Computerpraktikum

Block I

Die erste und die zweite Cohomologiegruppe

Matthias Künzer

Universität Stuttgart

12. Mai 2019

Inhalt

Vorwort

Sei G eine endliche Gruppe.

Sei M ein G-Modul. D.h. sei M eine abelsche Gruppe, zusammen mit einer Abbildung $G \times M \to M$, die Eigenschaften einer äußeren Operation von G auf M erfüllt.

Z.B. können wir $M = \mathbb{Z}$ wählen, zusammen mit der Abbildung $G \times \mathbb{Z} \to \mathbb{Z}$, für welche $g \cdot z = z$ ist für $g \in G$ und $z \in \mathbb{Z}$.

Eine Derivation sei eine Abbildung g von G nach M, für die stets $d(g \cdot g') = d(g) + g \cdot d(g')$ ist.

Eine innere Derivation sei eine solche Abbildung d der Form $d(g) = g \cdot m - m$ für ein festes $m \in M$.

Man definiert nun die erste Cohomologiegruppe $\mathrm{H}^1(G, M)$ als die Gruppe der Derivationen modulo der Untergruppe der inneren Derivationen.

Ähnlich, nur etwas aufwendiger dann die zweite Cohomologiegruppe.

Wir wollen die erste und die zweite Cohomologiegruppe berechnen mittels eines Verfahrens, das auf Zassenhaus [**6**] zurückgeht, von Holt, Eick und O'Brien [**2,** §7.6] in den Cohomologie-Kontext übersetzt wurde und welches ich von Christian Weber [**4,** §1.5], [**5,** §5] kenne.

Dank geht an Lukas Hauger, Felicitas Walter und Julien Flad für Korrekturen.

Für weitere Hinweise auf Fehler und Unklarheiten bin ich dankbar.

Stuttgart, Frühjahr 2019

Matthias Künzer

1 Magma

Wir verwenden das Computeralgebra-System Magma [**1**].

Das Handbuch findet sich unter `magma.maths.usyd.edu.au/magma/`.

Es kann Magma aufgerufen werden durch Eingabe von `magma` in einer Shell auf einem `stud`-Rechner.

Man teste einmal `1+1;` und dann `Enter`.

2 Derivationen und H^1

Sei G eine Gruppe.

Definition 1 Ein *G-Modul* ist eine abelsche Gruppe M, zusammen mit einer Abbildung

$$\begin{array}{ccccc} G & \times & M & \to & M \\ (g & , & m) & \mapsto & g \cdot m \end{array}$$

für welche

$$\begin{array}{rcl} g \cdot (m + m') & = & g \cdot m + g \cdot m' \\ 1 \cdot m & = & m \\ g \cdot (g' \cdot m) & = & (g \cdot g') \cdot m \end{array}$$

gelten für g, $g' \in G$ und m, $m' \in M$.

Beispiel 2 Sei $k \in \mathbb{Z}$. Sei $M = \mathbb{Z}/k\mathbb{Z}$. Sei $g \cdot m := m$ für $g \in G$ und $m \in M$.

Man spricht vom *trivialen* G-Modul $\mathbb{Z}/k\mathbb{Z}$.

Für $k > 0$ prim werden wir dieses Beispiel für erste direkte Berechnungen verwenden können, da dann alle betrachteten Mengen endlich werden.

Beispiel 3 Sei G endlich.

Der *Gruppenring* $\mathbb{Z}G$ hat als additive Gruppe den freien $\mathbb{Z}$-Modul auf der Menge G.

Es besteht also $\mathbb{Z}G$ aus den formalen $\mathbb{Z}$-Linearkombinationen der Elemente aus G, i.e.

$$\mathbb{Z}G \;=\; \{ \sum_{g \in G} z_g g : \; z_g \in \mathbb{Z} \text{ für } g \in G \} \; .$$

Die Multiplikation auf dem Ring $\mathbb{Z}G$ ist gegeben durch die Multiplikation auf G, distributiv fortgesetzt.

Also ist

$$(\sum_{g \in G} z_g g) \cdot (\sum_{g \in G} z'_g g) \;=\; \sum_{g,h \in G} z_g z_h g \cdot h \; ,$$

wobei z_g, $z'_g \in \mathbb{Z}$ und wobei $g \cdot h$ in G gebildet wird.

Mit anderen Worten, $\mathbb{Z}G$ trägt die Multiplikation von G, distributiv fortgesetzt.

Es ist $\mathbb{Z}G$ ein G-Modul, mit der Multiplikation

$$h \cdot \sum_{g \in G} z_g g \;:=\; \sum_{g \in G} z_g h \cdot g$$

für $h \in G$ und $\sum_{g \in G} z_g g \in \mathbb{Z}G$, i.e. mit der vom Ring $\mathbb{Z}G$ stammenden Multiplikation. Dieser Modul heißt auch *regulärer* G-Modul.

Bemerkung 4 Sei $d \geqslant 1$. Sei $M := \mathbb{Z}^{d\times 1}$. Sei

$$\gamma \,:\, G \;\to\; \mathbb{Z}^{d\times d}$$

eine Abbildung, für welche $\gamma(1) = \mathrm{E}_d$ und $\gamma(g \cdot g') = \gamma(g) \cdot \gamma(g')$ gilt.

Dann wird M zu einem G-Modul, wenn wir

$$g \cdot m \;:=\; \gamma(g) \cdot m$$

setzen für $g \in G$ und $m \in M$, wobei letztere Multiplikation eine Matrix $\gamma(g) \in \mathbb{Z}^{d\times d}$ mit einem Vektor $m \in M = \mathbb{Z}^{d\times 1}$ multipliziert.

Definition 5 Sei M ein G-Modul.

Eine *Derivation* von G in M ist eine Abbildung

$$\delta \,:\, G \;\to\; M$$

welche

$$\delta(g \cdot g') = \delta(g) + g \cdot \delta(g')$$

erfüllt für $g,\, g' \in G$.

Wir schreiben

$$\mathrm{Z}^1(G, M)$$

für die Menge der Derivationen von G in M.

Da die Nullabbildung, die jedes Element von G auf $0 \in M$ schickt, eine Derivation ist, und da für Derivationen δ und $\tilde{\delta}$ auch $\delta - \tilde{\delta}$ eine Derivation ist, ist $\mathrm{Z}^1(G, M)$ eine Untergruppe der abelschen Gruppe aller Abbildungen von G nach M.

Insbesondere ist $\mathrm{Z}^1(G, M)$ eine abelsche Gruppe.

Bemerkung 6 Sei $\delta \in \mathrm{Z}^1(G, M)$.

Es ist

$$0 \;=\; \delta(1) - \delta(1) \;=\; \delta(1 \cdot 1) - \delta(1) \;=\; \delta(1) + 1 \cdot \delta(1) - \delta(1) \;=\; \delta(1)\;.$$

Für $g \in G$ ist $0 = \delta(1) = \delta(g \cdot g^{-1}) = \delta(g) + g \cdot \delta(g^{-1})$ und also

$$\delta(g^{-1}) \;=\; -g^{-1} \cdot \delta(g)\;.$$

Für $m \geqslant 1$ und $g_1\,,\, g_2\,,\, \ldots\,,\, g_m \,\in\, G$ erhalten wir

$$\delta(g_1 \cdot g_2 \cdot \ldots \cdot g_m) \;=\; \delta(g_1) + g_1 \cdot \delta(g_2) + g_1 \cdot g_2 \cdot \delta(g_3) + \ldots + g_1 \cdot g_2 \cdot \ldots \cdot g_{m-1} \cdot \delta(g_m)$$

Definition 7 Sei $m \in M$.

