Vertizes einfacher Moduln der Symmetrischen und Alternierenden Gruppen

Diplomarbeit

zur Erlangung des akademischen Grades Diplom–Mathematiker

FRIEDRICH–SCHILLER–UNIVERSITÄT JENA
Fakultät für Mathematik und Informatik

eingereicht von René Zimmermann
geb. am 28.04.1976 in Gera

Betreuer: Prof. Dr. B. Külshammer

Jena, 12.04.2000

Zusammenfassung

Die vorliegende Arbeit befasst sich mit den Vertizes der einfachen Moduln der Symmetrischen und Alternierenden Gruppen. Nach einer Einführung in die grundlegenden Begriffe der modularen Darstellungstheorie werden zunächst die Permutationsmoduln sowie die einfachen Moduln der Symmetrischen Gruppen konstruiert. Anschließend werden bekannte Resultate zu Vertizes zusammengefasst sowie neue Ergebnisse zu Vertizes bestimmter Moduln präsentiert. Das Verhalten der Vertizes beim Einschränken eines einfachen Moduls einer Symmetrischen Gruppe auf die Alternierende Gruppe des gleichen Grades wurde ebenfalls untersucht. Schließlich werden Algorithmen zur Vertexberechnung und ihre Implementierung unter GAP vorgestellt.

Danksagung

An dieser Stelle möchte ich allen Personen danken, die mich bei dieser Arbeit unterstützt haben. Besonderer Dank gilt meinem Betreuer Prof. Dr. Külshammer, der mir stets mit guten Ratschlägen zur Seite stand.

Inhaltsverzeichnis

1 Einleitung

Seien F ein Körper der Charakteristik $p > 0$ und G eine endliche Gruppe. In der Darstellungstheorie untersucht man unter anderem die Moduln der Gruppenalgebra FG. Ist p ein Teiler der Gruppenordnung $|G|$, so ist FG nicht halbeinfach. Das bedeutet, dass nicht mehr jeder unzerlegbare Modul einfach ist. Eine wichtige Frage beim Studium der FG–Moduln in diesem Fall ist die nach den Vertizes, insbesondere denen der einfachen Moduln.

Im Fall der Symmetrischen Gruppen $\mathfrak{S}_n$ kennt man zwar Verfahren zur Konstruktion der einfachen Moduln, jedoch sind diese nur für kleine n praktikabel. Insbesondere kennt man keine effizienten Algorithmen zur Bestimmung der Dimensionen der einfachen Moduln. Auch über die Vertizes gibt es nur wenige allgemeine Resultate. Bekannt ist dagegen, dass es beim Einschränken eines einfachen $F\mathfrak{S}_n$–Moduls D auf die Alternierende Gruppe $\mathfrak{A}_n$ nur zwei Möglichkeiten gibt:

- Der eingeschränkte Modul ist ebenfalls einfach.
- Der eingeschränkte Modul zerfällt in die direkte Summe zweier einfacher Moduln.

Im zweiten Fall sind die Vertizes der $F\mathfrak{A}_n$–Moduln zugleich Vertizes von D, insbesondere sind diese dann Untergruppen der $\mathfrak{A}_n$.

Aufgabe meiner Diplomarbeit war es, die folgenden Fragen zu untersuchen:

(1) Wie sehen Vertizes einfacher Moduln der Symmetrischen Gruppen aus?

(2) Wie verhalten sich die Vertizes, wenn man einen einfachen Modul der $\mathfrak{S}_n$ auf die $\mathfrak{A}_n$ einschränkt?

Eine Vermutung, die zunächst bestand, war die folgende: Seien D ein einfacher $F\mathfrak{S}_n$–Modul mit Vertex V und $\bar{D}$ seine Einschränkung auf $\mathfrak{A}_n$. Interessant ist natürlich nur der Fall, in dem $\bar{D}$ wieder einfach ist. Die Vermutung war, dass in diesem Fall $V \cap \mathfrak{A}_n$ ein Vertex von $\bar{D}$ ist. Diese Vermutung in der allgemeinen Form konnte ich durch Gegenbeispiele widerlegen.

Um die erste Frage zu untersuchen, habe ich Algorithmen für die folgenden Probleme entwickelt und unter GAP implementiert:

- Berechnung einfacher $F\mathfrak{S}_n$–Moduln
- Test auf relative Projektivität
- Berechnung des Vertex eines einfachen Moduls

Mit Hilfe dieser Programme habe ich die Vertizes aller einfachen $F\mathfrak{S}_n$– und $F\mathfrak{A}_n$–Moduln für $n \leq 8$ in allen Charakteristiken ermittelt. Dabei habe ich gleich die Vielfachheiten der einfachen Moduln als Kompositionsfaktoren der Permutationsmoduln erhalten.

Daneben konnte ich folgende theoretischen Resultate finden: G. M. Murphy hat in [13] die Vertizes von Spechtmoduln zu Hakenpartitionen untersucht. Aus seinen Resultaten folgt, dass die Vertizes der einfachen Moduln $D^{(n-1,1)}$ in Charakteristik 2 im Fall $n \equiv 1 \ (2)$ konjugiert zu den 2–Sylowgruppen von $\mathfrak{S}_{n-2}$ sind. Dieses Resultat konnte ich verallgemeinern:

Satz. *Seien F ein Körper der Charakteristik $p > 0$ und $n \equiv 1 \ (p)$. Dann sind die p–Sylowgruppen von $\mathfrak{S}_{n-2}$ Vertizes des einfachen $F\mathfrak{S}_n$–Moduls $D^{(n-1,1)}$.*

Weiter folgt aus Murphys Resultaten:

Satz. *Seien F ein Körper der Charakteristik $p > 0$, $p \not| \ n$, $r < p - 1$ und $n \equiv 2r + 1 \ (p)$. Dann sind die p–Sylowgruppen von $\mathfrak{S}_{n-r}$ Vertizes des einfachen $F\mathfrak{S}_n$–Moduls $D^{(n-r,1^r)}$.*

Für $r = 1$ liefert dieser Satz die Vertizes von $D^{(n-1,1)}$ im Fall $p \geq 5$, $n \equiv 3 \ (p)$.

Im Fall $n < p^2$ sind die Vertizes eines einfachen $F\mathfrak{S}_n$–Moduls D^λ gleich den Defektgruppen des zugehörigen Blockes und von der Form $C_p \times \ldots \times C_p$. Dabei ist die Anzahl der Faktoren C_p gleich dem p–Gewicht von λ und leicht kombinatorisch zu bestimmen.

In Beantwortung der zweiten Frage kann ich folgende Resultate präsentieren: Ist $V \subseteq \mathfrak{A}_n$ ein Vertex des einfachen $F\mathfrak{S}_n$–Moduls D, so ist V auch Vertex der Einschränkung von D auf $\mathfrak{A}_n$ bzw. der beiden direkten Summanden dieser Einschränkung. Da im Fall $\mathrm{char}(F)$ ungerade alle Vertizes einfacher $F\mathfrak{S}_n$–Moduln Untergruppen der $\mathfrak{A}_n$ sind, ist die Vermutung in diesem Fall richtig. Im Fall $\mathrm{char}(F) = 2$ dagegen gibt es sowohl $F\mathfrak{S}_n$–Moduln D, für deren Vertizes V gilt: $V \cap \mathfrak{A}_n$ ist Vertex der Einschränkung von D auf $\mathfrak{A}_n$, als auch Moduln, für die das nicht gilt. Für den zweiten Fall kann ich die Beispiele $D^{(5,1)}$, $D^{(4,2)}$ und $D^{(5,2)}$ nennen. Für diese drei Moduln gilt: Jeder Vertex W der Einschränkung auf $\mathfrak{A}_n$ ist eine Untergruppe eines Vertex V des $F\mathfrak{S}_n$–Moduls, dabei ist $|V : W| = 4$. Eine allgemeine Aussage, wann $V \cap \mathfrak{A}_n$ ein Vertex der Einschränkung von D auf $\mathfrak{A}_n$ ist, konnte jedoch nicht formuliert werden.

Schließlich konnten noch zwei Aussagen über die Permutationsmoduln M^λ in Charakteristik p bewiesen werden:

- Ist λ eine p–singuläre Partition von n, so ist die Anzahl der zu jedem einfachen $F\mathfrak{S}_n$–Modul isomorphen Kompositionsfaktoren von M^λ durch p teilbar.

- Sind $p = 2$, λ eine Partition von n, die mindestens ein Teil 2 enthält, und μ die Partition, die man erhält, wenn man in λ ein Teil 2 durch zwei Teile 1 ersetzt, so kommt jeder einfache $F\mathfrak{S}_n$–Modul genau doppelt so oft als Kompositionsfaktor in M^μ vor wie in M^λ.

2 Modulare Darstellungstheorie

Seien F ein Körper und G eine endliche Gruppe.

2.1 Moduln der Gruppenalgebra

2.1 Definition. Die Gruppenalgebra FG ist der F–Vektorraum $\bigoplus_{g\in G} F\cdot g$, auf dem zusätzlich eine Multiplikation definiert wird, die die Gruppenmultiplikation linear fortsetzt:

$$\left(\sum_{g\in G}\lambda_g g\right)\cdot\left(\sum_{h\in G}\mu_h h\right)=\sum_{g,h\in G}\lambda_g\mu_h gh\ .$$

2.2 Definition.

(i) Ein FG–(Links–)Modul besteht aus einer Menge M und zwei Abbildungen

$$+ : M\times M\to M, (x,y)\mapsto x+y$$
$$\cdot : FG\times M\to M, (a,x)\mapsto ax\ ,$$

für die gilt:

(a) $(M,+)$ ist eine abelsche Gruppe

(b) $(a+b)x=ax+bx\ \ (\forall a,b\in FG, x\in M)$

(c) $(ab)x=a(bx)\ \ (\forall a,b\in FG, x\in M)$

(d) $a(x+y)=ax+ay\ \ (\forall a\in FG, x,y\in M)$

(e) $1_G x=x\ \ (\forall x\in M)$.

(ii) Ein Untermodul von M ist eine Teilmenge $N\subseteq M$, die mit den entsprechend eingeschränkten Verknüpfungen einen FG–Modul bildet.

(iii) Sei $N_1\subseteq M$ ein Untermodul. Ein Untermodul $N_2\subseteq M$ heißt direktes Komplement von N_1, wenn gilt: $N_1\cap N_2=0$ und $N_1+N_2=M$. Gegebenenfalls nennt man N_1 und N_2 direkte Summanden von M und schreibt $M=N_1\oplus N_2$.

Bemerkung.

(i) Ein FG–Modul wird zu einem F–Vektorraum, wenn man als Skalarmultiplikation $\alpha m:=(\alpha 1_G)m$ definiert. Unter der Dimension eines FG–Moduls versteht man dann seine Dimension über F.

(ii) Die Gruppenalgebra FG selbst kann als FG–Modul aufgefasst werden, man bezeichnet diesen als regulären FG–Modul.

(iii) Der triviale FG–Modul ist ein eindimensionaler F–Vektorraum, auf dem G trivial operiert.

(iv) Stets sind die trivialen Unterräume, d.h. $0 := \{0\}$ und M, Untermoduln. Ist $M \neq 0$ und sind dies die einzigen Untermoduln, so heißt M einfach.

(v) Stets sind die trivialen Untermoduln auch direkte Summanden. Ist $M \neq 0$ und sind dies die einzigen direkten Summanden, so heißt M unzerlegbar. Ist $M \neq 0$ die direkte Summe einfacher Untermoduln, so nennt man M halbeinfach.

(vi) Ist N ein Untermodul von M, so wird die Menge $\{m + N : m \in M\}$ durch $g(m + N) := gm + N$ ebenfalls zu einem FG–Modul. Dieser heißt Faktormodul M/N.

2.3 Satz (Maschke). *Im Fall* $\mathrm{char}(F) \nmid |G|$ *ist jeder unzerlegbare FG–Modul einfach.*

Beweis. Siehe z.B. Theorem 1.2. in [6, Seite 1].

2.4 Definition. Seien M, N zwei FG–Moduln und $\varphi : M \to N$ eine F–lineare Abbildung.

(i) φ heißt FG–Homomorphismus, wenn für alle $g \in G$, $m \in M$ gilt:

$$\varphi(gm) = g\varphi(m) \ .$$

Die Menge der FG–Homomorphismen zwischen M und N bezeichnet man mit $\mathrm{Hom}_{FG}(M, N)$.

(ii) φ heißt FG–Isomorphismus, wenn es sich um einen bijektiven FG–Homomorphismus handelt.

(iii) Ist $M = N$, so nennt man einen FG–Homomorphismus φ auch FG–Endomorphismus.

Bemerkung.

(i) Zwei FG–Moduln M und N heißen (zueinander) isomorph, wenn es einen FG–Isomorphismus $M \to N$ gibt. Man schreibt dafür $M \cong N$. Ist M zu einem direkten Summanden von N isomorph, so schreibt man dafür $M|N$.

(ii) Die Menge der FG–Endomorphismen eines Moduls M bildet einen Ring, den sogenannten Endomorphismenring $\mathrm{End}_{FG}(M)$.

Eine Folge $M = N_0 \supset N_1 \supset \ldots \supset N_k = 0$ heißt Kompositionsreihe von M, wenn alle Faktormoduln N_i/N_{i+1} einfach sind. Die Moduln N_i/N_{i+1} heißen Kompositionsfaktoren, ihre Anzahl Kompositionslänge von M. Die Kompositionslänge ist eindeutig bestimmt, ebenso die Kompositionsfaktoren bis auf Isomorphie und Reihenfolge. Man sagt, dass ein einfacher Modul N in M vorkommt, wenn ein Kompositionsfaktor von M zu N isomorph ist.

2.5 Satz. *Jeder einfache FG–Modul ist zu einem Kompositionsfaktor von FG isomorph.*

Beweis. Siehe z.B. Theorem 1.1. in [6, Seite 1].

Seien $H \leq G$ und R ein Repräsentantensystem für die Rechtsnebenklassen von H in G. Die Abbildung

$$\mathrm{Tr}_H^G : \mathrm{End}_{FH}(M) \to \mathrm{End}_{FG}(M), \varphi \mapsto \sum_{g \in R} g^{-1}\varphi g$$

nennt man relative Spur.

Seien H eine Untergruppe von G und M ein FG–Modul. Schränkt man die Verknüpfung $G \times M \to M$ auf $H \times M$ ein, so erhält man einen FH–Modul, den sogenannten eingeschränkten Modul $\mathrm{Res}_H^G(M)$. Auch wenn M einfach (unzerlegbar) ist, braucht dieser nicht einfach (unzerlegbar) zu sein. Analog kann man Moduln auch auf Teilkörper von F einschränken.

Seien H eine Untergruppe von G und M ein FH–Modul. Dann ist $FG \otimes_{FH} M$ ein FG–Modul. Man nennt ihn auch induzierten Modul und schreibt manchmal $FG \otimes_{FH} M = \mathrm{Ind}_H^G(M)$. Es gilt:

$$\mathrm{Ind}_H^G(M_1 \oplus M_2) = \mathrm{Ind}_H^G(M_1) \oplus \mathrm{Ind}_H^G(M_2)$$

Analog kann man Moduln auch zu Erweiterungskörpern von F induzieren.

Ein FG–Modul M heißt absolut irreduzibel, wenn er in jedem Erweiterungskörper $E \supseteq F$ irreduzibel ist. Ein Zerfällungskörper des FG–Moduls M ist ein Körper $E \supseteq F$, in dem M eine Kompositionsreihe mit absolut irreduziblen Faktoren besitzt.

2.6 Satz (Schurs Lemma).

(i) Seien M und N zwei einfache nichtisomorphe FG–Moduln. Dann ist $\mathrm{Hom}_{FG}(M, N) = \{0\}$.

(ii) Ist M ein einfacher FG–Modul, so ist $\mathrm{End}_{FG}(M)$ *ein Schiefkörper.*

Beweis. Siehe z.B. Lemma 1.8.1. in [4, Seite 23].

Bemerkung. Ist M absolut irreduzibel, so gilt sogar: $\mathrm{End}_{FG}(M) \cong F$.

2.2 Moduln und Darstellungen

Eine Darstellung von G auf einem F–Vektorraum V ist ein Homomorphismus $\Delta : G \to \mathrm{GL}(V)$. Darstellungen von G und FG–Moduln hängen eng miteinander zusammen. Zu jedem FG–Modul M erhält man eine Darstellung von G auf M durch $\Delta_M(g) : m \mapsto gm$. Umgekehrt wird der F–Vektorraum V durch eine Darstellung Δ von G zu einem FG–Modul, wenn man definiert: $gm := (\Delta(g))(m)$.

Zwei Darstellungen Δ und Δ' von G auf V bzw. V' heißen äquivalent, wenn es einen Isomorphismus $f : V \to V'$ gibt, so dass gilt:

$$\Delta'(g) \circ f = f \circ \Delta(g) \quad (\forall g \in G)$$

Zu äquivalenten Darstellungen gehören isomorphe Moduln.

Eine Darstellung von G auf V heißt irreduzibel, wenn 0 und V die einzigen Δ–invarianten Unterräume von V sind. Zu einfachen Moduln gehören irreduzible Darstellungen.

Ebenso wie man Einschränkung und Induktion von Moduln erklärt, kann man auch eingeschränkte und induzierte Darstellungen definieren. Auch hier entsprechen sich die Begriffe.

