion von Isomorphieklassen orientierter Matroide“. Bayreuther Mathematische Schriften Bd. 72. Dissertation. Universität Bayreuth, 2005. 130 S.

[36] A. R. Hammons Jr., P. V. Kumar, A. R. Calderbank, N. J. A. Sloane und P. Solé. „The $\mathbb{Z}_4$-linearity of Kerdock, Preparata, Goethals, and related codes“. *IEEE Transactions on Information Theory*, Bd. 40(2) (1994), S. 301–319.

[37] T. Honold und I. Landjev. „Linear codes over finite chain rings“. *Electronic Journal of Combinatorics*, Bd. 7 (2000), Research Paper 11.

[38] W. C. Huffman. „On the theory of $\mathbb{F}_q$-linear $\mathbb{F}_{q^t}$-codes“. *Advances in Mathematics of Communications*, Bd. 7(3) (2013), S. 349–378.

[39] W. C. Huffman und V. Pless. *Fundamentals of error-correcting codes*. Cambridge: Cambridge University Press, 2003. 646 S.

[40] M. Jerrum. „A compact representation for permutation groups“. *Journal of Algorithms*, Bd. 7(1) (1986), S. 60–78.

[41] T. Junttila und P. Kaski. „Engineering an efficient canonical labeling tool for large and sparse graphs“. *Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments and the Fourth Workshop on Analytic Algorithms and Combinatorics*. Hrsg. von D. Applegate, G. S. Brodal, D. Panario und R. Sedgewick. SIAM, 2007, S. 135–149.

[42] P. Kaski und P. R. J. Östergård. *Classification algorithms for codes and designs*. Bd. 15. Algorithms and Computation in Mathematics. Berlin: Springer-Verlag, 2006. 412 S.

[43] A. Kerber. *Applied finite group actions*. 2. Aufl. Bd. 19. Algorithms and Combinatorics. Berlin: Springer-Verlag, 1999. 454 S.

[44] M. Kiermaier. „Arcs und Codes über endlichen Kettenringen“. Diplomarbeit. Technische Universität München, 2006.

[45] M. Kiermaier. „Geometrische Konstruktionen linearer Codes über Galois-Ringen der Charakteristik 4 von hoher homogener Minimaldistanz“. Dissertation. Universität Bayreuth, 2012. 95 S.

[46] J.-L. Kim. „A Prize Problem in Coding Theory“. *Gröbner Bases, Coding, and Cryptography*. Hrsg. von M. Sala, T. Mora, L. Perret, S. Sakata und C. Traverso. Springer, 2009, S. 373–377.

[47] R. Koetter und F. Kschischang. „Coding for Errors and Erasures in Random Network Coding“. *IEEE Transactions on Information Theory*, Bd. 54(8) (2008), S. 3579–3591.

[48] A. Kreuzer. „Projektive Hjelmslev Räume“. Beiträge zur Geometrie und Algebra, 16, TUM-Bericht M8806. Dissertation. Technische Universität München, 1988. 88 S.

[49] A. S. Kuzmin und A. A. Nechaev. „Linearly representable codes and the Kerdock code over an arbitrary Galois field of characteristic 2“. *Russian Mathematical Surveys*, Bd. 49(5) (1994), S. 183–184.

[50] R. Laue. „Construction of combinatorial objects – a tutorial“. *Konstruktive Anwendungen von Algebra und Kombinatorik*. Hrsg. von A. Kerber, R. Laue und G. Tinhofer. Bd. 43. Bayreuth. Math. Schr. 1993, S. 53–96.

[51] J. Leon. „Computing automorphism groups of error-correcting codes“. *IEEE Transactions on Information Theory*, Bd. 28(3) (1982), S. 496–511.

[52] J. MacWilliams. „Error-Correcting Codes for Multiple-Level Transmission“. *Bell System Technical Journal*, Bd. 40(1) (1961), S. 281–308.

[53] *Magma V2.19-8*. 2013. URL: `http://magma.maths.usyd.edu.au/magma/`. (siehe auch [6]).