Sei die Abbildung $\delta_m^{\mathrm{inn}} : G \to M$ definiert durch

$$\delta_m^{\mathrm{inn}}(g) \ := \ g \cdot m - m$$

für $g \in G$.

Es ist δ_m^{inn} eine Derivation, da

$$\delta_m^{\mathrm{inn}}(g) + g \cdot \delta_m^{\mathrm{inn}}(g') \ = \ m - g \cdot m + g \cdot (m - g' \cdot m) \ = \ m - g \cdot g' \cdot m \ = \ \delta_m^{\mathrm{inn}}(g \cdot g')$$

ist für g, $g' \in G$. Eine Derivation der Form δ_m^{inn} für ein m heißt auch *innere Derivation.*

Wir schreiben

$$\mathrm{B}^1(G, M) \ := \ \{\, \delta_m^{\mathrm{inn}} : \ m \in M \,\} \ \subseteq \ \mathrm{Z}^1(G, M)$$

für die Menge der inneren Derivationen.

Es ist $\delta_0^{\mathrm{inn}} = 0$ und $\delta_m^{\mathrm{inn}} - \delta_{m'}^{\mathrm{inn}} = \delta_{m-m'}^{\mathrm{inn}}$. Folglich ist $\mathrm{B}^1(G, M)$ eine Untergruppe der abelschen Gruppe $\mathrm{Z}^1(G, M)$.

Definition 8 Die Faktorgruppe

$$\mathrm{H}^1(G, M) \ := \ \mathrm{Z}^1(G, M)/\mathrm{B}^1(G, M)$$

heißt *erste Cohomologiegruppe* von G in M.

Ihre Elemente sind also von der Form $\delta + \mathrm{B}^1(G, M)$ mit $\delta \in \mathrm{Z}^1(G, M)$.

Hierbei gilt $\delta + \mathrm{B}^1(G, M) = \delta' + \mathrm{B}^1(G, M)$ genau dann, wenn $\delta - \delta' \in \mathrm{B}^1(G, M)$ liegt, wobei δ, $\delta' \in \mathrm{Z}^1(G, M)$.

Ferner wird in der abelschen Gruppe $\mathrm{H}^1(G, M)$ repräsentantenweise addiert, i.e.

$$(\delta + \mathrm{B}^1(G, M)) + (\delta' + \mathrm{B}^1(G, M)) \ = \ (\delta + \delta') + \mathrm{B}^1(G, M) \ ,$$

wobei δ, $\delta' \in \mathrm{Z}^1(G, M)$.

> Uns soll weniger interessieren, wozu man $\mathrm{H}^1(G, M)$ ausrechnet, als wie man diese Gruppe ausrechnet.
>
> Wer es dennoch wissen will, der findet Antworten e.g. in [**3,** §3.1] oder in der Algebraischen Topologie.

Bemerkung 9 Ist G endlich, so ist

$$|G| \cdot \mathrm{H}^1(G, M) \ = \ 0$$

aus der Theorie bekannt.

Ist M endlich und $\exp(M)$ das kleinste gemeinsame Vielfache der Ordnungen der Elemente von M, so ist $\exp(M) \cdot \mathrm{H}^1(G, M) = 0$. Insbesondere ist dann

$$|M| \cdot \mathrm{H}^1(G, M) \ = \ 0 \ .$$

Sind also $|G|$ und $|M|$ teilerfremd, dann ist $\mathrm{H}^1(G, M) = 0$.

Beispiel 10 Sei $G = \mathrm{S}_3$, in Magma also `G := SymmetricGroup(3)`. Mit `;` und `Enter` eingeben.

Wir listen G auf: `G_list := [g : g in G]`. Die Liste können wir mittels `print G_list` betrachten.

```
[
    Id(G),
    (1, 2, 3),
    (1, 3, 2),
    (2, 3),
    (1, 2),
    (1, 3)
]
```

Wir können auch in G rechnen, e.g. gibt `G!(1,2) * G!(2,3)` das Ergebnis `(1,3,2)`. Magma komponiert also in der natürlichen Reihenfolge, nicht in der traditionellen, i.e. es wird zuerst `G!(1,2)` ausgeführt und dann `G!(2,3)`.

Sei $M := \mathbb{Z}/2\mathbb{Z}$ der triviale G-Modul, in Magma also `M := Integers(2)`.

Wir bilden die Liste der Elemente von M: `M_list := [m : m in M]`. Mit `print M_list` erhalten wir folgendes.

```
[ 0, 1 ]
```

Wir bilden die Menge aller Abbildungen von G nach M, welche wir interpretieren als Tupel, in welchem der i-te Eintrag das Bild des i-ten Gruppenelements in `G_list` sei.

`Abb := CartesianPower(M_list,6)`.

Wir sortieren nun aus `Abb` die Derivationen aus.

Derivationen sind hier nun zugleich die Gruppenmorphismen von G nach M, aber davon wollen wir keinen Gebrauch machen.

```
derivations := [];
for d in Abb do
 is_derivation := true;
 for i in [1..#G_list] do     // i: Nummer des ersten zu testenden Gruppenelements
  for j in [1..#G_list] do    // j: Nummer des zweiten zu testenden Gruppenelements
   if not d[i] + d[j] eq d[Index(G_list, G_list[i] * G_list[j])] then
    is_derivation := false;
    break i;
   end if;
  end for;
 end for;
 if is_derivation then
  derivations cat:= [d];
 end if;
end for;
print derivations;
```

Oder kürzer:

```
derivations :=
[d : d in Abb | &and[d[i] + d[j] eq d[Index(G_list, G_list[i] * G_list[j])] : i,j in [1..#G_list]]];
print derivations;
```

Beidemale erhalten wir `[ <0, 0, 0, 0, 0, 0>, <0, 0, 0, 1, 1, 1> ]`.

Die einzige nichtverschwindende Derivation δ bildet also die Elemente `Id(G), (1, 2, 3), (1, 3, 2)` auf 0 ab und die Elemente `(2, 3), (1, 2), (1, 3)` auf 1.

Wegen M trivial ist die Nullabbildung die einzige innere Derivation.

Also ist die Liste der Derivationen bereits die Liste der Elemente von H^1.

Folglich ist

$$\mathrm{H}^1(G, M) \simeq \mathbb{Z}/2\mathbb{Z} ,$$

wobei bei der rechten Seite nur die additive Gruppe gemeint ist.

Beispiel 11 Sei `G := SymmetricGroup(3)` und `G_list := [g : g in G]`.

Sei `M := Integers(3)` und `M_list := [m : m in M]` ein trivialer G-Modul.

Sei `Abb := CartesianPower(M_list,6)`. Wir sortieren nun aus `Abb` die Derivationen aus.

```
derivations :=
[d : d in Abb | &and[d[i] + d[j] eq d[Index(G_list, G_list[i] * G_list[j])] : i,j in [1..6]]];
print derivations;
```

Wir erhalten `[ <0, 0, 0, 0, 0, 0>]`, i.e.

$$\mathrm{H}^1(G, M) \simeq 0 .$$

Beispiel 12 Sei `G := SymmetricGroup(3)` und `G_list := [g : g in G]`.

Sei `M := Integers(3)` und `M_list := [m : m in M]`.