2.3 Vertizes

2.7 Definition. Sei $H \leq G$. Ein FG–Modul M heißt (relativ) H–projektiv, wenn es einen FH–Modul Q gibt, so dass M zu einem direkten Summanden von $FG \otimes_{FH} Q$ isomorph ist.

Bemerkung. Dies ist eine Erweiterung des üblichen Projektivitätsbegriffs, ein FG–Modul ist genau dann projektiv, wenn er relativ 1–projektiv.

Sei im folgenden M stets unzerlegbar.

2.8 Bemerkung.

(i) Sei $\mathfrak{P} := \{H \leq G : M \text{ ist } H\text{–projektiv}\}$. Diese Menge hat folgende Eigenschaften:

  (a) Seien $H \in \mathfrak{P}$ und $K \leq G$ mit $H \leq K$. Dann ist auch $K \in \mathfrak{P}$.

  (b) Sei $H \in \mathfrak{P}$. Dann gehört auch jede zu H konjugierte Untergruppe von G zu $\mathfrak{P}$.

(ii) $\mathfrak{P}$ bildet eine durch die Untergruppenrelation halbgeordnete Menge. Die minimalen Elemente heißen Vertizes von M. Alle Vertizes sind zueinander konjugiert. Die Menge der Vertizes bezeichnet man mit $\mathrm{vx}(M)$.

2.9 Satz. *Sei M ein unzerlegbarer FG–Modul mit Vertex V. Dann gibt es einen unzerlegbaren FV–Modul Q mit folgenden Eigenschaften:*

(a) $Q | \operatorname{Res}_V^G(M)$

(b) $M | FG \otimes_{FV} Q$

(c) $\operatorname{vx}(Q) = V$.

Beweis. Siehe z.B. Theorem 3.6. in [14, Seite 271].

Definition. In diesem Fall nennt man Q eine Quelle von M.

In Charakteristik 0 sind Vertizes uninteressant, wegen den Sätzen 2.3 und 2.5 sind die Vertizes dann stets 1. Ist dagegen $\operatorname{char}(F) = p > 0$, so sind Vertizes stets p–Gruppen. Weitere Eigenschaften von Vertizes sind in Kapitel 4 zusammengestellt.

2.4 Partitionen

Eine Partition von $n \in \mathbb{N}$ ist eine (endliche) Folge $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ natürlicher Zahlen mit $\sum_{j=1}^{\infty} \lambda_j = n$ und $\lambda_j \geq \lambda_{j+1} > 0 \ \ (\forall j \in \{1, \ldots, k-1\})$. Partitionen kürzt man wie im folgenden Beispiel ersichtlich ab: Für $(6, 4, 4, 4, 1, 1)$ schreibt man $(6, 4^3, 1^2)$.

Die λ_j nennt man Teile von λ. Eine Partition heißt p–singulär, wenn sie p gleiche Teile hat, andernfalls nennt man sie p–regulär. Obige Partition ist z.B. singulär für $p \leq 3$, dagegen regulär für $p > 3$.

Auf der Menge der Partitionen wird eine Halbordnung, die sogenannte Dominanzordnung, definiert:

$$\lambda \trianglelefteq \mu :\Leftrightarrow \sum_{j=1}^{k} \lambda_j \leq \sum_{j=1}^{k} \mu_j \ \ (\forall k \in \mathbb{N})$$

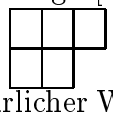
Ist zusätzlich $\lambda \neq \mu$, schreibt man auch $\lambda \triangleleft \mu$. Außerdem verwendet man die lexikographische Ordnung:

$$\lambda < \mu :\Leftrightarrow \exists k \in \mathbb{N} : \lambda_j = \mu_j \, (\forall j < k) \wedge \lambda_k < \mu_k$$

Ist $\lambda < \mu \ \ \vee \ \ \lambda = \mu$, so schreibt man auch $\lambda \leq \mu$. Es gilt: $\lambda \trianglelefteq \mu \Rightarrow \lambda \leq \mu$.

Den Elementen von $\mathfrak{S}_n$ ordnet man Partitionen zu: Hat π in Zyklenschreibweise die Form $\pi = (a_1^1, \ldots, a_{k_1}^1) \ldots (a_1^j, \ldots, a_{k_j}^j)$, wobei ggfs. auch Fixpunkte (1–Zyklen) mit angegeben sind, um $\sum_{l=1}^{j} k_l = n$ zu erreichen, und die Zyklen nach fallender Länge geordnet sind, dann sagt man: π hat den Typ $(k_1, \ldots, k_j)$. Zum Beispiel haben Transpositionen den Typ $(2, 1^{n-2})$ und n–Zyklen den Typ (n). Zwei Elemente sind genau dann konjugiert, wenn sie den gleichen Typ haben.

Zu jeder Partition $\lambda = (\lambda_1, \ldots \lambda_k)$ von n definiert man ferner eine Young–Gruppe: $\mathfrak{S}_\lambda := \mathfrak{S}_{\lambda_1} \times \cdots \times \mathfrak{S}_{\lambda_k} \leq \mathfrak{S}_n$.

Partitionen veranschaulicht man sich durch Diagramme. Das Diagramm von λ ist die Menge $[\lambda] = \{(i,j) \in \mathbb{N}^2 : 1 \leq j \leq \lambda_i\}$. Schreibweise am Beispiel $\lambda = (3,2)$: 

In natürlicher Weise spricht man von den Zeilen und Spalten eines Diagramms. Ein Haken eines Diagramms ist die Menge $H(i,j) := \{(k,j) : k \geq i\} \cup \{(i,l) : l \geq j\}$. Verschiebt man die Elemente des Hakens soweit wie möglich diagonal nach rechts unten, so erhält man den zugehörigen Randhaken. Den p–Kern einer Partition erhält man, indem man im zugehörigen Diagramm solange wie möglich Randhaken der Länge p streicht (das Ergebnis ist unabhängig von der Reihenfolge des Streichens). Die Anzahl der gestrichenen Randhaken nennt man p–Gewicht von λ.

Eine Abbildung $t : [\lambda] \to \{1, \ldots, n\}$ nennt man λ–Tableau. Dafür schreibt man dann z.B.:

3	1	4
2	5	

$\mathfrak{S}_n$ operiert von links auf der Menge der λ–Tableaus in offensichtlicher Weise. Ein λ–Tableau heißt Zeilenstandardtableau, falls die Einträge in jeder Zeile von links nach rechts wachsen. Es heißt Standardtableau, wenn die Einträge zusätzlich in den Spalten von oben nach unten wachsen.

Zwei λ–Tableaus t_1 und t_2 heißen zeilenäquivalent (spaltenäquivalent), wenn für alle $i \in \{1, \ldots, n\}$ gilt: i steht in t_1 und t_2 in der gleichen Zeile (Spalte). Das sind Äquivalenzrelationen. Die Klassen zeilenäquivalenter Tableaus nennt man λ–Tabloide. Schreibweise:

$$\{t\} = \begin{array}{|ccc|}\hline 1 & 3 & 4 \\ \hline 2 & 5 & \\ \cline{1-2}\end{array}$$

In jedem Tabloid gibt es genau ein Zeilenstandardtableau. Auf der Menge der Tabloide wird eine Operation von $\mathfrak{S}_n$ wie folgt definiert: Sei t ein Zeilenstandardtableau und $\sigma \in \mathfrak{S}_n$. Dann setzt man $\sigma\{t\} := \{\sigma t\}$.

Der Spaltenstabilisator eines Tableaus ist die Menge

$$C_t := \{\sigma \in \mathfrak{S} : \sigma t \text{ ist spaltenäquivalent zu } t\}.$$

Zu einem Zeilenstandardtableau t definiert man das Polytabloid $e_t := \sum_{\sigma \in C_t} (\operatorname{sgn} \sigma)\{\sigma t\}$. Ein Polytabloid e_t heißt Standardpolytabloid, wenn t ein Standardtableau ist.

2.5 $F\mathfrak{S}_n$– und $F\mathfrak{A}_n$–Moduln

2.10 Definition. Seien λ eine Partition von n und T der triviale $F\mathfrak{S}_\lambda$–Modul. Den $F\mathfrak{S}_n$–Modul $M^\lambda := F\mathfrak{S}_n \otimes_{F\mathfrak{S}_\lambda} T$ bezeichnet man als Permutationsmodul zur Partition λ.

Bemerkung. Eine explizite Realisierung von M^λ erhält man, wenn man die λ–Tabloide als Basis eines F–Vektorraums verwendet und die Operation von $\mathfrak{S}_n$ auf den Tabloiden linear fortsetzt.

2.11 Definition. Den Untermodul S^λ von M^λ, der von den λ–Polytabloiden aufgespannt wird, nennt man Spechtmodul.

Bemerkung.

(i) Nicht alle Polytabloide sind linear unabhängig über F. Die Standardpolytabloide bilden eine Basis von S^λ.

(ii) Im Fall $\mathrm{char}(F) = 0$ bilden die Spechtmoduln ein Repräsentantensystem für die Isomorphietypen einfacher $F\mathfrak{S}_n$–Moduln. Ist dagegen $\mathrm{char}(F) = p > 0$, so sind in der Regel nicht mehr alle Spechtmoduln einfach. Sie bleiben jedoch unzerlegbar. Man erhält dann die einfachen Moduln wie folgt: Sei λ eine p–reguläre Partition. Ist S^λ einfach, so setzt man $D^\lambda := S^\lambda$. Andernfalls gibt es einen eindeutig bestimmten maximalen Untermodul. Der zugehörige Faktormodul bekommt dann die Bezeichnung D^λ (zu den p–singulären Partitionen wird kein D^λ erklärt). Die so definierten D^λ bilden ein Repräsentantensystem für die Isomorphietypen einfacher $F\mathfrak{S}_n$–Moduln.

(iii) D^λ kommt in S^λ genau einmal vor. Alle anderen Kompositionsfaktoren sind zu D^μ's mit $\mu \rhd \lambda$ isomorph.

(iv) Es gilt: D^μ ist Kompositionsfaktor von $M^\lambda \Leftrightarrow \mu \trianglerighteq \lambda$.

2.12 Bemerkung.

(i) Jeder Körper ist Zerfällungskörper der $F\mathfrak{S}_n$–Moduln.

(ii) Die Einschränkung eines einfachen $F\mathfrak{S}_n$–Moduls M auf $\mathfrak{A}_n$ ist entweder absolut irreduzibel oder zerfällt in einem geeigneten Erweiterungskörper F' von F in die direkte Summe zweier einfacher $F'\mathfrak{A}_n$–Moduln M_1, M_2. In diesem Fall sind die Vertizes von M_1 und M_2 gleich den Vertizes von M; insbesondere sind die Vertizes von M Untergruppen von $\mathfrak{A}_n$.

2.6 Blöcke und Defektgruppen

Die Gruppenalgebra FG kann auch als $F(G \times G)$–Modul aufgefasst werden. Zunächst operiert $(G \times G)$ auf G durch $(g,h)x := gxh^{-1}$. Daraus erhält man durch lineare Fortsetzung eine Operation von $G \times G$ auf FG. Setzt man nun noch $(\lambda(g,h))x := \lambda((g,h)x)$, so erhält man eine $F(G \times G)$–Modulstruktur.

G kann durch die folgende Abbildung in $G \times G$ eingebettet werden:

$$\delta : G \to G \times G, g \mapsto (g, g) \ .$$

Die unzerlegbaren direkten Summanden von FG als $F(G \times G)$–Modul nennt man die Blöcke von FG. Zu jedem Block $\mathbb{B}$ gibt es eine Untergruppe $P \leq G$, so dass $\delta(P)$ ein Vertex von $\mathbb{B}$ ist. P heißt dann Defektgruppe von $\mathbb{B}$.

Betrachtet man die Blöcke wieder als FG–Linksmoduln, so sind sie projektiv. Kompositionsfaktoren verschiedener Blöcke sind nicht isomorph. Daher kann man davon sprechen dass ein einfacher Modul zu einem Block gehört. Eine andere Charakterisierung der Blöcke ist die folgende: Zwei einfache Moduln M und N sind miteinander verbunden, wenn es eine Folge unzerlegbarer FG–Moduln $M = M_1, M_2, \ldots, M_k = N$ gibt, so dass für jedes $i \in \{1, \ldots, k-1\}$ gilt: M_i und M_{i+1} besitzen einen gemeinsamen Kompositionsfaktor. Zwei einfache Moduln sind genau dann verbunden, wenn sie zum gleichen Block gehören. Daher gehören alle Kompositionsfaktoren eines unzerlegbaren Moduls M zum gleichen Block und man kann sagen, dass M zu diesem Block gehört.

Sei M ein unzerlegbarer Modul des Blocks $\mathbb{B}$. Zu jedem Vertex V von M gibt es eine Defektgruppe P von $\mathbb{B}$, so dass $V \leq P$ gilt. Zu jedem Block $\mathbb{B}$ gibt es einen einfachen Modul, dessen Vertizes gleich den Defektgruppen sind.

Der Block, zu dem der triviale FG–Modul gehört, heißt Hauptblock. Seine Defektgruppen sind die p–Sylowgruppen von G.

Die Zuordnung der einfachen $F\mathfrak{S}_n$–Moduln zu den Blöcken ist mit Hilfe des folgenden Satzes leicht möglich:

2.13 Satz (Nakayamas Vermutung). *Seien F ein Körper der Charakteristik $p > 0$ und $n \in \mathbb{N}$. Die einfachen $F\mathfrak{S}_n$–Moduln D^λ und D^μ gehören genau dann zum gleichen Block, wenn λ und μ den gleichen p–Kern besitzen.*

Beweis. Siehe z.B. [10].

2.14 Bemerkung. Seien F ein Körper der Charakteristik $p > 0$ und $n \in \mathbb{N}$ mit $p \not| \, n$. Der Spechtmodul $S^{(n-r,1^r)}$ ist im Fall $r < p$ einfach und damit gleich $D^{(n-r,1^r)}$. Begründung: Der p–Kern von $(n-r, 1^r)$ ist die Partition $(k, 1^r)$, wobei $k \in \{1, \ldots, n\}$ mit $k \equiv n - r \ \ (p)$ ist. Angenommen, der einfache Modul D^λ ist isomorph zu einem Kompositionsfaktor von $S^{(n-r,1^r)}$, dann ist der p–Kern von λ ebenfalls $(k, 1^r)$. Also muss λ mindestens $r + 1$ Teile haben. Dann ist aber $(n - r, 1^r) \trianglerighteq \lambda$, im Widerspruch zu Bemerkung 2.5 (iii).

3 Permutationsmoduln

Sei F ein Körper der Charakteristik $p > 0$.

3.1 Berechnung der Permutationsmoduln und ihrer Kompositionsfaktoren

Bevor man die Vertizes einfacher Moduln berechnen kann, müssen diese erst einmal konstruiert werden. Ein Repräsentantensystem für die Isomorphieklassen einfacher $F\mathfrak{S}_n$–Moduln erhält man, indem man den regulären $F\mathfrak{S}_n$–Modul in Kompositionsfaktoren zerlegt. Auf diese Weise ist es jedoch schwierig, die einfachen Moduln den p–regulären Partitionen von n zuzuordnen. Ich habe daher alle Permutationsmoduln berechnet und mir aus ihren Kompositionsfaktoren die entsprechenden einfachen Moduln herausgesucht. Auf diese Weise habe ich gleich noch die Vielfachheiten der Kompositionsfaktoren in den Permutationsmoduln erhalten. Ich habe eine Funktion KomposPerm programmiert, die bei Eingabe von λ, n und p die Kompositionsfaktoren von M^λ mit ihren Vielfachheiten berechnet.

Als nächstes ging es darum, die Kompositionsfaktoren den Partitionen zuzuordnen. Trivialerweise ist $D^{(n)} = M^{(n)}$. Arbeitet man dann die Permutationsmoduln in lexikographischer Ordnung der zugehörigen Partitionen ab, so kommt bei jedem Modul maximal ein neuer Isomorphietyp von Kompositionsfaktoren hinzu. Dieser gehört dann zur gleichen Partition wie der Permutationsmodul, in dem er erstmalig auftritt. Alle anderen Faktoren sind isomorph zu einem bereits berechneten D^λ. Ob zwei einfache Moduln isomorph sind, lässt sich mit der GAP-Funktion MTX.IsomorphismKomposPerm überprüfen.

Es ist sehr umständlich, dieses Verfahren zu programmieren, deshalb habe ich die Zuordnung "von Hand" gemacht. Auf diese Weise habe ich die Vielfachheiten der Kompositionsfaktoren der Permutationsmoduln bis $n = 7$ erhalten. Bei der $\mathfrak{S}_7$ in Charakteristik 5 und 7 traten erstmals Speicherprobleme auf: Die größten Permutationsmoduln konnten nicht mehr berechnet werden. Dennoch hatte ich nun eine Reihe einfacher Moduln, mit denen ich weiterarbeiten konnte.

Der Quellcode von KomposPerm sowie die damit berechneten Vielfachheiten der einfachen Moduln in den Permutationsmoduln sind im Anhang aufgeführt. In diesem Kapitel folgen noch zwei Sätze über diese Vielfachheiten.

3.2 Sätze zu Permutationsmoduln

Beim Betrachten der erhaltenen Tabellen fällt zunächst auf, dass die Vielfachheiten der Kompositionsfaktoren in Permutationsmoduln zu singulären Partitionen stets durch p teilbar sind.