[54] B. R. McDonald. *Finite rings with identity*. Bd. 28. Pure and Applied Mathematics Series. New York: M. Dekker, 1974. 429 S.

[55] B. D. McKay. „Practical graph isomorphism“. *Proceedings of the Tenth Manitoba Conference on Numerical Mathematics and Computing, Vol. I (Winnipeg, Man., 1980)*. Bd. 30. Congressus Numerantium. 1981, S. 45–87.

[56] B. D. McKay und A. Piperno. *Practical graph isomorphism, II*. 2013. arXiv:`1301.1493 [cs.DM]`.

[57] A. A. Nechaev. „Kerdock code in a cyclic form“. *Discrete Mathematics and Applications*, Bd. 1(4) (1991), S. 365–384.

[58] A. A. Nechaev. „Finite Principal Ideal Rings“. *Mathematics of the USSR-Sbornik*, Bd. 20 (1973), S. 364–382.

[59] A. A. Nechaev. „Finite Rings with Applications“. *Handbook of Algebra*. Hrsg. von M. Hazewinkel. Bd. 5. North-Holland, 2008, S. 213 –320.

[60] P. Östergård. „Classifying Subspaces of Hamming Spaces“. *Designs, Codes and Cryptography*, Bd. 27(3) (2002), S. 297–305.

[61] R. Overbeck und N. Sendrier. „Code-based cryptography“. *Post-Quantum Cryptography*. Hrsg. von D. J. Bernstein, J. Buchmann und E. Dahmen. Berlin, Heidelberg: Springer, 2009, S. 95–145.

[62] E. Petrank und R. M. Roth. „Is code equivalence easy to decide?“ *IEEE Transactions on Information Theory*, Bd. 43(5) (1997), S. 1602–1604.

[63] R. Parris und R.C. Read. „A coding procedure for graphs“. *Scientific Report. UWI/CC10. Univ. of West Indies Computer Centre* (1969).

[64] A. Scheerhorn. „Trace- and norm-compatible extensions of finite fields.“ *Applicable Algebra in Engineering, Communication and Computing*, Bd. 3(3) (1992), S. 199–209.

[65] R. Schürer und W. C. Schmid. „MinT: a database for optimal net parameters“. *Monte Carlo and quasi-Monte Carlo methods 2004*. Hrsg. von H. Niederreiter und D. Talay. `http://mint.sbg.ac.at/` (besucht am 12.07.2013). Berlin: Springer, 2006, S. 457–469.

[66] N. Sendrier. „Finding the permutation between equivalent linear codes: the support splitting algorithm“. *IEEE Transactions on Information Theory*, Bd. 46(4) (2000), S. 1193–1203.

[67] N. Sendrier und D. Simos. „How easy is code equivalence over F_q?“ *Pre-Proceedings International Workshop on Coding and Cryptography WCC 2013* (Apr. 2013). Hrsg. von L. Budaghyan, T. Helleseth und M. G. Parker, S. 79–91.

[68] S. Singh und Y. Alkhamees. „Automorphisms of a chain ring“. *Annali di Matematica Pura ed Applicata*, Bd. 186(2) (2007), S. 289–301.

[69] D. Slepian. „Some Further Theory of Group Codes“. *Bell System Technical Journal*, Bd. 39(5) (1960), S. 1219–1252.

[70] W. A. Stein u. a. *Sage Mathematics Software (Version 6.1)*. The Sage Development Team. 2014. URL: `http://www.sagemath.org`.

[71] I. Wegener. *Komplexitätstheorie: Grenzen der Effizienz von Algorithmen*. Berlin: Springer, 2003. 327 S.

[72] J. A. Wood. „Foundations of linear codes defined over finite modules: the extension theorem and the MacWilliams identities“. *Codes Over Rings*. Hrsg. von P. Solé. Bd. 6. Series on Coding Theory and Cryptology. World Scientific, 2009, S. 124–190.

[73] J. Zwanzger. „Computergestützte Suche nach optimalen linearen Codes über endlichen Kettenringen unter Verwendung heuristischer Methoden“. Dissertation. Universität Bayreuth, 2011. 115 S.