Sei nun aber M kein trivialer G-Modul mehr, sei vielmehr

$$g \cdot m = \mathrm{sgn}(g) \cdot m .$$

Sei `Abb := CartesianPower(M_list,6)`. Wir sortieren nun aus `Abb` die Derivationen aus.

```
derivations :=
[d : d in Abb |
&and[d[i] + Sign(G_list[i]) * d[j] eq d[Index(G_list, G_list[i] * G_list[j])]
: i,j in [1..6]]];
print derivations;
```

Wir erhalten

```
[ <0, 0, 0, 0, 0, 0>, <0, 0, 0, 1, 1, 1>, <0, 0, 0, 2, 2, 2>,
  <0, 1, 2, 0, 2, 1>, <0, 1, 2, 1, 0, 2>, <0, 1, 2, 2, 1, 0>,
  <0, 2, 1, 0, 1, 2>, <0, 2, 1, 1, 2, 0>, <0, 2, 1, 2, 0, 1> ] ,
```

i.e.

$$\mathrm{Z}^1(G, M) \simeq \mathbb{Z}/3\mathbb{Z} \oplus \mathbb{Z}/3\mathbb{Z} .$$

Nun haben wir aber auch nichtverschwindende innere Derivationen.

```
inner_derivations :=
{<Sign(G_list[i]) * m - m : i in [1..6]> : m in M};
print inner_derivations;
```

Wir erhalten `{ <0, 0, 0, 0, 0, 0>, <0, 0, 0, 1, 1, 1>, <0, 0, 0, 2, 2, 2> }`, i.e.

$$\mathrm{B}^1(G, M) \simeq \mathbb{Z}/3\mathbb{Z} .$$

Folglich ist

$$\mathrm{H}^1(G, M) = \mathrm{Z}^1(G, M)/\mathrm{B}^1(G, M) \simeq \mathbb{Z}/3\mathbb{Z} .$$

3 Umgang mit Gruppen

Mit `SmallGroups` erhält man eine Liste von Gruppen von vorgegebener Ordnung.

Seien e.g. `G := SmallGroups(6)[1]` und `H := SmallGroups(6)[2]` die beiden Gruppen von Ordnung 6.

Es hat `IsAbelian(G)` den Wert `false`, während `IsAbelian(H)` den Wert `true` hat. In der Tat hat auch `IsIsomorphic(G,SymmetricGroup(3))` den Wert `true`.

Mit `H_perm := PermutationGroup(FPGroup(H))` erhält man eine zu `H` isomorphe Untergruppe `H_perm` einer symmetrischen Gruppe. Magma konstruiert `H_perm` als Untergruppe der S_6, erzeugt von `(1, 2)(3, 5)(4, 6)` und `(1, 3, 4)(2, 5, 6)`. In der Tat gibt `[x : x in H_perm]` die Liste

```
[
    Id(H_perm),
    (1, 2)(3, 5)(4, 6),
    (1, 3, 4)(2, 5, 6),
    (1, 4, 3)(2, 6, 5),
    (1, 5, 4, 2, 3, 6),
    (1, 6, 3, 2, 4, 5)
] ,
```

was zeigt, daß `H_perm` die von `(1,5,4,2,3,6)` erzeugte zyklische Gruppe ist.

`G_perm := PermutationGroup(FPGroup(G))` liefert dagegen, wenig überraschend, die S_3, namentlich als die von `(2,3)` und `(1,2,3)` erzeugte Untergruppe von S_3.

Der Zwischenschritt `G_fp<a,b> := FPGroup(G)` liefert die von als `a` und `b` erzeugte Gruppe, modulo den Relationen `a^2 = 1`, `b^3 = 1` und `b^a = b^2`. Mit anderen Worten,

$$\texttt{G\_fp} \;=\; \langle a, b : a^2, b^3, b^a \cdot b \rangle$$

Hierbei ist $b^a := a^{-1}ba$.

Dies bedeutet, daß zum einen in `G_fp` die Gleichungen $a^2 = 1$, $b^3 = 1$ und $b^a = b^2$ gelten. Zum anderen kann ein Gruppenmorphismus f von `G_fp` in eine andere Gruppe K dadurch eindeutig definiert werden, daß $f(a)$, $f(b) \in K$ so gewählt werden, daß in K die Gleichungen $f(a)^2 = 1$, $f(b)^3 = 1$ und $f(b)^{f(a)} = f(b)^2$ gelten.

Z.B. kann ein Gruppenmorphismus f von `G_fp` nach `G_fp` durch $f(a) := a$ und $f(b) := 1$ definiert werden.

Mehr Details hierzu in §4.

Es ist `G_fp` zu `G_perm` isomorph. Diesen Isomorphismus wird von Magma geliefert, wenn wir uns `G_perm` nochmals definieren, und dabei als zweite Ausgabe einen Isomorphismus `phi` mitliefern lassen: `G_perm,phi := PermutationGroup(G_fp)`. Es ist dann `phi` ein Gruppenisomorphismus von `G_fp` nach `G_perm`, für welchen `phi(a)` zu `(2,3)` und `phi(b)` zu `(1,2,3)` wird.

Sowieso wird `phi(G_fp!1)` zu `Id(G_perm)`. Es wird aber z.B. auch `phi(a*b)` zu `(1,2)`.

Rückwärts wird `(G_perm!(1,2))@@phi` wieder zu `a*b`.

4 Gruppen in Erzeugern und Relationen

Wir wollen eine Gruppen mit vorgegebenen Erzeugern und vorgegebenen Relationen definieren. Der erste Schritt ist dabei die Konstruktion der freien Gruppe.

Definition 13 Sei $n \geqslant 1$. Seien x_1, x_2, ..., x_n paarweise verschiedene Elemente.

Wir bilden die Elemente x_1^{-1}, x_2^{-1}, ..., x_n^{-1}, was zunächst nur eine Schreibweise sei.

Zusammen haben wir so eine Menge

$$A \;:=\; \{x_1, x_2, \ldots, x_n, x_1^{-1}, x_2^{-1}, \ldots, x_n^{-1}\}$$

aus $2n$ Elementen.

Sei F_0 die Menge der Wörter in A.

Es trägt F_0 als Multiplikation die Operation des Aneinandersetzens von Wörtern.

Das leere Wort wird 1_F oder kurz 1 geschrieben.

Sei $(\approx)$ die Relation auf F_0, die aus den Paaren der Form

$$(wx_kx_k^{-1}w', ww')$$

und den Paaren der Form

$$(wx_k^{-1}x_kw', ww')$$

besteht, wobei $w,\ w' \in F_0$ und wobei $k \in [1,n]$.

Sei $(\sim)$ die von $(\approx)$ erzeugte Äquivalenzrelation.

Sei $F := F_0/(\sim)$ die Menge der Äquivalenzklassen.

Es wird F eine Gruppe, indem man die Multiplikation repräsentantenweise durch Multiplikation in F_0 definiert.

Es heißt F die *freie Gruppe* auf $x_1\,,\ x_2\,,\ \ldots\,,\ x_n\,.$

Die Elemente von F werden repräsentantenweise notiert, i.e. ohne Kennzeichnung der Bildung der Äquivalenzklassen.

Beispiel 14 Sei $n := 2$. Sei F die freie Gruppe auf x_1 und $x_2\,.$

In F wird

$$(x_1x_2x_2x_1^{-1}) \cdot (x_1x_2^{-1}x_1x_1x_1) \ = \ x_1x_2x_1x_1x_1\ .$$

Weitere Kürzungen sind nicht mehr möglich. Man kann nur noch zusammenfassen zu

$$x_1x_2x_1x_1x_1 \ = \ x_1x_2x_1^3\ .$$

Bemerkung 15 Sei $n \geqslant 1$. Sei F die freie Gruppe auf $x_1\,,\ x_2\,,\ \ldots\,,\ x_n\,.$

Sei T eine Gruppe. Seien $t_1\,,\ t_2\,,\ \ldots\,,\ t_n \in G$.