Lemma. *Der reguläre FC_p–Modul besitzt genau eine Kompositionsreihe*

$FC_p \supset M_1 \supset M_2 \supset \ldots \supset M_{p-1} \supset 0$, dabei sind alle p Kompositionsfaktoren isomorph zum trivialen FC_p–Modul.

Beweis. Nach Lemma 7.25 in [12, Seite 101] sind alle einfachen FC_p–Moduln isomorph zum trivialen FC_p–Modul. Aus Satz 7.27 in [12, Seite 103] folgt dann die Behauptung.

3.1 Satz. *Sei λ eine p–singuläre Partition von n. Dann kommt jeder einfache $F\mathfrak{S}_n$–Modul in M^λ in durch p teilbarer Anzahl vor.*

Beweis. Es ist $M^\lambda = F\mathfrak{S}_n \otimes_{FH} T$ mit $H = \mathfrak{S}_k \times \ldots \times \mathfrak{S}_k \times H'$ (p-mal $\mathfrak{S}_k$), $H' \leq \mathfrak{S}_{n-kp}$ und T als trivialem FH–Modul. Sei
$K' := \langle (1, k+1, \ldots, (p-1)k+1)(2, k+2, \ldots, (p-1)k+2) \ldots (k, 2k, \ldots, pk) \rangle$
(diese Gruppe vertauscht gerade die Untergruppen $\mathfrak{S}_k < H$ zyklisch). Dann ist $K := H \times K'$ eine Untergruppe von $\mathfrak{S}_n$ mit $H \leq K \leq \mathfrak{S}_n$. Also ist $F\mathfrak{S}_n \otimes_{FH} T = F\mathfrak{S}_n \otimes_{FK} (FK \otimes_{FH} T)$. Wegen $K/H \cong C_p$ ist $FK \otimes_{FH} T$ isomorph zum regulären FC_p–Modul. Dieser besitzt nach obigem Lemma p–mal den trivialen Modul als Kompositionsfaktoren. Demzufolge muss beim weiteren Induzieren zu $\mathfrak{S}_n$ jeder Kompositionsfaktor in durch p teilbarer Anzahl vorkommen.

Bemerkung. Die Umkehrung ist trivial, denn ist λ nicht p–singulär, so kommt gerade D^λ in M^λ mit Vielfachheit 1 vor.

3.2 Satz. *Seien F ein Körper der Charakteristik 2, λ eine Partition von n mit mindestens einem Teil 2 und λ' die Partition, die man erhält, wenn man in λ ein Teil 2 durch zwei Teile 1 ersetzt. Dann kommt jeder einfache $F\mathfrak{S}_n$–Modul in $M^{\lambda'}$ genau doppelt so oft vor wie in M^λ.*

Beweis. Es ist $M^\lambda = FG \otimes_{F\mathfrak{S}_\lambda} T$ mit T als trivialem $F\mathfrak{S}_\lambda$–Modul und $M^{\lambda'} = FG \otimes_{F\mathfrak{S}_{\lambda'}} T' = FG \otimes_{F\mathfrak{S}_\lambda} (F\mathfrak{S}_\lambda \otimes_{F\mathfrak{S}_{\lambda'}} T')$ mit T' als trivialem $F\mathfrak{S}_{\lambda'}$–Modul. Wegen $\mathfrak{S}_\lambda / \mathfrak{S}_{\lambda'} \cong C_2$ ist $F\mathfrak{S}_\lambda \otimes_{F\mathfrak{S}_{\lambda'}} T'$ isomorph zum regulären FC_2–Modul. Dieser besitzt zwei zu T isomorphe Kompositionsfaktoren, also besitzt $M^{\lambda'}$ die gleichen Kompositionsfaktoren wie M^λ, nur mit doppelter Vielfachheit.

Bemerkung. Einen analogen Satz für $\mathrm{char}(F) > 2$ gibt es nicht, wie ein Blick auf die Tabellen in Anlage C zeigt.

4 Vertizes einfacher Moduln

Es seien F ein Körper der Charakteristik $p > 0$, G eine endliche Gruppe und D ein einfacher FG–Modul.

4.1 Allgemeine Resultate

Die Ausführungen zu den Vertizes beginnen mit bekannten Resultaten, welche die Kandidaten für die Vertizes eines einfachen Moduls weiter einschränken. Zunächst sei noch einmal daran erinnert, dass Vertizes und Defektgruppen stets p–Gruppen sind.

4.1 Satz. *Seien $P \in \mathrm{Syl}_p(G)$, V ein Vertex von D. Dann ist $|P : V|$ ein Teiler von $\dim_F(D)$.*

Beweis. Siehe z.B. Theorem 4.7.5 in [14, Seite 293].

Bemerkung.

(i) Mit diesem Satz erhält man eine untere Schranke für die Ordnung des Vertex: Sei p^k die höchste p–Potenz, die $\dim_F(D)$ teilt. Dann ist $|V| \geq \frac{|P|}{p^k}$.

(ii) Eine unmittelbare Folgerung daraus ist: Die Vertizes eindimensionaler Moduln sind die p–Sylowgruppen von G.

4.2 Satz. *Sei V ein Vertex von D. Dann gibt es eine Defektgruppe P des Blocks von D mit $C_G(P) \leq V \leq P$.*

Beweis. Siehe z.B. Korollar 9.7. in [9, Teil III].

Bemerkung. Insbesondere ist bei abelscher Defektgruppe der Vertex gleich der Defektgruppe.

4.3 Satz (Erdmann). *Ist der Vertex von D zyklisch, so ist er gleich der Defektgruppe des Blocks von D.*

Beweis. Siehe z.B. Korollar 9.9. in [9, Teil III].

Bemerkung. Dieser Satz liefert untere Schranken für Vertizes: Ist die Defektgruppe zyklisch, so ist der Vertex gleich der Defektgruppe, andernfalls kann er nicht zyklisch sein.

Der folgende Satz eignet sich gut als Projektivitätstest, er bildet die Grundlage für den in 5.1.5 beschriebenen Algorithmus.

4.4 Satz (Higman-Kriterium). *Seien $H \leq G$ und M ein unzerlegbarer FG-Modul. Folgende Aussagen sind äquivalent:*

(1) M ist H-projektiv.

(2) Es existiert ein $\vartheta \in \mathrm{End}_{FH}(M)$ mit $\mathrm{Tr}_H^G(\vartheta) = \mathrm{id}_M$.

Beweis. Siehe z.B. Theorem 2.2. in [14, Seite 265].

4.2 Vertizes einfacher $F\mathfrak{S}_n$–Moduln

Sei jetzt $G := \mathfrak{S}_n$.

4.5 Satz. *Sei λ eine p–reguläre Partition von n mit p-Gewicht b. Dann sind die Defektgruppen des Blocks von D^λ konjugiert zu* $\mathrm{Syl}_p(\mathfrak{S}_{(pb)!})$.

Beweis. Siehe z.B. Theorem 2.5. in [13, Seite 87].

Bemerkung. Nach Satz 2.13 gehören zu den einfachen Moduln eines Blockes Partitionen, die alle den gleichen p–Kern besitzen. Daher haben diese Moduln auch alle den gleichen p–Defekt und diese Charakterisierung der Defektgruppen ist wohldefiniert.

$L_p(r)$ sei die Anzahl der Stellen von r in der Darstellung zur Basis p. Es gilt also: $p^{L_p(r)-1} \leq r < p^{L_p(r)}$.

4.6 Satz. *Seien p eine Primzahl, $F := \mathbb{F}_p$, $\lambda = (n-r, 1^r)$. Wenn $n \equiv 2r+1 \ (p^{L_p(r)})$, $p \nmid n$ und $p \nmid n-r$ ist, dann sind die p–Sylowgruppen von $\mathfrak{S}_r \times \mathfrak{S}_{n-r}$ Vertizes des Spechtmoduls S^λ.*

Beweis. Siehe z.B. Theorem 3.2 in [13, Seite 89].

Bemerkung. Ist zusätzlich $r < p$ so ist λ eine p–reguläre Partition und nach Bemerkung 2.14 ist $D^\lambda = S^\lambda$. Außerdem sind in diesem Fall die p–Sylowgruppen von $\mathfrak{S}_r$ trivial.

Die zahlreichen Voraussetzungen für diesen Fall reduzieren sich noch etwas: Wegen $r < p$ ist zunächst $L_p(r) = 1$. Wäre $p = r+1$, so wäre wegen $n \equiv 2r+1 \ (p)$ auch $n \equiv r \ (p)$, im Widerspruch zu $p \nmid n-r$. Also ist sogar $p > r+1$. Dann aber folgt aus $n \equiv 2r+1 \ (p)$ bereits $p \nmid n-r$. Man erhält also den folgenden Satz:

4.7 Satz. *Seien F ein Körper der Charakteristik $p > 0$, $p \nmid n$, $r < p-1$ und $n \equiv 2r+1 \ (p)$. Dann sind die p–Sylowgruppen von $\mathfrak{S}_{n-r}$ Vertizes des einfachen $F\mathfrak{S}_n$–Moduls $D^{(n-r,1^r)}$.*

Beispiel. Seien $p = 5$, $n = 8$, $r = 1$. Da alle Voraussetzungen erfüllt sind, ist der Vertex von $D^{(7,1)}$ eine 5-Sylowgruppe von $\mathfrak{S}_7$, also vom Typ C_5.

4.8 Satz. *Seien F ein Körper der Charakteristik 2 sowie n und r ungerade. Die Vertizes von $S^{(n-r,1^r)}$ sind gleich den Defektgruppen des zugehörigen Blockes.*

Beweis. Siehe Theorem 4.6 in [13, Seite 95].

Bemerkung. Interessant ist der Fall $r = 1$. Nach Bemerkung 2.14 ist $S^{(n-1,1)} = D^{(n-1,1)}$. Nach Satz 4.5 sind also die 2–Sylowgruppen von $\mathfrak{S}_{n-3}$ Vertizes von $D^{(n-1,1)}$.

4.9 Satz. *Seien F ein Körper der Charakteristik p, $n < p^2$, $\mathbb{B}$ ein Block von $F\mathfrak{S}_n$ und D ein einfacher Modul in $\mathbb{B}$. Dann sind die Defektgruppen von $\mathbb{B}$ Vertizes von D.*

Beweis. Sei $n = kp + r$ mit $k, r \in \mathbb{Z}, 0 \leq r < p$. Dann ist wegen $n < p^2$ auch $k < p$. Die Ordnung der p–Sylowgruppen von $\mathfrak{S}_n$ ist dann p^k. Andererseits ist $\langle (1, 2, \ldots, p), (p+1, \ldots, 2p), \ldots, ((k-1)p+1, \ldots, kp) \rangle$ eine Untergruppe von $\mathfrak{S}_n$ mit Ordnung p^k. Es muss sich also um eine p–Sylowgruppe handeln. Da diese Gruppe abelsch ist, sind auch alle Defektgruppen abelsch. Dann sind aber nach Satz 4.2 die Vertizes gleich den Defektgruppen.

4.10 Satz. *Seien M_1 und M_2 zwei unzerlegbare FG–Moduln, M ein direkter Summand von $M_1 \otimes M_2$. Dann existieren Vertizes V, V_1 und V_2 von M, M_1 bzw. M_2, so dass gilt: $V = V_1 \cap V_2$.*

Bemerkung. Im Fall $\mathrm{char}(F) \neq 2$ gibt es neben dem trivialen Modul einen weiteren eindimensionalen Modul V. $\mathfrak{S}_n$ operiert auf W durch $\sigma x = \mathrm{sgn}(\sigma)x$. Die Vertizes dieses Moduls sind nach Satz 4.1 die p–Sylowgruppen von $\mathfrak{S}_n$.

Sei D^λ ein einfacher $F\mathfrak{S}_n$–Modul mit Vertex V. Bildet man $W \otimes_{F\mathfrak{S}_n} D^\lambda$, so erhält man wieder einen einfachen Modul. Dieser hat nach obigem Satz ebenfalls V als Vertex. Die (bewiesene) Mullineux–Vermutung gibt Auskunft darüber, zu welcher Partition dieser Modul gehört (mehr dazu z.B. in [2]).

4.3 Vertizes eingeschränkter Moduln

4.11 Satz. *Seien $H \leq G$ und M ein unzerlegbarer FG–Modul mit Vertex V.*

(i) Sind N ein direkter Summand von $\mathrm{Res}^G_H(M)$ und W ein Vertex von N, so ist W zu einer Untergruppe von V konjugiert.

(ii) Sei N ein unzerlegbarer FH–Modul mit Vertex W, für den gilt: $M \mid FG \otimes_{FH} N$. Dann ist V zu einer Untergruppe von W konjugiert.

Beweis. Siehe z.B. Lemma 3.4 in [14, Seite 270].

4.12 Satz. *Seien $H \leq G$ und M ein unzerlegbarer FG–Modul mit Vertex $V \leq H$. Dann gibt es einen direkten Summanden von $\mathrm{Res}^G_H(M)$ mit Vertex V.*

Beweis. Siehe z.B. Theorem 3.3 in [9, Seite 104].

Bemerkung. Im Fall $p \nmid |G : H|$ $(p = \mathrm{char}(F) > 0)$ gibt es eine p–Sylowgruppe von G, die Untergruppe von H ist. Daher hat jeder unzerlegbare FG–Modul einen Vertex, der Untergruppe von H ist und es gilt: Mindestens einer der direkten Summanden von $\mathrm{Res}^G_H(M)$ besitzt einen Vertex, der zugleich Vertex von M ist.

4.4 Einschränkung der einfachen $F\mathfrak{S}_n$-Moduln auf $\mathfrak{A}_n$

Eine der eingangs gestellten Fragen war die folgende:

Vermutung. Seien D ein einfacher $F\mathfrak{S}_n$–Modul und $\bar{D}$ seine (absolut irreduzible) Einschränkung auf $\mathfrak{A}_n$. Seien weiter V ein Vertex von D und W ein Vertex von $\bar{D}$. Gilt dann: W ist konjugiert zu $V \cap \mathfrak{A}_n$? Wenn das nicht allgemein richtig ist: Kann man Kriterien dafür angeben, wann das gilt?

Die allgemeine Vermutung erwies sich als falsch, wie die folgenden Gegenbeispiele zeigen:

4.13 Beispiel.

(i) Die Vertizes von $D^{(5,1)}$ in Charakteristik 2 sind die 2–Sylowgruppen von $\mathfrak{S}_6$, also z.B. $\langle (1,2),(1,3)(2,4),(5,6)\rangle =: V$ (Ordnung 16). Weiter ist $V \cap \mathfrak{A}_6 = \langle (1,3)(2,4),(1,2)(5,6)\rangle$ (Ordnung 8). Jedoch sind die Vertizes von $\mathrm{Res}^{\mathfrak{S}_6}_{\mathfrak{A}_6}(D^{(5,1)})$ vom Typ $\langle (1,2)(3,4),(3,4)(5,6)\rangle$ (Ordnung 4).

(ii) Die Vertizes von $D^{(4,2)}$ und $D^{(4,2,1)}$ in Charakteristik 2 sind ebenfalls die 2–Sylowgruppen von $\mathfrak{S}_6$. Jedoch sind die Vertizes der Einschränkungen dieser Moduln auf $\mathfrak{A}_6$ bzw. $\mathfrak{A}_7$ vom Typ V_4.

4.14 Bemerkung. Die Vermutung ist trotzdem in vielen Fällen richtig:

(i) Sind die Vertizes von D Untergruppen von $\mathfrak{A}_n$, so ist die Vermutung nach Satz 4.12 richtig.

(ii) Ist $\mathrm{char}(F) > 2$, so sind nach Bemerkung 4.12 die Vertizes einfacher $F\mathfrak{S}_n$–Moduln Untergruppen von $\mathfrak{A}_n$, also gilt die Vermutung für $\mathrm{char}(F) > 2$.

(iii) Die Einschränkung des trivialen $F\mathfrak{S}_n$–Moduls $D^{(n)}$ ist der triviale $F\mathfrak{A}_n$–Modul. Seine Vertizes sind nach Bemerkung 4.1 die p–Sylowgruppen von $\mathfrak{A}_n$, welche wiederum Durchschnitt von $\mathfrak{A}_n$ mit einer p-Sylowgruppe von $\mathfrak{S}_n$, also einem Vertex von $D^{(n)}$, sind.

(iv) Mit Ausnahme der Gegenbeispiele von Bemerkung 4.13 ist die Vermutung für alle von mir untersuchten Moduln richtig.

4.5 Vertizes einfacher Moduln zu Hakenpartitionen

Eine Partition der Form $(n-r,1^r)$ nennt man Hakenpartition.

4.15 Satz. *Der Permutationsmodul $M := M^{(n-1,1)}$ mit $n \equiv 0 \ (p)$ ist in Charakteristik p unzerlegbar und hat folgende Kompositionsreihe: $M \supset S^{(n-1,1)} \supset D^{(n)} \supset 0$ mit $M/S^{(n-1,1)} \cong D^{(n)}$, $S^{(n-1,1)}/D^{(n)} \cong D^{(n-1,1)}$. Die Einschränkung von M auf $\mathfrak{S}_{n-1}$ ist die direkte Summe aus $M^{(n-1)}$ und $M^{(n-2,1)}$.*

Beweis. Eine Basis von M bilden die $(n-1,1)$–Tabloide. Wir schreiben:

$$\overline{k} = \overline{\begin{array}{c|ccc} 1 & \ldots (k) \ldots & n \\ \hline k & & \end{array}}$$

M ist also n–dimensional. Die Standardpolytabloide bilden eine Basis des $(n-1)$–dimensionalen Moduls $S^{(n-1,1)}$. Auf dem Faktormodul $M^{(n-1,1)}/S^{(n-1,1)}$ operiert $\mathfrak{S}_n$ durch

$$(1,k)(\overline{1} + S^{(n-1,1)}) = \overline{k} + S^{(n-1,1)} = \overline{1} + \overline{k} - \overline{1} + S^{(n-1,1)} = \overline{1} + S^{(n-1,1)} \ ,$$

also trivial. Wäre $S^{(n-1,1)}$ ein direkter Summand, so müsste es also einen eindimensionalen trivialen Untermodul von $M^{(n-1,1)}$ geben, der nicht in $S^{(n-1,1)}$ liegt. Die einzigen Elemente von $M^{(n-1,1)}$, auf denen G trivial operiert, sind aber von der Form

$$\alpha\left(\overline{1} + \ldots \overline{n}\right) = \alpha\left(\left(\overline{2} - \overline{1}\right) + \ldots + \left(\overline{n} - \overline{1}\right) + n\overline{1}\right) \quad (\alpha \in F) \ ,$$

liegen also wegen $p|n$ in $S^{(n-1,1)}$. Der von diesen Elementen gebildete Untermodul ist $D^{(n)}$. Das muss zugleich der eindeutig bestimmte maximale Untermodul von $S^{(n-1,1)}$ sein. Daher ist $M^{(n-1,1)}$ unzerlegbar und besitzt die angegebene Kompositionsreihe.

$\mathfrak{S}_{n-1}$ operiert auf dem von $\overline{n}$ aufgespannten Unterraum trivial und auf dem von $\overline{1}, \ldots, \overline{n-1}$ aufgespannten Unterraum genau wie auf $M^{(n-2,1)}$. Also ist $Res^{\mathfrak{S}_n}_{\mathfrak{S}_{n-1}}(M^{(n-1,1)}) = M^{(n-2,1)} \oplus M^{(n-1)}$.

Bemerkung. Im folgenden benötigen wir noch eine weitere Basis von $M^{(n-1,1)}$:

$$\begin{aligned} \hat{1} &:= \overline{1} \\ \hat{2} &:= \overline{1} - \overline{2} \\ \hat{3} &:= \overline{1} - \overline{3} \\ &\ldots \\ \hat{n} &:= \overline{1} - \overline{n} \end{aligned}$$

Offenbar handelt es sich dabei tatsächlich um eine Basis. $\mathfrak{S}_n$ operiert darauf gemäß folgender Tabelle:

	$\hat{2}$	$\hat{3}$	$\ldots$	$\hat{n}$	$\hat{1}$
$(1,2)$	$-\hat{2}$	$\hat{3}-\hat{2}$	$\ldots$	$\hat{n}-\hat{2}$	$\hat{1}-\hat{2}$
$(1,3)$	$\hat{2}-\hat{3}$	$-\hat{3}$	$\ldots$	$\hat{n}-\hat{3}$	$\hat{1}-\hat{3}$
$\ldots$	$\ldots$				
$(1,n)$	$\hat{2}-\hat{n}$	$\hat{3}-\hat{n}$	$\ldots$	$-\hat{n}$	$\hat{1}-\hat{n}$

4.16 Satz. *Seien F ein Körper der Charakteristik $p > 0$, $n \in \mathbb{N}$ mit $n \equiv 1 \ (p)$. Dann sind Vertizes des einfachen $F\mathfrak{S}_n$–Moduls $D^{(n-1,1)}$ konjugiert zu den p–Sylowgruppen von $\mathfrak{S}_{n-2}$.*

Beweis. Im Fall $p \not| n$ ist der Spechtmodul $S^{(n-1,1)}$ nach Bemerkung 2.14 irreduzibel, d.h. $D^{(n-1,1)} = S^{(n-1,1)}$. Wir verwenden die Polytabloid–Basis und schreiben:

$$\overline{k} = \overline{\begin{array}{|ccccc|}\hline 1 & \dots & (k) & \dots & n \\ \hline k \\ \cline{1-1}\end{array}} - \overline{\begin{array}{|ccc|}\hline 2 & \dots & n \\ \hline 1 \\ \cline{1-1}\end{array}}$$

Dann ist $(\overline{2}, \dots, \overline{n})$ eine Basis von $D^{(n-1,1)}$. $\mathfrak{S}_{n-1}$ operiert darauf gemäß folgender Tabelle

	$\overline{2}$	$\overline{3}$	$\dots$	$\overline{n-1}$	$\overline{n}$
$(1,2)$	$-\overline{2}$	$\overline{3}-\overline{2}$	$\dots$	$\overline{n-1}-\overline{2}$	$\overline{n}-\overline{2}$
$(1,3)$	$\overline{2}-\overline{3}$	$-\overline{3}$	$\dots$	$\overline{n-1}-\overline{3}$	$\overline{n}-\overline{3}$
$\dots$	$\dots$				
$(1,n-1)$	$\overline{2}-\overline{n-1}$	$\overline{3}-\overline{n-1}$	$\dots$	$-\overline{n-1}$	$\overline{n}-\overline{n-1}$

Offenbar erhält man einen Isomorphismus zwischen $\mathrm{Res}^{\mathfrak{S}_n}_{\mathfrak{S}_{n-1}}(D^{(n-1,1)})$ und $M^{(n-2,1)}$ durch $\overline{2} \mapsto \hat{2}, \dots, \overline{n-1} \mapsto \widehat{n-1}, \overline{n} \mapsto \hat{1}$.

Nach Bemerkung 4.1 sind die Vertizes von $D^{(n-2)}$ gerade die p–Sylowgruppen von $\mathfrak{S}_{n-2}$. Wegen $M^{(n-2,1)} = F\mathfrak{S}_{n-1} \otimes_{F\mathfrak{S}_{n-2}} D^{(n-2)}$ gibt es nach Satz 4.11 eine Untergruppe eines Vertex von $D^{(n-2)}$, die zugleich Vertex von $M^{(n-2,1)}$ ist. Andererseits sind nach dem gleichen Satz wegen $\mathrm{Res}^{\mathfrak{S}_{n-1}}_{\mathfrak{S}_{n-2}} M^{(n-2,1)} = D^{(n-2)} \oplus D^{(n-3,1)}$ die Vertizes von $D^{(n-2)}$ Untergruppen der Vertizes von $M^{(n-2,1)}$. Daher sind die p–Sylowgruppen von $\mathfrak{S}_{n-2}$ auch Vertizes von $M^{(n-2,1)}$. Wegen
$|\mathfrak{S}_n : \mathfrak{S}_{n-1}| = n \not\equiv 0 \ \ (p)$ und $\mathrm{Res}^{\mathfrak{S}_n}_{\mathfrak{S}_{n-1}}(D^{(n-1,1)}) \cong M^{(n-2,1)}$ sind nach Bemerkung 4.12 die Vertizes von $M^{(n-2,1)}$ zugleich Vertizes von $D^{(n-1,1)}$.

Bemerkung. Im Fall $p = 2$ besagt dieser Satz nichts anderes als 4.8. Dort wurde bewiesen, dass die 2–Sylowgruppen von $\mathfrak{S}_{n-3}$ Vertizes von $D^{(n-1,1)}$ sind. Da n ungerade ist, sind diese zugleich 2–Sylowgruppen von $\mathfrak{S}_{n-2}$.

4.17 Bemerkung. Damit kennen wir nun die Vertizes der einfachen $F\mathfrak{S}_n$–Moduln zu folgenden Hakenpartitionen ($p = \mathrm{char}(F)$):

(i) $D^{(n)}$ hat die p–Sylowgruppen von $\mathfrak{S}_n$ als Vertizes.

(ii) Nach Satz 4.16 hat $D^{(n-1,1)}$ im Fall $n \equiv 1 \ \ (p)$ die p–Sylowgruppen von $\mathfrak{S}_{n-2}$ als Vertizes.

(iii) Im Fall $p > 3$ und $n \equiv 3 \ \ (p)$ erfüllt $D^{(n-1,1)}$ die Voraussetzungen von 4.7. Die Vertizes von $D^{(n-1,1)}$ sind also p–Sylowgruppen von $\mathfrak{S}_{n-1}$. Wegen $p \not| n$ sind die p–Sylowgruppen von $\mathfrak{S}_{n-1}$ zugleich p–Sylowgruppen von $\mathfrak{S}_n$.

(iv) Allgemeiner gilt im Fall $p > r+1$, $n \equiv 2r+1 \ \ (p)$, $p \not| n$: Die Vertizes von $D^{(n-r,1^r)}$ sind die p–Sylowgruppen von $\mathfrak{S}_{n-r}$.

5 Berechnung von Vertizes

Ich habe mich intensiv mit der Berechnung von Vertizes beschäftigt. Dazu habe ich Programme entwickelt, mit denen man Vertizes einfacher Moduln berechnen kann. Es ist nicht immer sinnvoll, diese Rechnungen voll dem Computer zu überlassen, oftmals kann man durch theoretische Resultate die Kandidaten für die Vertizes soweit einschränken, dass dann wenige Tests genügen und man schneller ans Ziel kommt. So habe ich die Vertizes aller einfachen Moduln der $\mathfrak{S}_n$ $(n \leq 8)$ in allen Charakteristiken berechnet.

5.1 Beispiel. Ich möchte zunächst anhand von Beispielen illustrieren, wie man mit theoretischen Überlegungen die Kandidaten für den Vertex eines Moduls einschränken kann.

(i) Sei F ein Körper der Charakteristik $p > 2$. Nach Satz 4.9 sind die Vertizes der einfachen $F\mathfrak{S}_n$–Moduln für $n < p^2$ gleich den Defektgruppen. Diese lassen sich aber nach Satz 4.5 leicht berechnen.

(ii) Seien jetzt $p = 2$ und $\lambda = (3, 2)$. Ferner seien V ein Vertex von D^λ und P eine Defektgruppe des zugehörigen Blockes mit $V \leq P$. Da das 2–Gewicht von λ gleich 2 ist, folgt: Die Ordnung von P ist 8, es handelt sich also um eine 2–Sylowgruppe. Da die Dimension von D^λ gleich 4 ist, kann $|P : V|$ ebenfalls maximal 4 sein, also ist $|V|$ mindestens 2. Jedoch sind Gruppen der Ordnung 2 zyklisch, scheiden also aus. Es verbleiben noch Gruppen vom Typ $C_2 \times C_2$ und die 2–Sylowgruppen. Tatsächlich sind die Vertizes von D^λ Kleinsche Vierergruppen.

(iii) Seien $p = 2$ und $\lambda = (4, 1)$. Das 2–Gewicht von λ ist 1, also ist $\langle (1, 2) \rangle$ eine Defektgruppe des Blocks von D^λ. Da diese Gruppe zyklisch ist, bildet sie zugleich einen Vertex von D^λ.

5.1 Algorithmen

5.1.1 Vertexberechnung

Sei M ein FG–Modul, von dem bekannt ist, dass er für eine p–Untergruppe H von G projektiv ist. Dann muss es eine Untergruppe V von H geben, die Vertex von M ist. Dann ist M aber auch K–projektiv für alle K mit $V \leq K \leq H$. Man kann also den Vertex finden, indem man für alle maximalen Untergruppen K von H testet, ob M relativ K–projektiv ist. Findet man eine solche Untergruppe, so kann das Verfahren mit dieser wiederholt werden, andernfalls ist H bereits der Vertex. Wenn man keine Kenntnisse über den Modul hat, kann man mit einer p–Sylowgruppe von G beginnen, denn für diese Gruppen ist jeder FG–Modul relativ projektiv. Man erhält also folgenden Algorithmus:

1. Berechne ein $H \in \mathrm{Syl}_p(G)$
2. Berechne Repräsentanten $K_1, \ldots K_k$ für die Konjugationsklassen maximaler Untergruppen von H, Setze $i := 1$
3. Teste, ob M relativ K_i–projektiv ist.
   - JA $\Rightarrow$ Setze $H := K_i$ und gehe zu 2.
   - NEIN $\Rightarrow$ Ist $i < l$, so erhöhe i um 1 und wiederhole diesen Schritt, andernfalls gib H als Vertex aus.

Den ersten Schritt kann man, wenn man bereits Kenntnisse über M hat, auch durch die Eingabe einer Untergruppe $H \leq G$, für die M relativ projektiv ist, umgehen. Diesen Algorithmus habe ich in GAP als Funktion Vertex implementiert. Die Funktion bekommt als Parameter M, G und H übergeben, wobei M relativ H-projektiv sein muss. In den meisten Fällen habe ich beim Aufruf der Funktion für H eine Defektgruppe verwendet.

Weitere Verbesserungen erhält man unter Ausnutzung der Sätze 4.1 und 4.3. So kann man durch eine einfache Abfrage den Test auf relative Projektivität bezüglich zyklischer Gruppen unterbinden. Dabei ist aber darauf zu achten, dass H eine p–Gruppe ist, da es sonst zu fehlerhaften Ausgaben kommen kann. Außerdem habe ich der Funktion eine natürliche Zahl als zusätzlichen Parameter gegeben. Dieser dient als untere Schranke für die Ordnung eines Vertex. Bei jedem rekursiven Funktionsaufruf wird zunächst geprüft, ob die Ordnung von H gleich diesem Wert ist. Ist das der Fall, so wird H als Vertex ausgegeben, andernfalls wird wie oben beschrieben verfahren. Dies hat den Vorteil, dass in den Fällen, in denen der Vertex diese minimale Ordnung hat, nicht mehr getestet werden muss, ob M relativ projektiv ist für die Untergruppen des Vertex. Diese Verbesserung habe ich in Vertex2 implementiert.

Die Berechnung von Sylowgruppen ist bereits in GAP implementiert (SylowSubgroup). Man benötigt also nur noch Funktionen zum Test auf relative Projektivität. Dazu habe ich verschiedene Algorithmen implementiert, die im Folgenden erläutert werden sollen.

5.1.2 Naiver Projektivitätstest

Ein einfaches Verfahren zum Test auf relative Projektivität erhält man direkt aus der Definition. Sei M ein FG–Modul. Wir suchen einen FH–Modul Q mit $M|Q \otimes_{FH} FG$. Nach Satz 2.9 müssen nur die direkten Summanden von $Res^G_H(M)$ als Kandidaten für Q betrachten. Es ergibt sich also folgendes Verfahren:

1. Berechne $M_H := Res^G_H(M)$.
2. Zerlege M_H in direkte Summanden $Q_1, \ldots, Q_l$.

3. Teste für jedes $i \in \{1, \ldots, l\}$, ob $M|Q_i \otimes_{FH} FG$ gilt.

Konnte für ein $i \in \{1, \ldots, l\}$ die Frage in 3. mit "ja" beantwortet werden, dann ist M ein H–projektiver Modul und der Algorithmus kann abgebrochen werden.

Für diesen Algorithmus benötigt man wiederum Funktionen, die einen Modul in direkte Summanden zerlegen bzw. die testen, ob ein Modul zu einem direkten Summanden eines zweiten isomorph ist.

Es zeigt sich, dass dieses Verfahren ungeeignet, da viel zu langsam und speicherintensiv ist. Das Problem ist das Induzieren eines Moduls von einer als Vertex in Verdacht stehenden Untergruppe zur Ausgangsgruppe, da die Dimension des entstehenden Moduls gleich dem Produkt aus Dimension des als Quelle in Verdacht stehenden Moduls und Index der Untergruppe ist.

5.1.3 Verbesserung des naiven Projektivitätstests

Eine Verbesserung erhält man unter Ausnutzung der Tatsache, dass Vertizes stets p–Gruppen sind. Sei $H < G$ eine Untergruppe, für die der Projektivitätstest durchgeführt werden soll. Findet man eine Untergruppe $H' \leq G$ mit $H \in \mathrm{Syl}_p(H')$, so gilt: M ist H–projektiv $\Leftrightarrow$ M ist H'–projektiv.

Der Algorithmus wird also wie folgt abgeändert:

1. Suche eine möglichst große Gruppe H' mit $H \leq H' \leq G$ und $H \in \mathrm{Syl}_p(H')$.
2. Berechne $M_{H'} := Res^G_{H'}(M)$.
3. Zerlege $M_{H'}$ in direkte Summanden $Q_1, \ldots, Q_l$.
4. Teste für jedes $i \in \{1, \ldots, l\}$, ob $M|(Q_i \otimes_{FH'} FG)$ gilt.

Konnte für ein $i \in \{1, \ldots, l\}$ die Frage in 4. mit "ja" beantwortet werden, dann ist M ein H–projektiver Modul und der Algorithmus kann abgebrochen werden.

Mit diesem Verfahren erreicht man eine leichte Verbesserung. Die Gruppenordnung kann bei vergleichbarem Zeit– und Speicheraufwand etwa zehnmal so groß sein.

5.1.4 Direkte Summanden

Die Funktion IsDirectSummand testet, ob ein gegebener Modul zu einem direkten Summanden eines anderen Moduls isomorph ist. Der verwendete Algorithmus ist nicht besonders effizient.