Es gibt genau einen Gruppenmorphismus $\varphi : F \to T$ mit $\varphi(x_k) = t_k$ für $k \in [1,n]$.

Definition 16 Sei $n \geqslant 1$. Sei F die freie Gruppe auf $x_1\,,\ x_2\,,\ \ldots\,,\ x_n\,.$

Sei $r \geqslant 1$. Seien $w_1\,,\ \ldots\,,\ w_r \in F$.

Sei N die Untergruppe von F, die von der Menge $\{\, w_k^v :\ k \in [1,r],\ v \in F \,\}$ erzeugt wird. Es ist $N \trianglelefteqslant F$.

Sei

$$G \ := \ \langle\, x_1\,,\ x_2\,,\ \ldots\,,\ x_n \ :\ w_1\,,\ \ldots\,,\ w_r \,\rangle \ := \ F/N \ = \ \{\, vN :\ v \in F \,\}$$

die Gruppe, die von den *Erzeugern* $x_1\,,\ x_2\,,\ \ldots\,,\ x_n$ modulo den *Relationen* $w_1\,,\ \ldots\,,\ w_r$ definiert ist.

Es sei $\rho : F \to G : v \mapsto vN$ der Restklassenmorphismus.

Häufig wird statt vN einfach nur v geschrieben, falls aus dem Kontext hervorgeht, daß ein Element in G gemeint ist.

Bemerkung 17 Sei $n \geqslant 1$. Sei F die freie Gruppe auf $x_1, x_2, \ldots, x_n$.

Sei $r \geqslant 1$. Seien $w_1, \ldots, w_r \in F$. Sei

$$G := \langle x_1, x_2, \ldots, x_n : w_1, \ldots, w_r \rangle .$$

Sei T eine Gruppe. Seien $t_1, t_2, \ldots, t_n \in T$.

Es gibt genau einen Gruppenmorphismus $\varphi : F \to T$ mit $\varphi(x_k) = t_k$ für $k \in [1, n]$; cf. Bemerkung 15.

Sei nun $\varphi(w_k) = 1$ für $k \in [1, r]$.

Es gibt genau einen Gruppenmorphismus $\psi : G \to T$ mit $\psi(\rho(x_k)) = t_k$ für $k \in [1, n]$.

Beispiel 18 Sei `F<x1,x2> := FreeGroup(2)`.

Sei `G<y1,y2>,rho := quo< F | x1^5, x2^3, (x1*x2)^2>`. Es ist `Order(G)` gleich 60.

Es ist `rho` der Restklassenmorphismus von `F` nach `G`, und für diesen ist nun `rho(x1)` gleich `y1`, und `rho(x2)` gleich `y2`.

In `G` kann er aber noch nicht vernünftig rechnen. E.g. erkennt er `y1^5` nicht als 1_G, obwohl dies richtig ist.

Sei daher noch `G_perm,sigma := PermutationGroup(G)`. Dann ist `G_perm` eine Untergruppe der S_6 und `sigma` ein Isomorphismus von `G` nach `G_perm`. Wir setzen `z1 := sigma(y1)` und `z2 := sigma(y2)`. Dann verschwinden nun in der Tat `z1^5`, `z2^3` und `(z1*z2)^2`.

Magma hat nun zur Konstruktion von `sigma` Gebrauch von Bemerkung 17 gemacht: Es wurde von Magma `z1` zu `(2,3,4,5,6)` und `z2` zu `(1,2,3)(4,6,5)` gewählt, wobei Magma darauf geachtet hat, daß `z1^5`, `z2^3` und `(z1*z2)^2` verschwinden. Dank Bemerkung 17 existiert ein Gruppenmorphismus von `G` nach S_6, der `y1` nach `z1` und der `y2` nach `z2` schickt. Schränkt man diesen im Zielbereich auf das Erzeugnis `G_perm` von `z1` und `z2` ein, so entsteht der surjektive Gruppenmorphismus `sigma` von `G` nach `G_perm`. Magma hat zudem darauf geachtet, daß `sigma` auch injektiv ist, insgesamt also ein Isomorphismus.

5 Berechnung von H^1 für Gruppen in Erzeugern und Relationen

Sei $n \geqslant 1$. Sei F die freie Gruppe auf $x_1, x_2, \ldots, x_n$.

Sei $r \geqslant 1$. Seien $w_1, \ldots, w_r \in F$.

Sei

$$G := \langle x_1, x_2, \ldots, x_n : w_1, \ldots, w_r \rangle .$$

Wir haben den Gruppenmorphismus

$$\begin{array}{ccc} F & \stackrel{\rho}{\to} & G \\ x_i & \mapsto & x_i \quad \text{für } i \in [1,n] \end{array}$$

Sei M ein G-Modul, der als $\mathbb{Z}$-Modul endlich erzeugt frei von Rang $\operatorname{rk} M$ sei. Wenn nötig, wird als abelsche Gruppe $M = \mathbb{Z}^{(\operatorname{rk} M)\times 1}$ identifiziert.

Via $y \cdot m := \rho(y) \cdot m$ für $y \in F$ und $m \in M$ ist M auch ein F-Modul.

Insbesondere wird $z \cdot m = \rho(z) \cdot m = 1 \cdot m = m$ für $z \in \operatorname{Kern}(\rho)$ und $m \in M$.

Bemerkung 19 Seien $m_1 , m_2 , \ldots , m_n \in M$ gegeben.

Es gibt genau eine Derivation $\delta \in \mathrm{Z}^1(F, M)$ mit $\delta(x_j) = m_j$ für $j \in [1,n]$.

Diese erfüllt

$$\delta(x_j^{-1}) \;=\; -x_j^{-1} \cdot \delta(x_j)$$

und

$$\delta(x_{i_1}^{\varepsilon_1} \cdot \ldots \cdot x_{i_t}^{\varepsilon_t}) \;=\; \sum_{s\in[1,t]} x_{i_1}^{\varepsilon_1} \cdot \ldots \cdot x_{i_{s-1}}^{\varepsilon_{s-1}} \cdot m_{i_s,\,\varepsilon_s}$$

für $t \geqslant 0$, für $i_1 , \ldots i_t \in [1,n]$ und für $\varepsilon_1 , \ldots , \varepsilon_n \in \{-1,+1\}$.

Dann definieren wir $m_{j,1} := m_j$ und $m_{j,-1} := -x_j^{-1} \cdot m_j$ für $j \in [1,n]$ und eine Abbildung $\delta_0 : F_0 \to M$ via

$$\delta_0(x_{i_1}^{\varepsilon_1} \cdot \ldots \cdot x_{i_t}^{\varepsilon_t}) \;:=\; \sum_{s\in[1,t]} x_{i_1}^{\varepsilon_1} \cdot \ldots \cdot x_{i_{s-1}}^{\varepsilon_{s-1}} \cdot m_{i_s,\,\varepsilon_s} \;,$$

so wird

$$\delta_0(w \cdot x_j \cdot x_j^{-1} \cdot w') = \delta_0(w \cdot w')$$

und

$$\delta_0(w \cdot x_j^{-1} \cdot x_j \cdot w') = \delta_0(w \cdot w')$$

für $w, w' \in F_0$ und $j \in [1,n]$.

Also ist $\delta : F \to M$ durch $\delta(w) := \delta_0(w)$ für $w \in F_0$ definierbar, wobei das Argument von δ vereinbarungsgemäß bereits die Äquivalenzklasse von w in F bezeichne.

Die definierende Formel gibt dann auch

$$\delta(w \cdot w') \;=\; \delta(w) + w \cdot \delta(w') \;.$$

Also ist $\delta \in \mathrm{Z}^1(F, M)$.

Die Eindeutigkeit von δ folgt aus der Tatsache, daß die angegebene Formel für δ notwendigerweise gilt.

Beispiel 20 Wir nehmen

```
G := SymmetricGroup(3);
G_fp<a,b>,phi := FPGroup(G);
print phi(a), phi(b); // gibt (1,2,3) und (1,2)
F<A,B> := FreeGroup(2);
rho := hom< F -> G_fp | [<A,a>,<B,b>] >; // Restklassenmorphismus
n := #Generators(G_fp);
```

Wir beschaffen uns G-Moduln wie folgt.

```
IM_rat := IrreducibleModules(G,Rationals());
M_rat := IM_rat[3];
rk := Dimension(M_rat);
MR := MatrixRing(Integers(),rk);
```

Es operiert `phi(a)` via der Matrix `ActionGenerator(M_rat,1)`. Es operiert `phi(b)` via der Matrix `ActionGenerator(M_rat,2)`.

Wir bilden eine Liste dieser Matrizen.

```
rep_mat := [MatrixRing(Integers(),Dimension(M_rat))!ActionGenerator(M_rat,i) :
                                                    i in [1..#Generators(G)]];
```

Damit bauen wir einen G-Modul mit Grundring $\mathbb{Z}$.

```
M := GModule(G,rep_mat);
```

Es ist dann $M = \mathbb{Z}^{2\times 1}$, worauf die gewählten Erzeuger von G mit den Matrizen aus `rep_mat` operieren.

Wir bauen uns die Multiplikation eines Elements $w \in F$ auf M.

```
prod := function(w,m)
  // w: Element von F
  // m: Element von M
 w_seq := ElementToSequence(w);
 t := #w_seq;
 index_seq := [Abs(x) : x in w_seq];
 sign_seq  := [Sign(x) : x in w_seq];
 return &*([MR!1] cat [rep_mat[index_seq[i]]^sign_seq[i] : i in [1..t]]) * m;
end function;
```

Wir testen das:

```
prod(A,Matrix([[1],[0]]));
/* gibt
[-1]
[-1]
*/
prod(A,Matrix([[0],[1]]));
/* gibt
[1]
[0]
*/
```

Das paßt auch zur Matrix `rep_mat[1]`;

Noch ein Test.

```
 w1 := B * A^-2 * B * A^-1;
 w2 := A * B^-1 * A^3;
 prod(w1,prod(w2,Matrix([[3],[5]]))) eq prod(w1 * w2,Matrix([[3],[5]])); // gibt true
```

Folgendermaßen erhalten wir für eine Matrix $U \in \mathbb{Z}^{(\mathrm{rk}\, M)\times n} = \mathbb{Z}^{2\times 2}$ eine Derivation $\delta \in \mathrm{Z}^1(F, M)$, die den ersten Erzeuger A auf die erste Spalte von U schickt und die den zweiten Erzeuger b auf die zweite Spalte von U schickt, jeweils gesehen als Elemente von M.

Sei noch `umnum := map<{-1,+1} -> {1,2} | [<1,1>, <-1,2>]>` global festgelegt.

Sei nun die angekündigte Derivation wie folgt konstruiert.

```
derivation := function(U,w)
  // U: ganzzahlige Matrix, Spalten in M, Spaltenzahl gleich Erzeugerzahl
  // w: abzubildendes Element von F
 w_seq := ElementToSequence(w);
 t := #w_seq;
 index_seq := [Abs(x) : x in w_seq];
 sign_seq  := [Sign(x) : x in w_seq];
 elt_seq := [[ColumnSubmatrix(U,index_seq[i],1) : i in [1..t]],
             [-rep_mat[index_seq[i]]^-1 * ColumnSubmatrix(U,index_seq[i],1) : i in [1..t]]];
 return &+([RMatrixSpace(Integers(),rk,1)!0] cat
           [&*([MR!1] cat [rep_mat[index_seq[j]]^sign_seq[j] : j in [1..i-1]])
            * elt_seq[umnum(sign_seq[i])][i] : i in [1..#w_seq]]);
end function;
```

Wir testen das:

```
 U := Matrix([[1,2],[3,4]]);
 w1 := B * A^-2 * B * A^-1;
 w2 := A * B^-1 * A^3;
 w3 := A^-1 * B^-1 * A^3 * B;
```

```
derivation(U,w1 * w2) eq derivation(U,w1) + prod(w1,derivation(U,w2)); // gibt true
derivation(U,w1 * w2 * w3) eq derivation(U,w1)
                            + prod(w1,derivation(U,w2))
                            + prod(w1 * w2,derivation(U,w3)); // gibt true
```

Bemerkung 21 Seien $m_1\,,\, m_2\,,\, \ldots\,,\, m_n \in M$ gegeben.

Sei $\delta \in \mathrm{Z}^1(F, M)$ mit $\delta(x_j) = m_j$ für $j \in [1, n]$.

Es gibt genau dann eine Derivation $\bar{\delta} : G \to M$ mit $\bar{\delta}(\rho(x_j)) = m_j$ für $j \in [1, n]$, wenn $\delta(w_i) = 0$ ist für $i \in [1, n]$.

Dazu setzen wir $\bar{\delta}(\rho(y)) := \delta(y)$ für $y \in F$.

Zu zeigen ist nur, daß $\bar{\delta}$ wohldefiniert ist, denn dann vererbt sich die Derivationseigenschaft von δ auf $\bar{\delta}$.

Seien also $y,\, \tilde{y} \in F$ mit $\rho(y) = \rho(\tilde{y})$ gegeben. Zu zeigen ist $\delta(y) \stackrel{!}{=} \delta(\tilde{y})$.

Sei $N \trianglelefteq F$ definiert wie in Definition 16. Es ist dann $N = \mathrm{Kern}(\rho)$.

Es ist $\rho(y^{-1} \cdot \tilde{y}) = \rho(y)^{-1} \cdot \rho(\tilde{y})$, also $z := y^{-1} \cdot \tilde{y} \in N$. Es ist $\tilde{y} = y \cdot z$. Also ist

$$\delta(\tilde{y}) \;=\; \delta(y \cdot z) \;=\; \delta(y) + y \cdot \delta(z)\ .$$

Es genügt also, $\delta(z) \stackrel{!}{=} 0$ zu zeigen.

Dazu genügt es, $\delta(w_k^v) \stackrel{!}{=} 0$ zu zeigen für $k \in [1, r]$ und $v \in F$.

Es wird

$$\begin{aligned}
\delta(w_k^v) &= \delta(v^{-1} \cdot w_k \cdot v) \\
&= \delta(v^{-1}) + v^{-1} \cdot \delta(w_k) + v^{-1} \cdot w_k \cdot \delta(v) \\
&= -v^{-1} \cdot \delta(v) + v^{-1} \cdot w_k \cdot \delta(v) \\
&= -v^{-1} \cdot \delta(v) + v^{-1} \cdot \delta(v) \\
&= 0\ .
\end{aligned}$$

Bemerkung 22 (Skizze des Vorgehens)

Um Bemerkung 21 umzusetzen, müssen wir testen, welche Matrizen $U \in \mathbb{Z}^{(\mathrm{rk}\, M) \times n}$ eine Derivation δ_U liefern, die auf allen Relationen $w_1\,,\, \ldots\,,\, w_r$ verschwindet.