Seien M und N zwei FG–Moduln, N einfach. Zunächst wird eine Basis von $\mathrm{Hom}_{FG}(N, M)$ berechnet. Da N einfach ist, sind die Bilder der nichttrivialen

Homomorphismen zu N isomorph. Es bietet sich an, zum Speichern der Basishomomorphismen ihre Bilder zu verwenden. Diese kann man wiederum durch Angabe einer Basis darstellen. Wenn nun das Bild eines Homomorphismus f ein direkter Summand von M ist, dann gibt es einen maximalen Untermodul M' von M, so dass die Vereinigung der Basen von $f(N)$ und M' eine Basis von M bildet. Man benötigt also noch die Basen aller maximalen Untermoduln von M.

Der Test, ob die Vereinigung der Basen $(b_1, \ldots, b_k)$ von $f(N)$ und $(c_1, \ldots, c_l)$ von M' eine Basis von M bildet, geschieht wie folgt: Als erstes wird geprüft, ob $\dim(M') = \dim(M) - \dim(N)$ gilt. Ist das der Fall, so wird der Rang der Matrix $(b_1 \ldots b_k \quad c_1 \ldots c_l)$ bestimmt. Hat diese Matrix vollen Rang, so ist tatsächlich M' ein direktes Komplement von $f(N)$ und der Algorithmus wird abgebrochen.

Die Funktion Decomposed berechnet eine Zerlegung eines FG–Moduls M in direkte Summanden. Für die Effizienz gilt das Gleiche wie oben.

Zunächst wird ein System von Basen für alle Untermoduln von M, geordnet nach ihrer Dimension berechnet. Anschließend wird, beginnend mit den eindimensionalen, für jeden Untermodul U geprüft, ob er direkter Summand in M ist. Ist das der Fall, so ist U der erste Summand der Zerlegung und Decomposed wird rekursiv auf sein Komplement angewendet.

5.1.5 Verwendung des Higman-Kriteriums

Satz 4.4 liefert ein gutes Projektivitätskriterium: Wir wollen testen, ob der FG-Modul M relativ H–projektiv ist. Zunächst ist der Endomorphismenring $E := \mathrm{End}_{FH}(M)$ zu berechnen. Da $\mathrm{End}_{FG}(M) \cong F$ ist, gilt: Gibt es ein $\vartheta \in E$ mit nichtverschwindender relativer Spur, so gibt es auch ein $\tilde{\vartheta} \in E$ mit $\mathrm{Tr}_H^G(\tilde{\vartheta}) = \mathrm{id}_M$. Es genügt also, die relative Spur für die Elemente einer Basis von E zu berechnen. Findet man eins mit nichtverschwindender relativer Spur, so ist M relativ H–projektiv und der Algorithmus kann abgebrochen werden.

Die Berechnung des Endomorphismenringes geschieht nach folgendem Verfahren. Sei Δ die zu M gehörige Darstellung. Zunächst wird ein Erzeugendensystem von H bestimmt. Für jedes Element h dieses Erzeugendensystems wird dann der Vektorraum aller linearen Abbildungen, die mit $\Delta(h)$ kommutieren, bestimmt. $\mathrm{End}_{FH}(M)$ ist dann der Durchschnitt dieser Vektorräume.

Die linearen Abbildungen werden durch Matrizen bezüglich einer festen Basis repräsentiert. Den Vektorraum der mit einer Matrix $A = (a_{ij})$ kommutierenden Matrizen bestimmt man durch Lösen eines linearen Gleichungssystem. Für eine mit A kommutierende Matrix B muss gelten: $AB - BA = 0$. Koeffizientenvergleich liefert n^2 Gleichungen der Form

$$\sum_{k=1}^{n} a_{ik} b_{kj} - \sum_{k=1}^{n} b_{ik} a_{kj} = 0 \quad (i, j = 1, \ldots, n) \; .$$

Schreibt man die Koeffizienten der Matrix B als Spaltenvektor $b := (b_{11}, \ldots, b_{1n}, \ldots, b_{n1}, \ldots, b_{nn})^T$, so kann man diese zu $\tilde{A}b = 0$ mit der Matrix

$$\tilde{A} = \begin{pmatrix} a_{11}I & \ldots & a_{1n}I \\ \vdots & & \vdots \\ a_{n1}I & \ldots & a_{nn}I \end{pmatrix} - \begin{pmatrix} A^T & & 0 \\ & \ddots & \\ 0 & & A^T \end{pmatrix} = A \otimes I - I \otimes A^T$$

zusammenfassen.

Mit der GAP–Funktion NullspaceMat kann man sich nun den vollständigen Lösungsraum dieser Gleichungssysteme berechnen lassen. Nachdem man den Durchschnitt der Nullräume berechnet hat, müssen noch die Basiselemente, die ja immer noch als Zeilenvektoren gespeichert sind, in Matrizen umgerechnet werden.

5.2 Beispiele zur Berechnung

Ich habe mit den oben vorgestellten Programmen (und den theoretischen Resultaten) die Vertizes aller einfachen Moduln der $\mathfrak{S}_2$ bis $\mathfrak{S}_8$ sowie ihrer Einschränkungen auf die entsprechende Alternierende Gruppe in allen Charakteristiken berechnet. Als Körper habe ich stets den kleinsten Zerfällungskörper verwendet.

Bevor ich die Ergebnisse vorstelle, möchte ich Beispiele für den Programmablauf geben. Außerdem werden Angaben zur Rechenzeit gemacht, die auf meinen privaten Computer bezogen sind. Wichtige technische Daten:

Prozessor	AMD K6/3 400 MHz
RAM	128 MByte
Betriebssystem	Red Hat Linux 6.0
GAP-Version	4

5.2.1 Restriktions–Induktions–Verfahren

Dieses Verfahren soll am Beispiel $D := D^{(3,2)}$ in Charakteristik 2 erläutert werden. Nach Beispiel 5.1 ist der Vertex D_8 oder eine Untergruppe von D_8 mit Index 2. Der Programmstart beginnt also mit $H_0 := \langle (1,2), (1,3)(2,4) \rangle$. Nun werden Repräsentanten für die Konjugationsklassen maximaler Untergruppen von H_0 bestimmt. Das sind z.B.:

H_1^1	C_4	$\langle (1,3,2,4) \rangle$
H_1^2	V_4	$\langle (1,2)(3,4), (1,3)(2,4) \rangle$
H_1^3	$C_2 \times C_2$	$\langle (1,2), (3,4) \rangle$

H_1^1 scheidet nach Satz 4.3 aus.

H_1^2 ist eine 2–Sylowgruppe von A_5. Schränkt man D auf $\mathbb{F}_2 A_5$ ein, so erhält man einen irreduziblen Modul (Anmerkung: Dieser eingeschränkte Modul zerfällt in $\mathbb{F}_4$ in die direkte Summe zweier zweidimensionaler Moduln). Die Induktion dieses Moduls zu $\mathfrak{S}_5$ liefert einen achtdimensionalen Modul, der direkte Summe

zweier zu D isomorpher Moduln ist. Also ist D relativ A_5–projektiv und damit auch H_1^2–projektiv. Der Test auf H_1^3–Projektivität entfällt also.

Da alle Untergruppen von V_4 zyklisch sind, haben wir den Vertex gefunden.

Lässt man sich die Zwischenresultate ausgeben, so sieht die Programmausgabe wie folgt aus. Dabei ist zu beachten, dass GAP die Untergruppen nicht in der gleichen Reihenfolge untersucht hat wie oben beschrieben und dass Gruppen meist durch ein Erzeugendensystem, welches oft nicht minimal ist, angegeben werden.

```
gap> G := SymmetricGroup(5);; H := SylowSubgroup(G,2);;
gap> Read("/daten/gap/5_2/D32");
gap> Vertex(D32,G,H);
Grosse Zwischengruppe fuer Group( [ (1,3)(2,4), (1,2)(3,4) ] ) ist
Group( [ (1,3,2), (2,4,3), (3,5,4) ] )
M ist proj. bzgl. Group( [ (1,3)(2,4), (1,2)(3,4) ] )
DER VERTEX IST Group( [ (1,3)(2,4), (1,2)(3,4) ] )
gap>
```

Dafür benötigte der Computer etwa 10 Sekunden. Läßt man die Suche nach der Zwischengruppe weg, so sind es immerhin 70 Sekunden. Bei größeren Gruppen ist es daher unmöglich, mit dem einfacheren Algorithmus Vertizes zu berechnen.

5.2.2 Higman-Kriterium

Da dieser Algorithmus im Prinzip so abläuft wie beim Restriktions–Induktions–Verfahren, werden hier nicht mehr alle Schritte detailliert erläutert. Die kommentierte Programmausgabe bei $D := D^{(4,3)}$ in Charakteristik 2 sieht so aus:

```
gap> G := SymmetricGroup(7);; H := SylowSubgroup(G,2);;
gap> Read("/daten/gap/7_2/D43");
gap> Vertex2(D43,G,H,4);
Der Modul ist projektiv bzgl.: Group( [ (5,6), (1,3)(2,4), (1,2)(3,4) ] )
Der Modul ist NICHT projektiv bzgl.: Group( [ (5,6), (1,3)(2,4) ] )
Der Modul ist projektiv bzgl.: Group( [ (1,3)(2,4), (1,2)(3,4) ] )

DER VERTEX IST
Group( [ (1,3)(2,4), (1,2)(3,4) ])
gap>
```

Die Rechnung dauerte 2:57 Minuten.

Auf den Projektivitätstest möchte ich am Beispiel: Test auf H–Projektivität mit $H := \langle (1,3)(2,4), (1,2)(3,4) \rangle$ näher eingehen. Zunächst ist $\mathrm{End}_{FH}(D)$ zu

berechnen. Die darstellenden Matrizen der Erzeuger von H sind:

$$(1,2)(3,4) \qquad (1,3)(2,4)$$

$$\begin{pmatrix} 0&0&1&0&0&0&0&0\\ 0&1&0&1&1&0&1&1\\ 1&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&1&1\\ 0&0&0&0&1&0&1&1\\ 0&0&1&0&0&0&0&1\\ 1&0&0&0&0&1&1&1\\ 1&0&0&0&0&1&0&0 \end{pmatrix} \quad \begin{pmatrix} 0&0&1&1&1&0&0&0\\ 0&1&1&1&1&1&1&0\\ 1&0&0&1&1&0&0&0\\ 1&0&0&0&1&1&1&0\\ 1&0&0&1&0&1&1&0\\ 0&0&0&1&1&1&1&1\\ 1&0&1&0&0&0&0&1\\ 1&0&1&0&0&0&1&0 \end{pmatrix}$$

Der Vektorraum der mit diesen Matrizen kommutierenden Matrizen ist 24–dimensional. Unter anderem gehört

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

dazu. Die Relative Spur dieser Abbildung ist die Identität, daher ist D relativ H–projektiv.

5.2.3 Rechenzeit

Einige Beispiele sollen die benötigte Zeit zur Berechnung der Vertizes und die Unterschiede in der Laufzeit der verschiedenen Algorithmen verdeutlichen. Ich habe dazu einige Vertizes mit den im Laufe meiner Arbeit implementierten Algorithmen berechnet. Diese sind wie folgt abgekürzt: RI = Restriktions–Induktions–Verfahren ohne Zwischengruppe, RIZ = Restriktions–Induktions–Verfahren mit Zwischengruppe, Hig1 = Higman–Verfahren ohne Angabe der minimalen Vertexgröße, Hig2 = Higman–Verfahren mit Angabe der minimalen Vertexgröße. In jedem Fall wurde eine p–Sylowgruppe als Start vorgegeben. Die Rechenzeit ist in Minuten angegeben.

p	Modul	dim	Vertex	RI	RIZ	Hig1	Hig2
2	$D^{(3,2)}$	4	V_4	1:10	0:10	0:05	0:02
	$D^{(4,1)}$	4	$\langle(1,2)\rangle$	0:55	—	0:05	0:01
	$D^{(4,3)}$	8	$\mathrm{Syl}_2(\mathfrak{S}_5)$	—	≈ 180	13:56	2:57
	$D^{(7,1)}$	6	$\mathrm{Syl}_2(\mathfrak{S}_8)$	—	—	0:45	0:45
3	$D^{(4,1^2)}$	6	$C_3 \times C_3$	—	—	0:32	0:00[1]

[1]: Aus theoretischen Gründen ist klar, dass die Sylowgruppen Vertizes sind.

Wie man sieht, kostet die zusätzliche Angabe der minimalen Vertexgröße keine Zeit, bringt aber einen enormen Gewinn, wenn die Vertizes tatsächlich diese minimale Größe annehmen.

5.3 Ergebnisse

Die folgenden Tabellen enthalten alle im Rahmen dieser Diplomarbeit berechneten Vertizes. Die Angabe der Vertizes erfolgt unterschiedlich. Gibt es zu einem Gruppentyp nur eine Konjugationsklasse von Untergruppen der $\mathfrak{S}_n$ bzw. $\mathfrak{A}_n$, so wird der Gruppentyp angegeben. Das ist jedoch nicht immer der Fall. So sind z.B. $\langle(1,2)\rangle$ und $\langle(1,2)(3,4)\rangle$ zwei zueinander nicht konjugierte Untergruppen der $\mathfrak{S}_4$ vom Typ C_2. Daher habe ich teilweise ein Erzeugendensystem eines konkreten Vertex verwendet, manchmal auch Angaben der Form $\mathrm{Syl}_2(\mathfrak{S}_4)$. Bei solchen Angaben ist stets zu beachten, dass alle zu einem Vertex konjugierte Untergruppen ebenfalls Vertizes sind.

5.3.1 Charakteristik 2

Bis $n = 6$ wurden alle Vertizes mit dem Restriktions–Induktions–Verfahren berechnet. Nachdem ich den Projektivitätstest mittels Higman–Kriterium programmiert hatte, habe ich auch damit nochmals diese Vertizes berechnet und konnte so die Ergebnisse bestätigen. Anschließend wurden auch die Vertizes der einfachen $\mathfrak{S}_7$– und $\mathfrak{S}_8$–Moduln mit diesem Verfahren berechnet. Die Vertizes der $\mathfrak{A}_7$– und $\mathfrak{A}_8$–Moduln wurden nur dann berechnet, wenn sie nicht ohnehin nach Bemerkung 4.14 gleich den Vertizes des entsprechenden $\mathfrak{S}_7$– bzw. $\mathfrak{S}_8$–Moduls sind.

Die Tabelle ist wie folgt aufgebaut: Zu jedem einfachen $F\mathfrak{S}_n$–Modul gehört eine Zeile. Zunächst sind die zugehörige Partition und seine Dimension angegeben. Anschließend folgen der Vertex des $F\mathfrak{S}_n$–Moduls und dann der Vertex des zugeörigen $F\mathfrak{A}_n$–Moduls bzw. der beiden zugehörigen $F\mathfrak{A}_n$–Moduln. Die verschiedenen Gruppen sind durch eine doppelte Linie, die einzelnen Blöcke durch eine einfache Linie getrennt.

n	λ	$\dim(D^\lambda)$	$\mathrm{vx}(D^\lambda)$	$\mathrm{vx}(\tilde{D}^\lambda)$
2	(2)	1	C_2	1
3	(3)	1	C_2	1
	$(2,1)$	2	1	1
4	(4)	1	$\mathrm{Syl}_2(\mathfrak{S}_4)$	$\mathrm{Syl}_2(A_4)$
	$(3,1)$	2	$\mathrm{Syl}_2(A_4)$	$\mathrm{Syl}_2(A_4)$
5	(5)	1	$\mathrm{Syl}_2(\mathfrak{S}_5)$	$\mathrm{Syl}_2(A_5)$
	$(3,2)$	4	$\mathrm{Syl}_2(A_5)$	$\mathrm{Syl}_2(A_5)$ / $\mathrm{Syl}_2(A_5)$
	$(4,1)$	4	$\langle(1,2)\rangle$	1
6	(6)	1	$\mathrm{Syl}_2(\mathfrak{S}_6)$	$\mathrm{Syl}_2(A_6)$
	$(5,1)$	4	$\mathrm{Syl}_2(\mathfrak{S}_6)$	$\langle(1,2)(3,4),(1,2)(5,6)\rangle$
	$(4,2)$	4	$\mathrm{Syl}_2(\mathfrak{S}_6)$	$\mathrm{Syl}_2(A_5)$
	$(3,2,1)$	16	1	1 / 1
7	(7)	1	$\mathrm{Syl}_2(\mathfrak{S}_7)$	$\mathrm{Syl}_2(A_7)$
	$(5,2)$	14	$\langle(1,2),(3,4),(5,6)\rangle$	$\langle(1,2)(3,4),(1,2)(5,6)\rangle$
	$(4,2,1)$	20	$\mathrm{Syl}_2(\mathfrak{S}_7)$	$\mathrm{Syl}_2(A_5)$
	$(6,1)$	6	$\mathrm{Syl}_2(\mathfrak{S}_5)$	$\mathrm{Syl}_2(A_5)$
	$(4,3)$	8	$\mathrm{Syl}_2(A_5)$	$\mathrm{Syl}_2(A_5)$ / $\mathrm{Syl}_2(A_5)$
8	(8)	1	$\mathrm{Syl}_2(\mathfrak{S}_8)$	$\mathrm{Syl}_2(A_8)$
	$(7,1)$	6	$\mathrm{Syl}_2(\mathfrak{S}_8)$	$\mathrm{Syl}_2(A_8)$
	$(6,2)$	14	$\mathrm{Syl}_2(\mathfrak{S}_8)$	$\mathrm{Syl}_2(A_8)$
	$(5,3)$	8	$\mathrm{Syl}_2(A_8)$	$\mathrm{Syl}_2(A_8)$ / $\mathrm{Syl}_2(A_8)$
	$(4,3,1)$	40	$\mathrm{Syl}_2(A_8)$	$\mathrm{Syl}_2(A_8)$ / $\mathrm{Syl}_2(A_8)$
	$(5,2,1)$	64	$\langle(1,2)\rangle$	1

5.3.2 Charakteristik 3

Bis $n = 6$ wurden alle Vertizes sowohl mit dem Restriktions–Induktions–Verfahren als auch nach dem Higman–Kriterium berechnet. Jedoch liefern diese Resultate keine neuen Erkenntnisse, sie dienen vielmehr als Indiz für die Korrektheit meiner Programme.