Dazu benötigen wir etwas Lineare Algebra über $\mathbb{Z}$.

Wir schreiben U als Spalte $\mathrm{Col}_U \in \mathbb{Z}^{(\mathrm{rk}\, M) \cdot n \times 1}$, indem wir U zeilenweise auslesen und in die Spalte Col_U einlesen.

Wir werden uns in den unten folgenden Beispielen eine Matrix

$$Z \;\in\; \mathbb{Z}^{(\mathrm{rk}\, M) \cdot r \times (\mathrm{rk}\, M) \cdot n}$$

derart, daß $Z \cdot \mathrm{Col}_U$ genau dann verschwindet, wenn δ_U eine Derivation von G in M liefert.

Wir werden also $\mathrm{Z}^1(G,M)$ isomorph durch den Kern der $\mathbb{Z}$-linearen Abbildung

$$\begin{array}{ccc} \mathbb{Z}^{(\mathrm{rk}\,M)\cdot n\times 1} & \xrightarrow{Z\cdot(-)} & \mathbb{Z}^{(\mathrm{rk}\,M)\cdot r\times 1} \\ \mathrm{Col}_U & \mapsto & Z\cdot\mathrm{Col}_U \end{array}$$

ersetzen.

Wir werden ferner eine Matrix

$$I \;\in\; \mathbb{Z}^{(\mathrm{rk}\,M)\cdot n\times(\mathrm{rk}\,M)}$$

bauen, die einem Element $m\in M=\mathbb{Z}^{(\mathrm{rk}\,M)\times 1}$ die zu $m\in M$ gehörige innere Derivation δ_U zuweist, wobei sich für diese Matrix U

$$\mathrm{Col}_U \;=\; I\cdot m$$

ergibt.

Wir werden also $\mathrm{B}^1(G,M)$ isomorph durch das Bild der $\mathbb{Z}$-linearen Abbildung

$$\begin{array}{ccc} \mathbb{Z}^{(\mathrm{rk}\,M)\times 1} & \xrightarrow{I\cdot(-)} & \mathbb{Z}^{(\mathrm{rk}\,M)\cdot n\times 1} \\ m & \mapsto & I\cdot m \end{array}$$

ersetzen.

Da jede innere Derivation eine Derivation ist, ist $Z\cdot I=0$.

Der Elementarteilersatz liefert dann invertierbare Matrizen $S\in\mathbb{Z}^{(\mathrm{rk}\,M)\cdot r\times(\mathrm{rk}\,M)\cdot r}$ und $T\in\mathbb{Z}^{(\mathrm{rk}\,M)\cdot n\times(\mathrm{rk}\,M)\cdot n}$ derart, daß

$$D \;:=\; S\cdot Z\cdot T \;\in\; \mathbb{Z}^{(\mathrm{rk}\,M)\cdot r\times(\mathrm{rk}\,M)\cdot n}$$

eine Diagonalmatrix ist, i.e. an allen Positionen mit Zeilenindex ungleich Spaltenindex einen Nulleintrag hat.

Sei

$$I' \;:=\; T^{-1}\cdot I$$

So werden wir $\mathrm{Z}^1(G,M)$ isomorph durch den Kern von $D\cdot(-)$ und $\mathrm{B}^1(G,M)$ durch das Bild von $I'\cdot(-)$ ersetzt haben.

$$\begin{array}{ccccc} \mathbb{Z}^{(\mathrm{rk}\,M)\times 1} & \xrightarrow{I\cdot(-)} & \mathbb{Z}^{(\mathrm{rk}\,M)\cdot n\times 1} & \xrightarrow{Z\cdot(-)} & \mathbb{Z}^{(\mathrm{rk}\,M)\cdot r\times 1} \\ \| & & \wr\uparrow T\cdot(-) & & \wr\downarrow S\cdot(-) \\ \mathbb{Z}^{(\mathrm{rk}\,M)\times 1} & \xrightarrow{I'\cdot(-)} & \mathbb{Z}^{(\mathrm{rk}\,M)\cdot n\times 1} & \xrightarrow{D\cdot(-)} & \mathbb{Z}^{(\mathrm{rk}\,M)\cdot r\times 1} \end{array}$$

Der Kern von $D\cdot(-)$ besteht aus den Elementen von $\mathbb{Z}^{(\mathrm{rk}\,M)\cdot n\times 1}$, die in den Positionen 1 bis $\mathrm{rk}\,D$ einen Nulleintrag haben. Wir schreiben

$$c \;:=\; (\mathrm{rk}\,M)\cdot n-(\mathrm{rk}\,D)$$

und definieren dementsprechend folgende Blockmatrix.

$$J \;:=\; \begin{pmatrix} 0 \\ \mathrm{E}_c \end{pmatrix} \;\in\; \mathbb{Z}^{(\operatorname{rk} M)\cdot n \times c} \,.$$

Dann ist das Bild von $J \cdot (-)$ gleich dem Kern von $D \cdot (-)$.

Es ist nun $D \cdot I' = 0$. Ist also

$$I'' \;\in\; \mathbb{Z}^{c \times (\operatorname{rk} M)}$$

die Matrix, die aus I' durch Weglassen der Nullzeilen in Position 1 bis $\operatorname{rk} D$ entsteht, so ist

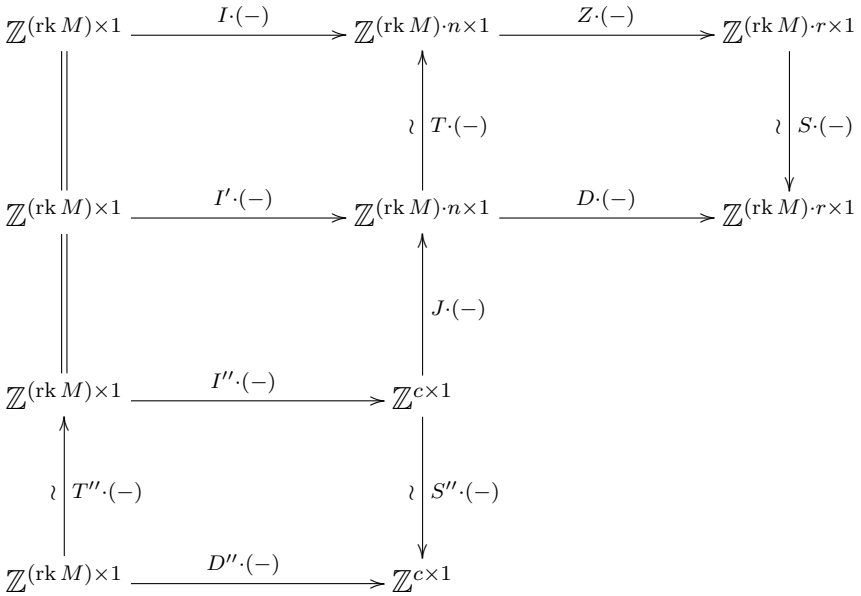
$$J \cdot I'' \;=\; I' \,.$$

Nun ist $\mathrm{Z}^1(G, M)$ isomorph ersetzt durch $\mathbb{Z}^{c\times 1}$ und $\mathrm{B}^1(G, M)$ durch das Bild von $I'' \cdot (-)$.

Der Elementarteilersatz liefert dann invertierbare Matrizen $S'' \in \mathbb{Z}^{c\times c}$ und $T'' \in \mathbb{Z}^{(\operatorname{rk} M)\times(\operatorname{rk} M)}$ derart, daß

$$D'' \;:=\; S'' \cdot I'' \cdot T'' \;\in\; \mathbb{Z}^{c \times (\operatorname{rk} M)}$$

eine Diagonalmatrix ist.



Nun ist $\mathrm{Z}^1(G, M)$ isomorph ersetzt durch $\mathbb{Z}^{c\times 1}$ und $\mathrm{B}^1(G, M)$ durch das Bild von $D'' \cdot (-)$.

Mit anderen Worten, sind $d''_1, \ldots, d''_c \in \mathbb{Z}$ die Diagonaleinträge von D'', dann ist

$$\mathrm{H}^1(G, M) \;\simeq\; \mathbb{Z}/d''_1\mathbb{Z} \oplus \ldots \oplus \mathbb{Z}/d''_c\mathbb{Z} \,.$$

Aus der Theorie weiß man auch, daß $\operatorname{rk} I + \operatorname{rk}(\operatorname{Kern}(Z \cdot (-))) = (\operatorname{rk} M) \cdot n$ ist. Wir hätten also auch direkt die Elementarteiler von I anstelle der Elemenentarteiler $d''_1, \ldots, d''_c$ von I'' heranziehen können. Wir wollen hier aber von dieser Theorieaussage keinen Gebrauch machen, sondern sie wie oben geschildert im jeweiligen Beispiel verifizieren: es sollte $\operatorname{rk} D'' = c$ sein, i.e. es sollten alle Elemente $d''_1, \ldots, d''_c$ ungleich 0 sein.

Beispiel 23 Wir setzen Beispiel 20 fort.

Es hängt $\delta_U(w) :=$ `derivation(U,w)` linear von der Matrix `U` ab, welche `rk` Zeilen und `n` Spalten hat.

Wir müssen ermitteln, welche Matrizen `U` eine Derivation δ_U liefern, die $\delta_U(w_k) = 0$ gibt für $k \in [1, n]$.

Wir besorgen uns die Relationen unserer Gruppe `G_fp`.

```
w := [x[1] : x in Relations(G_fp)];
r := #w;
```

Es ist also $w_k =$ `w[k]` für $k \in [1, r]$.

Wir ersetzen `U` durch einen Spaltenvektor `Col_U`, für welchen dann im nachhinein wieder `U[i,j] := Col_U[(i-1) * n + j]` sein soll.

Wir produzieren uns Matrizen mit nur einer 1 als Eintrag, auch Elementarmatrizen genannt.

```
U_elem := function(i,j,rk)
 U := RMatrixSpace(Integers(),rk,n)!0;
 U[i,j] := 1;
 return U;
end function;
```

Es ist nun `derivation(U,w[k])` gleich der Summe der Elemente

```
derivation(U_elem(i,j,rk),w[k]) * U[i,j] ,
```

i.e. der Elemente

```
derivation(U_elem(i,j,rk),w[k]) * Col_U[(i-1) * n + j] .
```

Wir bauen nun eine Matrix `Z` derart, daß das Produkt `Z * Col_U` genau dann verschwindet, wenn `derivation(U,w[k])` verschwindet für alle $k \in [1, r]$.

```
Z := RMatrixSpace(Integers(),rk*r,rk*n)!0;
for k in [1..r] do
 for i in [1..rk] do
  for j in [1..n] do
   InsertBlock(~Z,derivation(U_elem(i,j,rk),w[k]), 1 + (k-1)*rk, (i-1) * n + j);
  end for;
 end for;
end for;
```

```
print Z;
/* gibt
[ 0  1  0  1]
[ 0  1  0  1]
[ 0  0  0  0]
[ 0  0  0  0]
[ 0  0  0  0]
[-2  2  1 -1]
*/
```

Nun noch die inneren Derivationen.

```
U_inn := function(m) // m in M
 U := RMatrixSpace(Integers(),rk,n)!0;
 for j in [1..n] do
  InsertBlock(~U, rep_mat[j] * m - m, 1, j);
 end for;
 return U;
end function;
```

Wir lassen `m` über eine Basis von M laufen und schreiben die resultierenden Matrizen `U` als Spalten in eine Matrix `I`.

```
I := RMatrixSpace(Integers(),rk*n,rk)!0;
for k in [1..rk] do
 m := RMatrixSpace(Integers(),rk,1)!0;
 m[k,1] := 1;
 U := U_inn(m);
 for i in [1..rk] do
  for j in [1..n] do
   I[(i-1) * n + j,k] := U[i,j];
  end for;
 end for;
end for;
print I;
/*
[-2  1]
[-1  1]
[-1 -1]
[ 1 -1]
*/
```

Innere Derivationen sollten Derivationen sein: in der Tat gibt `Z * I` die Nullmatrix.

Sei nun

```
D,S,T := SmithForm(Z);
```

Dann sind `S` und `T` invertierbare ganzzahlige Matrizen mit `S * Z * T = D` diagonal. Bei uns:

```
> D;
[1 0 0 0]
[0 1 0 0]
[0 0 0 0]
[0 0 0 0]
[0 0 0 0]
[0 0 0 0]
> S;
[ 1  0  0  0  0  0]
[-2  0  0  0  0  1]
[ 0  0  1  0  0  0]
[ 0  0  0  1  0  0]
[ 0  0  0  0  1  0]
[-1  1  0  0  0  0]
> T;
[ 0  0  1  0]
[ 1  0  0 -1]
[ 0  1  2  3]
[ 0  0  0  1]
```

Es ist genau dann $Zx = 0$, wenn $D(T^{-1}x) = SZTT^{-1}x = 0$ ist. Dies ist genau dann der Fall, wenn $T^{-1}x$ mit `rkD := Rank(D)` Nulleinträgen beginnt.

So etwa ist `T^-1 * I` eine Matrix mit Nullen in den den ersten beiden Zeilen.

Sei `II := RowSubmatrixRange(T^-1 * I, rkD + 1, rk * n)`.

Sei

```
DD,SS,TT := SmithForm(II);
print DD;
/*
[1 0]
[0 1]
*/
```

Folglich ist $\mathrm{B}^1(G, M) = \mathrm{Z}^1(G, M)$ und also

$$\mathrm{H}^1(G, M) \;=\; 0\,.$$

Wir vergleichen das Resultat noch kurz mit dem des in Magma eingebauten Befehls. Vorsicht, Magma arbeitet mit Rechtsmoduln, daher muß zwecks Vergleich dualisiert werden.

```
rep_mat_dual := [Transpose(x)^-1 : x in rep_mat];
CM := CohomologyModule(G,GModule(G,rep_mat_dual));
CohomologyGroup(CM,1);
```

```
/*
Full Quotient RSpace of degree 0 over Integer Ring
Column moduli:
[ ]
*/
```

Beispiel 24 Wir variieren Beispiel 20, 23.