Der Grund dafür ist Satz 4.9, wonach für $n \leq 8$ die Vertizes aller einfachen $F\mathfrak{S}_n$–Moduln in Charakteristik 3 gleich den entsprechenden Defektgruppen sind. Diese lassen sich leicht mit Satz 4.5 berechnen.

Nach Bemerkung 4.14 (ii) bleibt beim Einschränken eines $F\mathfrak{S}_n$–Moduls auf $\mathfrak{A}_n$ der Vertex erhalten. Daher werden in den Tabellen nur noch die ersteren angegeben.

n	λ	$\dim(D^\lambda)$	$\mathrm{vx}(D^\lambda)$
3	(3)	1	C_3
	$(2,1)$	1	C_3
4	(4)	1	C_3
	(2^2)	1	C_3
	$(3,1)$	3	1
	$(2,1^2)$	3	1
5	(5)	1	C_3
	$(2^2,1)$	4	C_3
	$(4,1)$	4	C_3
	$(3,2)$	1	C_3
	$(3,1^2)$	6	1
6	(6)	1	$C_3 \times C_3$
	$(5,1)$	4	$C_3 \times C_3$
	$(4,1^2)$	6	$C_3 \times C_3$
	(3^2)	1	$C_3 \times C_3$
	$(3,2,1)$	4	$C_3 \times C_3$
	$(4,2)$	9	1
	$(2^2,1^2)$	9	1

n	λ	$\dim(D^\lambda)$	$\mathrm{vx}(D^\lambda)$
7	(7)	1	$C_3 \times C_3$
	$(5,2)$	13	$C_3 \times C_3$
	$(4,3)$	1	$C_3 \times C_3$
	$(4,2,1)$	20	$C_3 \times C_3$
	$(3,2,1^2)$	13	$C_3 \times C_3$
	$(6,1)$	6	$\langle(1,2,3)\rangle$
	$(3,2^2)$	15	$\langle(1,2,3)\rangle$
	$(5,1^2)$	15	$\langle(1,2,3)\rangle$
	$(3^2,1)$	6	$\langle(1,2,3)\rangle$
8	(8)	1	$C_3 \times C_3$
	$(5,3)$	28	$C_3 \times C_3$
	$(5,2,1)$	35	$C_3 \times C_3$
	$(4,3,1)$	7	$C_3 \times C_3$
	$(3^2,1^2)$	13	$C_3 \times C_3$
	$(7,1)$	7	$C_3 \times C_3$
	$(6,2)$	13	$C_3 \times C_3$
	(4^2)	1	$C_3 \times C_3$
	$(4,2^2)$	35	$C_3 \times C_3$
	$(3,2^2,1)$	28	$C_3 \times C_3$
	$(6,1^2)$	21	$\langle(1,2,3)\rangle$
	$(3^2,2)$	21	$\langle(1,2,3)\rangle$
	$(4,2,1^2)$	90	1

5.3.3 Charakteristik 5 und 7

Nach Satz 4.9 sind für $n < 25$ bzw. $n < 49$ die Vertizes der einfachen $F\mathfrak{S}_n$–Moduln gleich den entsprechenden Defektgruppen. Es sind also wiederum nur die Defektgruppen der Blöcke zu bestimmen.

Ich habe daher die Tabellen anders aufgebaut. Jede Zeile gehört jetzt zu einem Block. Angegeben ist zunächst der p–Kern des Blocks, anschließend die Partitionen, die zu den einfachen Moduln des Blocks gehören. Danach folgt der Typ der Vertizes dieser Moduln. Um Platz zu sparen, habe ich die projektiven Moduln, die alle einen eigenen Block bilden, ebenfalls in einer Zeile zusammengefasst.

Charakteristik 5:

n	Block	zugehörige Partitionen	Vertex
5	$\emptyset$	$(5),(4,1),(3,1^2),(2,1^3)$	C_5
	projektiv	$(3,2),(2^2,1)$	1
6	(1)	$(6),(4,2),(3,2,1),(2^2,1^2)$	C_5
	projektiv	$(5,1),(4,1^2),(3^2),(3,1^3),(2^3),(2,1^4)$	1
7	(2)	$(7),(4,3),(3^2,1),(2^2,1^3)$	C_5
	(1^2)	$(6,1),(5,2),(3,2^2),(2^3,1)$	C_5
	projektiv	$(5,1^2),(4,2,1),(4,1^3),(3,2,1^2),(3,1^4)$	1
8	(3)	$(8),(4^2),(3^2,1^2),(3,2,1^3)$	C_5
	$(2,1)$	$(7,1),(3^2,2)$	C_5
	(1^3)	$(6,1^2),(5,2,1),(4,2^2),(2^4)$	C_5
	projektiv	$(6,2),(5,3),(5,1^3),(4,3,1),(4,2,1^2),$ $(4,1^4),(3,2^2,1),(2^3,1^2),(2^2,1^4)$	1
9	(4)	$(9),(4^2,1),(4,3,1^2),(4,2,1^3)$	C_5
	$(3,1)$	$(8,1),(5,4),(3^2,2,1),(3,2^2,1^2)$	C_5
	(2^2)	$(7,2),(6,3),(3^3),(2^3,1^3)$	C_5
	$(2,1^2)$	$(7,1^2),(5,3,1),(4,3,2),(2^4,1)$	C_5
	(1^4)	$(6,1^3),(5,2,1^2),(4,2^2,1),(3,2^3)$	C_5
	projektiv	$(6,2,1),(5,2^2),(5,1^4),(3^2,1^3),(3,2,1^4)$	C_5

Charakteristik 7:

n	Block	zugehörige Partitionen	Vertex
7	$\emptyset$	$(7),(6,1),(5,1^2),(4,1^3),(3,1^4),(2,1^5)$	C_7
	projektiv	$(5,2),(4,3),(4,2,1),(3^2,1),(3,2^2),$ $(3,2,1^2),(2^3,1),(2^2,1^3)$	1
8	(1)	$(8),(6,2),(5,2,1),(4,2,1^2),(3,2,1^3),(2^2,1^4)$	C_7
	projektiv	$(7,1),(6,1^2),(5,3),(5,1^3),(4^2),(4,3,1),$ $(4,2^2),(4,1^4),(3^2,2),(3^2,1^2),(3,2^2,1),$ $(3,1^5),(2^4),(2^3,1^2),(2,1^6)$	1
9	(2)	$(9),(6,3),(5,3,1),(4,3,1^2),(3^2,1^3),(2^2,1^5)$	C_7
	(1^2)	$(8,1),(7,2),(5,2^2),(4,2^2,1),(3,2^2,1^2),(2^3,1^3)$	C_7
	projektiv	$(7,1^2),(6,2,1),(6,1^3),(5,4),(5,2,1^2),(5,1^4)$ $(4^2,1),(4,3,2),(4,2,1^3),(4,1^5),(3^3)$ $(3^2,2,1),(3,2^3),(3,2,1^4),(3,1^6),(2^4,1)$	1

In Charakteristik $p \geq 11$ ist bis $n = 10$ nach dem Satz von Maschke (2.3) die Gruppenalgebra $F\mathfrak{S}_n$ halbeinfach, d.h. alle Vertizes sind 1.

Literatur

Literatur

[1] J. L. Alperin: Local Representation Theory: Modular Representations as an introduction to the Local Representation Theory of Finite Groups. Cambridge University Press, Cambridge 1993

[2] Christine Bessenrodt / Jörn B. Olsson: On Residue Symbols and the Mullineux Conjecture. Journal of Algebraic Combinatorics 7, 227 – 251 (1998)

[3] Karin Erdmann: Principal Blocks of groups with Dihedral Sylow–2–Subgroups. Communications in Algebra 5, 665 – 694 (1977)

[4] Walter Feit: The Representation Theory of finite groups. North–Holland Mathematical Library 25, North–Holland Publishing Company, Amsterdam 1982

[5] Gordon D. James: The Representation Theory of the Symmetric Groups. Proceedings of Symposia in Pure Mathematics 47, 111 – 126 (1987)

[6] Gordon D. James: The Representation Theory of the Symmetric Groups. Springer–Verlag, Berlin / Heidelberg / New York, 1978

[7] Reinhard Knörr: On the Vertices of Irreducible Modules Annals of Mathematics, II. Series 110, 487 – 499 (1979)

[8] Burkhard Külshammer: Vorlesungen zur Darstellungstheorie. Vorlesungsskript, im Internet unter
http://www.minet.uni-jena.de/algebra/skripten/skripten.html

[9] P. Landrock: Finite Group Algebras and their Modules. London Math. Society Lecture Notes Series 84, Cambridge University Press, Cambridge 1983

[10] N. Meier / J. Tappe: Ein neuer Beweis der Nakayama–Vermutung über die Blockstruktur symmetrischer Gruppen. Bull. London Math. Soc. 8, 34 – 37 (1976).

[11] Gerhard O. Michler / Jorn B. Olsson: Character Correspondences in Finite General Linear, Unitary and Symmetric Groups. Mathemat. Zeitschrift 184, 203 – 233 (1983)

[12] Wolfgang Müller: Darstellungstheorie von endlichen Gruppen. Teubner, Stuttgart 1980

[13] G. M. Murphy / M. H. Peel: Vertices of Specht Modules. Journal of Algebra 86, 85 – 97 (1984)

[14] Hiroshi Nagao / Yukio Tsushima: Representations of finite groups. Academic Press, Boston 1989

[15] Jörn B. Olsson: McKay Numbers and heights of characters. Math. Scandinavica 38, 25 – 42 (1976)

[16] Magdolna Szöke: Examining Green Correspondents of Weight Modules. Aachener Beiträge zur Mathematik 24, Wissenschaftsverlag Mainz in Aachen 1998

Anlagen

A Implementierung der Algorithmen in GAP

A.1 Kompositionsfaktoren der Permutationsmoduln

Die Berechnung der Kompositionsfaktoren der Permutationsmoduln wurden soweit automatisiert, dass man nur noch n, λ und p eingeben muss, um den $\mathbb{F}_p\mathfrak{S}_n$–Modul M^λ zu berechnen. Die Partition λ kann abgekürzt werden, indem man die Teile 1 weglässt. Jedoch muss mindestens ein Teil angegeben werden. λ wird als Liste eingegeben, also z.B. [4,3] für $(4,3)$ oder auch $(4,3,1^2)$. Die Rückgabe ist eine Liste einfacher Moduln mit ihren Vielfachheiten. Die Reihenfolge der Kompositionsfaktoren wird dabei nicht berücksichtigt. Die Syntax ist dabei die folgende: Jeder Eintrag dieser Liste ist wieder eine Liste, bestehend aus dem Modul als MTX.Module und seiner Vielfachheit, also z.B.
[[M_1,5], [M_2,2]]

Die Funktion KomposPerm

```
KomposPerm := function(n, lambda, p)
local G, H, erz, Eins, i, TD, D, M;
1  G := SymmetricGroup(n);
2  H := SymmetricGroup(lambda[1]);
3  for i in [2..Size(lambda)] do
4      if lambda[i]>1 then
5          H := DirectProduct( H, SymmetricGroup(lambda[i]));
6      fi; od;
7  erz := GeneratorsOfGroup(H);
8  Eins := [ ];
9  for i in [1..Size(erz)] do
10     Add(Eins,[[1]]);
11     od;
12 TD := GroupHomomorphismByImages ( H, GL(1,p), erz, Eins);
13 D := InducedRep( G, H, TD);
14 M := GModuleByMats( GeneratorsOfGroup(Image(D)), GF(p));
15 return MTX.CollectedFactors(M);
end;
```

Bedeutung der lokalen Variablen: G und H sind die Gruppen $\mathfrak{S}_n$ bzw. $\mathfrak{S}_\lambda$, erz ist ein Erzeugendensystem von H, Eins ist eine Liste von 1×1–Einheitsmatrizen (die Bilder von erz unter der trivialen Darstellung). i dient als Laufvariable, TD ist die triviale Darstellung von $\mathbb{F}_p\mathfrak{S}_\lambda$, D die Permutationsdarstellung zu λ und M schließlich der Permutationsmodul.

Zum Programmablauf: In den Zeilen 2 bis 6 wird die Gruppe H konstruiert, in 7 bis 11 werden erz und Eins erzeugt. In 12 wird TD definiert und in 13 mittels InducedRep die Induzierte Darstellung berechnet. Schließlich wird mittels der induzierten Darstellung der induzierte Modul und mittels MTX.CollectedFactors seine Kompositionsfaktoren berechnet. Diese werden zurückgegeben.

Die Funktion InducedRep wurde mir von Dr. Burkhard Höfling zur Verfügung gestellt. Die Eingabe ist eine Gruppe G, eine Untergruppe $H \leq G$ und eine Darstellung α von H. Zurückgegeben wird die Darstellung $\mathrm{Ind}_H^G(\alpha)$ von G.

```
InducedRep := function (G, H, alpha)
local rc, t, L, F, n,m,h,i,j,k,l,g, imgs, mat, perm;
L := Image (alpha,H);
if Length (GeneratorsOfGroup(L)) > 0 then
    F := Field (Flat (GeneratorsOfGroup(L)));
else F := Field (Flat (One(L)));
fi;
m := Length (One(L)); # dimension of representation
rc := RightCosets (G,H);
t := List ( rc, Representative);
n := Length (t);
imgs := [ ];
for g in GeneratorsOfGroup(G) do
    mat := NullMat (m*n, m*n, F);
    perm := Permutation (g, rc, OnRight);
    for i in [1..n] do
        j := i^perm;
        h := Image (alpha, t[i]*g*t[j]^-1);
        for k in [1..m] do
            for l in [1..m] do
                mat [(i-1)*m+k][(j-1)*m+l] := h[k][l];
            od;
        od;
    od;
    Add (imgs, mat);
od;
return GroupHomomorphismByImages (G, Group (imgs, IdentityMat (n*m,F)),
    GeneratorsOfGroup(G), imgs);
end;
```

A.2 Restriktions–Induktions–Verfahren

Ich gebe hier nur den letzten Stand des Verfahrens an, also mit der in 5.1.3 beschriebenen Verbesserung.

Die Funktion Vertex berechnet einen Vertex eines einfachen FG–Moduls M, von dem bekannt ist, dass er H–projektiv ist ($H \leq G$). Als Parameter sind anzugeben: Ein MTX–Module M sowie die Gruppen G und H. Die Ausgabe ist dann ein Vertex von M. In der Praxis habe ich mir noch Zwischenresultate ausgeben lassen, und zwar jede getestete Gruppe mit der Angabe, ob M bzgl. dieser Gruppe projektiv ist. Das hat den Vorteil, dass ich dann evtl. schon vorher das Programm stoppen konnte, wenn ich genug wusste.

```
Vertex := function(M, G, H)
local p, subH, subG, U, big, i, j, fertig;
1  p := Factors(Size(M.field))[1];
2  subH := MaximalSubgroupClassReps(H);
3  subG := ConjugacyClassesSubgroups(G);
4  for U in subH do
5      i := Size(subG)-1; fertig := false;
6      repeat
7          for j in subG[i] do
8              if IsSubgroup(j,U) and not (p in Factors(Index(j,U)))
9              then big := j; fertig := true;
10              fi; od;
11          i := i-1;
12     until fertig;
13     if IsProjective(M,G,big) then return Vertex(M,G,U); fi;
14 od;
15 return(H);
end;
```

Die lokalen Variablen haben folgende Bedeutung: p ist die Charakteristik von F, subH ein Repräsentantensystem für die maximalen Untergruppen von H, subG die Menge der Konjugationsklassen von Untergruppen von G, U ist eine Untergruppe von H und big eine Untergruppe von G (die "große Zwischengruppe"); i, j sind Laufvariablen und fertig ist eine boolesche Variable.

1 bis 3: p, subH, und subG werden bestimmt.
In der for–Schleife werden dann die Elemente U von subH durchlaufen.
6 bis 13: big wird berechnet. Dazu werden die Konjugationsklassen von Untergruppen von G in absteigender Ordnung durchlaufen, bis man eine findet für die gilt: Es gibt ein Element diese Konjugationsklasse, welches U als Untergruppe mit einem Index hat, der nicht durch p teilbar ist. Schlimmstenfalls bricht dieser Prozess bei U selbst ab.
In 13 wird getestet ob M relativ big–projektiv ist. Ist das der Fall so wird Vertex rekursiv mit U statt H aufgerufen. Wurde keine Untergruppe von H gefunden, für die M projektiv ist, so ist H ein Vertex von M (unter der Voraussetzung, dass M bereits H–projektiv ist). In diesem Fall wird in 15 H zurückgegeben.

Die Funktion sf IsProjective testet, ob ein einfacher FG–Modul M relativ H–projektiv ist. Dabei muss H eine Untergruppe von G sein. Die Parameter werden genau so angegeben wie bei Vertex. Die Rückgabe erfolgt als boolesche Variable.

```
IsProjective := function( M, G, H)
local Mres, Zerl, i;
1 Mres := RestrictedMod(M,G,H);
2 Zerl := Decomposed(Mres);
3 for i in Zerl do
4 if IsDirectSummand( InducedMod(i,G,H), M) then return true; fi;
5 od;
6 return false;
end;
```

Bedeutung der lokalen Variablen: Mres ist der eingeschränkte Modul $\mathrm{Res}^G_H(M)$, Zerl seine Zerlegung in unzerlegbare Moduln, i ein direkter Summand (dient als Laufvariable).

Zunächst wird mittels RestrictedMod die Einschränkung berechnet und dann mittels Decomposed in unzerlegbare Moduln zerlegt. In 3 bis 5 wird für jeden Modul i dieser Zerlegung geprüft ob $M|\,\mathrm{Ind}^G_H(i)$. Ist das der Fall, wird true zurückgegeben und der Algorithmus abgebrochen. In 6 wird false zurückgegeben, wenn kein direkter Summand i gefunden wurde, für den $M|\,\mathrm{Ind}^G_H(i)$ gilt.

Die Funktion InducedMod dient der Berechnung eines induzierten Moduls, also etwa $M \otimes_{FH} FG$, wobei $H \leq G$ und M ein FH–Modul ist. Die Parameter werden genauso angegeben wie bei Vertex. Die Rückgabe ist ein MTX–Module.

```
InducedMod := function( M, G, H )
local F, D, Dind, Mat;
1 F := M.field;
2 D := GroupHomomorphismByImages( H, GL( M.dimension, Size(F)), Gene-
    ratorsOfGroup(H), M.generators);
3 Dind := InducedRep( G, H, D);
4 Mat := Image( Dind, GeneratorsOfGroup(G));
5 if Mat=[ ] then return GModuleByMats([ ],M.dimension*Index(G,H),F);
6              else return GModuleByMats( Mat, F);
7              fi;
end;
```

Die lokalen Variablen haben folgende Bedeutung: F ist der Körper F, D die zu M gehörige Darstellung von H, Dind die entsprechende induzierte Darstellung von G und Mat sind die darstellenden Matrizen eines Erzeugendensystems von G unter der Darstellung Dind.

Zur Erläuterung des Programmes:
2: Die zu M gehörige Darstellung wird berechnet. Es handelt sich um einen Gruppenhomomorphismus $H \to GL(\dim(M), q)$ mit q als Ordnung des Körpers. Dabei werden den Erzeugern der Gruppe H die erzeugenden Matrizen des Moduls M zugeordnet. Dieses Vorgehen erscheint zunächst gewagt, da ja weder das eine noch das andere Erzeugendensystem wohldefiniert sind. Es zeigt sich jedoch, dass GAP die Erzeugendensysteme in deterministischer Weise bestimmt, wodurch das Verfahren funktioniert.
3: Diese Darstellung wird mittels der Funktion InducedRep (siehe vorheriger Abschnitt) zu G induziert.
4: Die Bilder eines Erzeugendensystems von G unter der induzierten Darstellung werden berechnet.
5: Die Abfrage ist eigentlich pathologisch. Sie ist erforderlich, da bei Verwendung der trivialen Gruppe für G Mat eine leere Liste ist und der Aufruf von GModuleByMats in 6 zu einer Fehlermeldung führt. Im Normalfall wird in
6: der induzierte Modul zurückgegeben.

Die Funktion RestrictedMod berechnet die Einschränkung eines FG–Moduls M auf eine Untergruppe $H \leq G$. Die Parameter und Rückgabe sind vom gleichen Format wie bei Vertex.

```
RestrictedMod := function( M, G, H )
local F, D, bild;
1 F := M.field;
2 D := GroupHomomorphismByImages( G, GL( M.dimension, Size(F) ), Ge-
    neratorsOfGroup(G), M.generators );
3 bild := Image( D, GeneratorsOfGroup(H) );
4 if Size(bild)=0 then return GModuleByMats( bild, M.dimension, F);
5                 else return GModuleByMats( bild, F);
6                 fi;
end;
```

Die lokalen Variablen haben folgende Bedeutung: F ist der Körper F, D die Darstellung von G, die zu M gehört, bild ist das Bild eines Erzeugendensystems von H unter D.

Zur Erläuterung des Programms:
2: Die Darstellung D wird berechnet. Es gilt das Gleiche wie in der Bemerkung zu InducedMod.
4-6: Die Fallunterscheidung ist aus analogen Gründen wie in InducedMod erforderlich, nur tauchen hier die Probleme auf, wenn H die triviale Gruppe ist.

Die Funktion IsDirectSummand prüft, ob ein Modul zu einem direkten Summanden eines zweiten Moduls isomorph ist. Als Parameter werden MTX–Moduln

M, N erwartet, wobei N der einfache Modul ist. Die Rückgabe ist eine boolesche Variable.

```
IsDirectSummand := function( M, N)
local hom, f, submods, v;
1 hom := MTX.Homomorphisms( N, M);
2 submods := MTX.BasesMaximalSubmodules(M);
3 for v in submods do
4     if Size(v) = M.dimension - N.dimension then
5         for f in hom do
6             if RankMat(Concatenation(f,v)) = M.dimension then
7                 return true; fi;
8     od; fi; od;
9 return false;
end;
```

Die lokalen Variablen haben folgende Bedeutung: hom ist eine Basis von $\mathrm{Hom}(N, M)$. Dabei werden die Homomorphismen durch eine Basis ihres Bildes gespeichert. submods ist eine Liste, die die Basen aller maximalen Untermoduln von M enthält. f und v dienen als Laufvariablen.

Zur Erläuterung des Programms:
4: Test, ob v die richtige Dimension hat. Ist das nicht der Fall, so wird 5-7 übersprungen.
6: Test, ob die Basen von f(N) und v zusammen eine Basis von M bilden. Ist das der Fall, so wird der Algorithmus in 7 mit der Rückgabe true abgebrochen
9: Wird nur ausgeführt, wenn kein direktes Komplement für ein f(N) gefunden wurde.

Die Funktion Decomposed berechnet die Zerlegung eines Moduls in direkte Summanden. Eingabe ist ein MTX–Module M. Die Rückgabe ist eine Liste von MTX–Moduln. Dabei handelt es sich um die unzerlegbaren direkten Summanden von M.

```
Decomposed := function(M)
local submods, i,j, M1, M2;
1 submods := MTX.BasesSubmodules(M);
2 for i in [2 .. Size(submods)-2] do
3 for j in [i+1 .. Size(submods)-1] do
4     if Size(submods[i])+Size(submods[j])=M.dimension then
5     if RankMat(Concatenation(submods[i],submods[j])) = M.dimension
6         then M1 := MTX.InducedActionSubmodule(M,submods[i]);
7         M2 := MTX.InducedActionSubmodule(M,submods[j]);
8         return(Concatenation([M1],Decomposed(M2)));
9 fi; fi; od; od;
```

```
10 return([M]);
end;
```

Zur Bedeutung der lokalen Variablen: submods ist ein System von Basen für alle Untermoduln von M. i und j sind Laufvariablen, M1 und M2 sind zwei direkte Summanden von M.

In den beiden for–Schleifen wird die Menge aller zweielementigen Teilmengen von submods durchlaufen; mit wenigen Ausnahmen: Der erste Eintrag von submods ist stets der Nullmodul, der letzte ist stets M. Deshalb werden diese nicht berücksichtigt. In den beiden if–Anweisungen wird überprüft, ob die Summe der Dimensionen beider Untermoduln und die Dimension der Summe beider Moduln gleich der Dimension von M ist. Nur dann sind diese Moduln direkte Summanden. Da die Untermoduln geordnet nach ihrer Dimension, beginnend mit den eindimensionalen, durchsucht werden, ist der erste Summand stets unzerlegbar, während der zweite nicht unzerlegbar sein muss. Deshalb wird in 8 Decomposed rekursiv aufgerufen.

A.3 Verfahren unter Verwendung des Higman–Kriterums

Bei diesem Verfahren stelle ich die Funktion Vertex2 vor, die auf die Suche der "großen Zwischengruppe" verzichtet und dafür als zusätzlichen Parameter die minimale Vertexgröße bekommt.

Die Funktion Vertex2 berechnet den Vertex eines FG–Moduls M, wobei zusätzlich eine p–Gruppe H, bezüglich der M relativ projektiv ist und eine untere Schranke für die Ordnung des Vertex gegeben sein müssen. Als Parameter sind anzugeben: ein MTX–Module M, die Gruppen G und H sowie eine natürliche Zahl min. Die Rückgabe ist ein Vertex von M.

```
Vertex2 := function( M, G, H, min )
local sub, i;
1 if Size(H) > min then
2     sub := MaximalSubgroupClassReps(H);
3     for i in sub do
4         if not IsCyclic(i) then
5         if IPHigman(M,G,i) then return Vertex2(M,G,i,min);
6         fi; fi;
7     od; fi;
8 return H;
end;
```

Die lokale Variable sub ist ein Repräsentantensystem für die Konjugationsklassen maximaler Untergruppen von H, i ist ein Element dieses Systems und

dient als Laufvariable.
1: Der Algorithmus wird nur ausgeführt, wenn die Ordnung von H größer ist als die angegebene untere Schranke für die Ordnung des Vertex. Andernfalls wird sofort in 6 die Gruppe H als Vertex zurückgegeben.
3 - 7: Für die Elemente i von sub wird nacheinander geprüft, ob M relativ i–projektiv ist. Dabei werden zyklische Gruppen übersprungen. Wurde ein solches i gefunden, so wird Vertex2 rekursiv mit i anstelle von H aufgerufen, andernfalls wird in 8 die Gruppe H als Vertex zurückgegeben.

Die Funktion IPHigman prüft mittels Higman–Kriterium, ob ein FG–Modul H–projektiv ist. Als Parameter sind ein MTX–Module M, die Gruppe G und die Untergruppe H, für die der Test durchgeführt werden soll, anzugeben. Die Rückgabe ist eine boolesche Variable.

```
IPHigman := function(M,G,H)
local F,d,E,i;
1 d := M.dimension; F := M.field;
2 E := Endos(M,G,H);
3 for i in E do
4 if not (RelTr(i,M,G,H)=NullMat(d,d,F)) then return true; fi;
5 od;
6 return false;
end;
```

Bedeutung der lokalen Variablen: F ist der Körper F, d die Dimension von M, E ein Erzeugendensystem für den Endomorphismenring $\mathrm{End}_{FH}(M)$ und i ein Element von E (dient als Laufvariable).
In 4 wird für jedes Element von E geprüft, ob die relative Spur von der Nullabbildung verschieden ist. Wenn dies der Fall ist, so ist M relativ H–projektiv und der Algorithmus wird mit der Rückgabe von true abgebrochen. 6 wird nur ausgeführt, wenn kein solcher Endomorphismus gefunden wurde.

Die Funktion Endos berechnet zu einem FG–Modul M den Endomorphismenring $\mathrm{End}_{FH}(M)$. Anzugeben sind ein MTX–Module M sowie die Gruppen G und H. Die Rückgabe ist eine Liste von Matrizen, die $\mathrm{End}_{FH}(M)$ erzeugen.

```
Endos := function(M,G,H)
local F,d,delta,gen,i,V,b,E;
1 F := M.field; d := M.dimension;
2 delta := GroupHomomorphismByImages(G,GL(d,Size(F)), GeneratorsOfGroup(G),
    M.generators);
3 gen := Image(delta,GeneratorsOfGroup(H));
4 V := VectorSpace(F,IdentityMat(d^2,F));
```

```
[5] for i in gen do V := Intersection(V,VectorSpace(F,Vertauscht(i,F))); od;
[6] E := [ ];
[7] for b in Basis(V) do Add(E,AlsMatrix(b,d)); od;
[8] return E;
end;
```

Die lokalen Variablen haben folgende Bedeutung: F und d wie oben, delta ist die zu M gehörende Darstellung, gen ist eine Liste mit den Bildern eines Erzeugendensystems von H unter dieser Darstellung, i ein Element davon (dient als Laufvariable), V ist der Vektorraum der mit den Elementen von gen kommutierenden F–Endomorphismen von M, b wieder eine Laufvariable und E schließlich der gesuchte Endomorphismenring.
2: Berechnung von delta, siehe dazu auch die Anmerkungen zu InducedMod.
4 bis 5: Bestimmung von V. Dazu wird zunächst vom Raum aller F–Endomorphismen von M ausgegangen. Anschließend wird für jedes Element i von gen der Unterraum der mit i vertauschenden Abbildungen berechnet. Der Variablen V wird dann der Durchschnitt dieses Unterraumes und V zugewiesen.
7: Die Elemente einer Basis von V werden in Matrizen umgerechnet und dem zunächst leeren E hinzugefügt.

Die Funktion RelTr berechnet die relative Spur eines FH–Endomorphismus φ des FG–Moduls M. Dabei ist $H \leq G$. Es sind anzugeben: phi als Gruppenhomomorphismus, der MTX–Module M und die Gruppen G und H.

```
RelTr := function(phi,M,G,H)
local d,F,delta,g,tr;
[1] d := M.dimension; F := M.field;
[2] delta := GroupHomomorphismByImages(G,GL(d,Size(F)), GeneratorsOfGroup(G),
    M.generators);
[3] tr := NullMat(d,d,F);
[4] for g in RightTransversal(G,H) do
[5] tr := tr + Image(delta,g)^(-1) * phi * Image(delta,g);
[6] od;
[7] return tr;
end;
```

Die lokalen Variablen haben folgende Bedeutung: d, F und delta wie oben, g ist ein Element von G (dient als Laufvariable) und tr die gesuchte Relative Spur. Zunächst werden d, F und delta berechnet. In 3 - 6 folgt die Berechnung von tr.

Die Funktion Vertauscht berechnet den Kommutator einer quadratischen Matrix M. Anzugeben sind die Matrix und der Körper F, dem die Elemente von M angehören. Die Rückgabe ist eine Liste von Elementen, die den Kommutator als

additive Gruppe erzeugen. Jedoch sind die Elemente nicht als Matrizen, sondern als Vektoren, gespeichert; wie in Abschnitt 5.1.5 beschrieben.

```
Vertauscht := function(M,F)
local I, MschI;
I := IdentityMat(Size(M),F);
MschI := TransposedMat(KroneckerProduct(M,I) - KroneckerProduct(I,
    TransposedMat(M)));
return NullspaceMat(MschI);
end;
```

Die Rechnung erfolgt genau wie in 5.1.5 beschrieben. I ist die Einheitsmatrix und MschI die Matrix $\tilde{M}$.

Die Funktion AlsMatrix dient der Umrechnung der Rückgabe von Vertauscht in eine Matrix. Anzugeben sind ein Vektor v der Länge d^2 sowie (um den Algorithmus zu vereinfachen) d.

```
AlsMatrix := function(v,d)
local M,i,j;
M := NullMat(d,d,Field(v[1]));
for i in [1 .. d] do
for j in [1 .. d] do
M[i][j] := v[(i-1)*d+j];
od; od;
return M;
end;
```

B GAP–Funktionen und –Datenstrukturen

Im folgenden werden alle GAP–Funktionen und –Datenstrukturen, auf die obige Programme zurückgreifen bzw. die in dieser Arbeit zitiert werden, aufgelistet und kurz erläutert.

B.1 Datenstrukturen

Abbildungen GAP kennt eine eigene Datenstruktur für Abbildungen.

Bereiche GAP kennt eigene Datenstrukturen für viele "strukturierte Bereiche", z.B. Gruppen, Körper und Vektorräume.

Gruppen können z.B. durch die Funktion Group erzeugt oder aus der Gruppenbibliothek eingelesen werden. Von letzterer werden verwendet:

SymmetricGroup(n)	$\mathfrak{S}_n$
AlternatingGroup(n)	$\mathfrak{A}_n$
GL(n,q)	$\mathrm{GL}(n, \mathbb{F}_q)$

Körper können z.B. durch GF(q), wobei q eine Primzahlpotenz ist, erzeugt werden.

Liste Listen dienen zum Speichern einer (endlichen) Folge beliebiger Elemente. Die Einträge in einer Liste müssen nicht vom gleichen Typ sein, außerdem kann die Liste Lücken enthalten.
Syntax: [element1, element2, ...]
Der Zugriff auf das i-te Element der Liste List erfolgt durch List[i].

Matrizen, Vektoren Vektoren werden als Listen implementiert, Matrizen sind Listen von Vektoren.

MTX–Module Datenstruktur für FG-Moduln mit endlichem Körper F. Es handelt sich um ein Record, in dem umfangreiche Informationen über den Modul gespeichert sind. Die (für uns) wichtigsten sind:

- .field: selbsterklärend
- .dimension: selbsterklärend
- .generators: Liste mit Matrizen, die die Operation eines Erzeugendensystems von G repräsentieren
- .degreeFieldExt: Grad der Körpererweiterung des Zerfällungskörpers
- .IsIrreducible: Boolesche Variable, die angibt, ob der Modul über F irreduzibel ist

- .IsAbsolutelyIrreducible: Boolesche Variable, selbsterklärend

Permutationen werden in Zyklenschreibweise dargestellt.

Record Ein Record ist die Zusammenfassung mehrerer Variablen zu einer. Der Record hat einen Namen, ebenso seine Komponenten. Mit dem Recordnamen spricht man den gesamten Record an, mit Recordname.Komponentenname eine einzelne Komponente.

B.2 Funktionen

Add Hängt ein einzelnes Element an eine Liste an.

Basis Berechnet eine Basis eines Vektorraums. Die Rückgabe erfolgt als Liste von Vektoren.

Concatenation Gibt die Verknüpfung von zwei Listen zurück.

ConjugacyClassesSubgroups Benötigt eine Gruppe G als Eingabe. Die Rückgabe ist eine Liste mit den Konjugationsklassen von Untergruppen von G. Diese sind nach ansteigender Ordnung der Untergruppen sortiert. Die Konjugationsklassen sind wiederum Listen mit den einzelnen Untergruppen.

DirectProduct Benötigt eine beliebige (positive) Anzahl Gruppen als Eingabe und gibt das direkte Produkt dieser Gruppen zurück.

Factors Zerlegt eine natürliche Zahl in Primfaktoren. Die Rückgabe erfolgt als Liste natürlicher Zahlen in aufsteigender Reihenfolge.

Field Gibt den Körper zurück, zu dem das eingegebene Element gehört.

GeneratorsOfGroup Berechnet ein (nicht notwendig minimales) Erzeugendensystem der eingegebenen Gruppe. Die Rückgabe erfolgt als Liste von Permutationen.

GModuleByMats Benötigt eine Liste von Matrizen, eine natürliche Zahl d und einen Körper F als Eingabe und gibt einen MTX–Module zurück, der durch die Operation der Matrizen auf einem d–dimensionalen Vektorraum über F entspricht. Ist die Matrizenliste nichtleer, so kann auf die Angabe von d verzichtet werden.

Group Benötigt eine Liste von Permutationen als Eingabe und gibt die Gruppe zurück, die von diesen Permutationen erzeugt wird.

GroupHomomorphismByImages Die Eingabe muss die folgende Form haben: G, H, g, h. Dabei sind G und H zwei Gruppen, g eine Liste von Elementen aus G, die G erzeugt und h eine Liste von Elementen aus H, die genau so lang wie g ist. Die Rückgabe ist der Gruppenhomomorphismus $G \to H, g[i] \mapsto h[i]$ als Abbildung.

IdentityMat Bekommt eine natürliche Zahl n und einen Körper F als Eingabe und gibt die $n \times n$–Einheitsmatrix über F zurück.

Image Diese Funktion kann auf zwei Arten verwendet werden. Image(f) gibt das vollständige Bild der Abbildung f zurück (also z.B. eine Gruppe), während Image(f,x) das Bild von x zurückgibt. x kann ein einzelnes Element oder eine Liste von Elementen sein.

Intersection Berechnet den Durchschnitt zweier "verträglicher Bereiche", also z.B. von zwei Gruppen oder zwei Vektorräumen.

IsSubgroup Bekommt zwei Gruppen G, H als Eingabe und prüft, ob H eine Untergruppe von G ist. Die Rückgabe ist true oder false.

KroneckerProduct Berechnet das Kroneckerprodukt zweier Matrizen.

MaximalSubgroupClassReps Die Eingabe ist ein Gruppe G, zurückgegeben wird ein Repräsentantensystem für die Konjugationsklassen maximaler Untergruppen von G.

MTX.BasesMaximalSubmodules Es wird eine Liste der maximalen Untermoduln des eingegebenen MTX–Modules M zurückgegeben. Die Untermoduln sind dabei nicht als MTX–Modules gespeichert, sondern durch eine Basis als Untervektorraum von M.

MTX.BasesSubmodules Wie MTX.BasesMaximalSubmodules, nur dass hier alle Untermoduln berücksichtigt werden.

MTX.CollectedFactors Benötigt einen MTX–Module als Eingabe und berechnet seine Kompositionsfaktoren. Die Rückgabe erfolgt als Liste, deren Einträge wiederum Listen sind, bestehend aus dem Kompositionsfaktor als MTX–Module und seiner Vielfachheit.

MTX.Homomorphisms Benötigt zwei MTX–Modules M, N als Eingabe, wobei von M bekannt sein muss, dass er einfach ist. Die Rückgabe ist eine Liste von Untermoduln von N, welche die Bilder von M unter einer Basis von $\mathrm{Hom}(M, N)$ darstellen. Die Untermoduln sind dabei nicht als MTX–Modules gespeichert, sondern durch eine Basis als Untervektorraum von N.

MTX.InducedActionSubmodule Benötigt einen MTX–Module M und eine Liste von Elementen von M, die eine Basis eines Untermoduls bilden, als Eingaben. Zurückgegeben wird der entsprechende Untermodul als MTX–Module. Dies hat nichts mit dem induzierten Modul zu tun!

MTX.Isomorphism Benötigt zwei MTX–Modules, von denen einer als irreduzibel bekannt ist, als Eingabe, und berechnet einen Isomorphismus zwischen diesen Moduln. Sind die Moduln nicht isomorph, so ist die Rückgabe fail. Ich verwende diese Funktion nur als Isomorphietest, der konkrete Isomorphismus interessiert mich nicht.

NullMat Bekommt zwei natürliche Zahlen m, n und einen Körper als Eingabe und gibt die $m \times n$–Nullmatrix zurück.

NullspaceMat Bekommt eine Matrix A als Eingabe und berechnet den Lösungsraum von $Ax = 0$.

RankMat Berechnet den Rang einer Matrix.

RightTransversal Bekommt zwei Gruppen G, H als Eingabe und berechnet eine Rechtstransversale von H in G.

Size Gibt die Länge einer Liste zurück.

SylowSubgroup Benötigt eine Gruppe G und eine Primzahl p als Eingabe und gibt eine p–Sylowgruppe von G zurück.

TransposedMat Gibt die Transponierte der eingegebenen Matrix zurück.

VectorSpace Benötigt einen Körper F und eine Liste von Vektoren gleicher Länge als Eingabe und gibt den F–Vektorraum, der von diesen Vektoren aufgespannt wird, zurück.

C Kompositionsfaktoren der Permutationsmoduln

Ich habe die Vielfachheiten der einfachen Moduln in den Permutationsmoduln der $\mathfrak{S}_2$ bis $\mathfrak{S}_7$ in allen Charakteristiken in den interessanten Fällen (char$(F)|\,|G|$) berechnet. Die Tabellen sind wie folgt zu lesen: Jede Zeile entspricht einem Permutationsmodul. Links steht die zugehörige Partition, gefolgt von der Dimension. Jede Spalte gehört zu einem einfachen Modul. Oben stehen die zugehörige Partition und die Dimension. In den Tabellenzellen steht schließlich, wie oft der einfache Modul im Permutationsmodul vorkommt. Zur besseren Lesbarkeit wurden Linien eingefügt, die sonst keine Bedeutung haben.

C.1 Charakteristik 2

$\mathfrak{S}_2$		(2)
		1
(2)	1	1
(1^2)	2	2

$\mathfrak{S}_3$		(3)	(2,1)
		1	2
(3)	1	1	0
$(2,1)$	3	1	1
(1^3)	6	2	2

$\mathfrak{S}_4$		(4)	(3,1)
		1	2
(4)	1	1	0
$(3,1)$	4	2	1
(2^2)	6	2	2
$(2,1^2)$	12	4	4
(1^4)	24	8	8

$\mathfrak{S}_5$		(5)	(4,1)	(3,2)
		1	4	4
(5)	1	1	0	0
$(4,1)$	5	1	1	0
$(3,2)$	10	2	1	1
$(3,1^2)$	20	4	2	2
$(2^2,1)$	30	6	2	4
$(2,1^3)$	60	12	4	8
(1^5)	120	24	8	16

$\mathfrak{S}_6$		(6)	(5,1)	(4,2)	(3,2,1)
		1	4	4	16
(6)	1	1	0	0	0
$(5,1)$	6	2	1	0	0
$(4,2)$	15	3	2	1	0
$(4,1^2)$	30	6	4	2	0
(3^2)	20	4	2	2	0
$(3,2,1)$	60	8	5	4	1
$(3,1^3)$	120	16	10	8	2
(2^3)	90	10	6	6	2
$(2^2,1^2)$	180	20	12	12	4
$(2,1^4)$	360	40	24	24	8
(1^6)	720	80	48	48	16

$\mathfrak{S}_7$		(7)	(6,1)	(5,2)	(4,3)	(4,2,1)
		1	6	14	8	20
(7)	1	1	0	0	0	0
$(6,1)$	7	1	1	0	0	0
$(5,2)$	21	1	1	1	0	0
$(5,1^2)$	42	2	2	2	0	0
$(4,3)$	35	1	2	1	1	0
$(4,2,1)$	105	3	3	4	1	1
$(4,1^3)$	210	6	6	8	2	2
$(3^2,1)$	140	4	4	4	2	2
$(3,2^2)$	210	6	4	6	2	4
$(3,2,1^2)$	420	12	8	12	4	8
$(3,1^4)$	840	24	16	24	8	16
$(2^3,1)$	630	18	10	16	6	14
$(2^2,1^3)$	1260	36	20	32	12	28
$(2,1^5)$	2520	72	40	64	24	56
(1^7)	5040	144	80	128	48	112

C.2 Charakteristik 3

$\mathfrak{S}_3$		(3)	(2,1)
		1	1
(3)	1	1	0
$(2,1)$	3	2	1
(1^3)	6	3	3

$\mathfrak{S}_4$		(4)	(3,1)	(2^2)	$(2,1^2)$
		1	3	1	3
(4)	1	1	0	0	0
$(3,1)$	4	1	1	0	0
(2^2)	6	2	1	1	0
$(2,1^2)$	12	2	2	1	1
(1^4)	24	3	3	3	3

$\mathfrak{S}_5$		(5)	(4,1)	(3,2)	$(3,1^2)$	$(2^2,1)$
		1	4	1	6	4
(5)	1	1	0	0	0	0
$(4,1)$	5	1	1	0	0	0
$(3,2)$	10	1	2	1	0	0
$(3,1^2)$	20	1	3	1	1	0
$(2^2,1)$	30	2	4	2	1	1
$(2,1^3)$	60	3	6	3	3	3
(1^5)	120	6	9	6	6	9

$\mathfrak{S}_6$		(6)	(5,1)	(4,2)	$(4,1^2)$	(3^2)	(3,2,1)	$(2^2,1^2)$
		1	4	9	6	1	4	9
(6)	1	1	0	0	0	0	0	0
$(5,1)$	6	2	1	0	0	0	0	0
$(4,2)$	15	2	1	1	0	0	0	0
$(4,1^2)$	30	3	3	1	1	0	0	0
(3^2)	20	2	2	1	0	1	0	0
$(3,2,1)$	60	4	5	2	2	2	1	0
$(3,1^3)$	120	6	9	3	6	3	3	0
(2^3)	90	6	6	3	3	3	3	0
$(2^2,1^2)$	180	9	12	4	8	6	6	1
$(2,1^4)$	360	15	21	6	18	12	15	3
(1^6)	720	27	36	9	36	27	36	9

$\mathfrak{S}_7$		(7)	(6,1)	(5,2)	$(5,1^2)$	(4,3)	(4,2,1)	$(3^2,1)$	$(3,2^2)$	$(3,2,1^2)$
		1	6	13	15	1	20	6	15	13
(7)	1	1	0	0	0	0	0	0	0	0
(6,1)	7	1	1	0	0	0	0	0	0	0
(5,2)	21	2	1	1	0	0	0	0	0	0
$(5,1^2)$	42	2	2	1	1	0	0	0	0	0
(4,3)	35	2	1	2	0	1	0	0	0	0
(4,2,1)	105	4	2	4	1	2	1	0	0	0
$(4,1^3)$	210	6	3	6	3	3	3	0	0	0
$(3^2,1)$	140	4	2	5	2	3	1	1	0	0
$(3,2^2)$	210	6	3	7	2	4	2	1	1	0
$(3,2,1^2)$	420	10	4	11	5	8	6	2	2	1
$(3,1^4)$	840	18	6	18	9	15	15	3	3	3
$(2^3,1)$	630	15	6	15	6	12	9	3	3	3
$(2^2,1^3)$	1260	27	9	24	12	24	21	6	6	9
$(2,1^5)$	2520	51	15	39	21	48	45	12	15	24
(1^7)	5040	99	27	63	36	99	90	27	36	63

C.3 Charakteristik 5

$\mathfrak{S}_5$		(5)	(4,1)	(3,2)	$(3,1^2)$	$(2^2,1)$	$(2,1^3)$
		1	3	5	3	5	1
(5)	1	1	0	0	0	0	0
(4,1)	5	2	1	0	0	0	0
(3,2)	10	2	1	1	0	0	0
$(3,1^2)$	20	3	3	1	1	0	0
$(2^2,1)$	30	3	3	2	1	1	0
$(2,1^3)$	60	4	6	3	4	2	1
(1^5)	120	5	10	5	10	5	5

$\mathfrak{S}_6$		(6)	(5,1)	(4,2)	$(4,1^2)$	(3^2)	(3,2,1)	$(3,1^3)$	(2^3)	$(2^2,1^2)$	$(2,1^4)$
		1	5	8	10	5	8	10	5	1	5
(6)	1	1	0	0	0	0	0	0	0	0	0
(5,1)	6	1	1	0	0	0	0	0	0	0	0
(4,2)	15	2	1	1	0	0	0	0	0	0	0
$(4,1^2)$	30	2	2	1	1	0	0	0	0	0	0
(3^2)	20	2	1	1	0	1	0	0	0	0	0
(3,2,1)	60	3	2	3	1	1	1	0	0	0	0
$(3,1^3)$	120	4	3	5	3	1	2	1	0	0	0
(2^3)	90	4	2	5	1	1	2	0	1	0	0
$(2^2,1^2)$	180	5	3	8	3	2	5	1	1	1	0
$(2,1^4)$	360	7	4	14	6	3	11	4	2	3	1
(1^6)	720	10	5	25	10	5	25	10	5	10	5

$\mathfrak{S}_7$		(7)	$(6,1)$	$(5,2)$	$(5,1^2)$	$(4,3)$	$(4,2,1)$	$(4,1^3)$	$(3^2,1)$	$(3,2^2)$	$(3,2,1^2)$	$(3,1^4)$	$(2^3,1)$	$(2^2,1^3)$
		1	6	8	15	13	35	20	8	13	35	15	1	6
(7)	1	1	0	0	0	0	0	0	0	0	0	0	0	0
$(6,1)$	7	1	1	0	0	0	0	0	0	0	0	0	0	0
$(5,2)$	21	1	2	1	0	0	0	0	0	0	0	0	0	0
$(5,1^2)$	42	1	3	1	1	0	0	0	0	0	0	0	0	0
$(4,3)$	35	2	2	1	0	1	0	0	0	0	0	0	0	0
$(4,2,1)$	105	2	4	2	1	1	1	0	0	0	0	0	0	0
$(4,1^3)$	210	2	6	3	3	1	2	1	0	0	0	0	0	0
$(3^2,1)$	140	3	4	2	1	3	1	0	1	0	0	0	0	0
$(3,2^2)$	210	3	5	4	1	3	2	0	1	1	0	0	0	0
$(3,2,1^2)$	420	4	7	5	3	5	4	1	2	1	1	0	0	0
$(3,1^4)$	840	5	10	8	6	7	8	4	3	2	3	1	0	0
$(2^3,1)$	630	5	8	8	3	7	6	1	3	4	2	0	1	0
$(2^2,1^3)$	1260	7	11	12	6	12	11	4	7	7	6	1	2	1
$(2,1^5)$	2520													
(1^7)	5040													

C.4 Charakteristik 7

$\mathfrak{S}_7$		(7)	(6,1)	(5,2)	$(5,1^2)$	(4,3)	(4,2,1)	$(4,1^3)$	$(3^2,1)$	$(3,2^2)$	$(3,2,1^2)$	$(3,1^4)$	$(2^3,1)$	$(2^2,1^3)$	$(2,1^5)$
		1	5	14	10	14	35	10	21	21	35	5	14		
(7)	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
(6,1)	7	2	1	0	0	0	0	0	0	0	0	0	0	0	0
(5,2)	21	2	1	1	0	0	0	0	0	0	0	0	0	0	0
$(5,1^2)$	42	3	3	1	1	0	0	0	0	0	0	0	0	0	0
(4,3)	35	2	1	1	0	1	0	0	0	0	0	0	0	0	0
(4,2,1)	105	3	3	2	1	1	1	0	0	0	0	0	0	0	0
$(4,1^3)$	210	4	6	3	4	1	2	1	0	0	0	0	0	0	0
$(3^2,1)$	140	3	3	2	1	2	1	0	1	0	0	0	0	0	0
$(3,2^2)$	210	3	3	3	1	2	2	0	1	1	0	0	0	0	0
$(3,2,1^2)$	420	4	6	4	4	3	4	1	2	1	1	0	0	0	0
$(3,1^4)$	840	5	10	6	10	4	8	5	3	2	3	1	0	0	0
$(2^3,1)$	630	4	6	5	4	4	6	1	3	3	2	0	1	0	0
$(2^2,1^3)$	1260													1	0
$(2,1^5)$	2520														1
(1^7)	5040														

Erklärung

Ich erkläre, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Jena, 12.04.2000 (René Zimmermann)