Der Vorspann war folgender.

```
G := SymmetricGroup(3);
G_fp<a,b>,phi := FPGroup(G);
print phi(a), phi(b); // gibt (1,2,3) und (1,2)
F<A,B> := FreeGroup(2);
rho := hom< F -> G_fp | [<A,a>,<B,b>] >; // Restklassenmorphismus
n := #Generators(G_fp);
IM_rat := IrreducibleModules(G,Rationals());
M_rat := IM_rat[3];
rk := Dimension(M_rat);
MR := MatrixRing(Integers(),rk);
rep_mat := [MR!ActionGenerator(M_rat,i) : i in [1..#Generators(G)]];
M := GModule(G,rep_mat);

prod := function(w,m)
 w_seq := ElementToSequence(w);
 t := #w_seq;
 index_seq := [Abs(x) : x in w_seq];
 sign_seq := [Sign(x) : x in w_seq];
 return &*([MR!1] cat [rep_mat[index_seq[i]]^sign_seq[i] : i in [1..t]]) * m;
end function;

umnum := map<{-1,+1} -> {1,2} | [<1,1>, <-1,2>]>;
w := [x[1] : x in Relations(G_fp)];
r := #w;

U_elem := function(i,j,rk)
 U := RMatrixSpace(Integers(),rk,n)!0;
 U[i,j] := 1;
 return U;
end function;
```

Wir ersetzen nun M durch den zu M dualen Modul M^*. Dazu müssen wir ersetzen:

```
rep_mat_M := rep_mat;
rep_mat_M_dual := [Transpose(x)^-1 : x in rep_mat_M];
rep_mat := rep_mat_M_dual;
```

Wir wiederholen:

```
derivation := function(U,w)
 w_seq := ElementToSequence(w);
 t := #w_seq;
 index_seq := [Abs(x) : x in w_seq];
 sign_seq  := [Sign(x) : x in w_seq];
 elt_seq := [[ColumnSubmatrix(U,index_seq[i],1) : i in [1..t]],
             [-rep_mat[index_seq[i]]^-1 * ColumnSubmatrix(U,index_seq[i],1) : i in [1..t]]];
 return &+([RMatrixSpace(Integers(),rk,1)!0] cat
           [&*([MR!1] cat [rep_mat[index_seq[j]]^sign_seq[j] : j in [1..i-1]])
            * elt_seq[umnum(sign_seq[i])][i] : i in [1..#w_seq]]);
end function;


Z := RMatrixSpace(Integers(),rk*r,rk*n)!0;
for k in [1..r] do
 for i in [1..rk] do
  for j in [1..n] do
   InsertBlock(~Z,derivation(U_elem(i,j,rk),w[k]), 1 + (k-1)*rk, (i-1) * n + j);
  end for;
 end for;
end for;
print Z;
/* gibt
[ 0  1  0  1]
[ 0  1  0  1]
[ 0  0  0  0]
[ 0  0  0  0]
[ 1 -1  0  0]
[-2  2  0  0]
*/


U_inn := function(m) // m in M
 U := RMatrixSpace(Integers(),rk,n)!0;
 for j in [1..n] do
  InsertBlock(~U, rep_mat[j] * m - m, 1, j);
 end for;
 return U;
end function;


I := RMatrixSpace(Integers(),rk*n,rk)!0;
for k in [1..rk] do
 m := RMatrixSpace(Integers(),rk,1)!0;
 m[k,1] := 1;
 U := U_inn(m);
 for i in [1..rk] do
  for j in [1..n] do
   I[(i-1) * n + j,k] := U[i,j];
  end for;
 end for;
end for;
```

```
print I;
/*
[-1  1]
[-1  1]
[-1 -2]
[ 1 -1]
*/


D,S,T := SmithForm(Z);
rkD := Rank(D);
II := RowSubmatrixRange(T^-1 * I, rkD + 1, rk * n);
DD,SS,TT := SmithForm(II);
print DD;
/*
[1 0]
[0 3]
*/
```

Also ist $\mathrm{H}^1(G, M^*) \simeq \mathbb{Z}/3\mathbb{Z}$.

Sei noch

```
J := RMatrixSpace(Integers(), rk*n, rk*n - rkD)!0;
InsertBlock(~J, MatrixRing(Integers(), rk*n - rkD)!1, rkD+1, 1);
```

die Einbettungsmatrix. Dann ist `J * II eq T^-1 * I`.

Wir wollen nun noch einen Erzeuger dieser Gruppe von einer Derivation repräsentieren lassen.

Dem Element $\binom{0}{1}$, also `e2 := Matrix([[0],[1]])`, entspricht das Element `T * J * SS^-1 * e2`, also

```
/*
[ 0]
[-1]
[ 3]
[ 1]
*/ .
```

Übersetzt wird diese Spalte zur folgenden Matrix.

```
U_rep := Matrix([[0,-1],[3,1]])
```

Wir bauen daraus eine Derivation auf G.

```
d_rep := function(U_rep,g);
 return derivation(U_rep,(G!g)@@phi@@rho);
end function;
```

Wir testen, ob `d_rep` mittels `U_rep` tatsächlich eine Derivation auf G liefert.

```
M_dual := GModule(G,rep_mat);
rep := Representation(M_dual);
&and[d_rep(U_rep,g) + rep(g) * d_rep(U_rep,h) eq d_rep(U_rep,g*h) : g,h in G];
/*
true
*/
print [<g, d_rep(U_rep,g)> : g in G];
// alle Gruppenelemente samt Bildern
/*
[
    <
        Id(G),

        [0]
        [0]
    >,

    <
        (1, 2, 3),

        [0]
        [3]
    >,

    <
        (1, 3, 2),

        [3]
        [3]
    >,

    <
        (2, 3),

        [2]
        [4]
    >,

    <
        (1, 2),

        [-1]
        [ 1]
    >,

    <
        (1, 3),

        [2]
        [1]
    >
```

```
]
*/
```

Beispiel 25 Wir variieren Beispiel 24 und wollen die erste Cohomologiegruppe des regulären Moduls $\mathbb{Z}G$ bestimmen; cf. Beispiel 3.

Es ist aus der Theorie bekannt, daß $\mathrm{H}^1(G, \mathbb{Z}G) = 0$ ist. Wir wollen hier also die Übereinstimmung von Theorie und Praxis prüfen.

Für eine Permutationsgruppe G mit festgelegter Auflistung `G_list` erhalten wir die beschreibende Matrix eines Elements $x \in G$ wie folgt.

```
rep_mat_reg := function(G,G_list,x)
// G: Permutationsgruppe
// x: Element von G
 n := Order(G);
 A := MatrixRing(Integers(),n)!0;
 for i in [1..n] do
  A[Index(G_list, x * G_list[i]), i] := 1;
 end for;
 return A;
end function;
```

Wie in Beispiel 24 erhalten wir nun nach Eingabe des dortigen Vorspanns folgendes.

```
G_list := [x : x in G];
rep_mat := [rep_mat_reg(G,G_list,phi(a)),rep_mat_reg(G,G_list,phi(b))];
M_reg := GModule(G,rep_mat);
rk := Order(G);
MR := MatrixRing(Integers(),rk);
```

Nach weiterem Vorgehen wie in Beispiel 24 erhalten wir folgendes.

```
DD;
/*
[1 0 0 0 0 0]
[0 1 0 0 0 0]
[0 0 1 0 0 0]
[0 0 0 1 0 0]
[0 0 0 0 1 0]
*/
```

Also ist in der Tat $\mathrm{H}^1(G, \mathbb{Z}G) = 0$.

6 2-Cozykel und H^2

Sei G eine endliche Gruppe.

Sei M ein G-Modul.

Derivationen von G in M werden alternativ auch 1-*Cozykel* genannt.

Im nächsten Schritt definieren wir nun 2-Cozykel.

Definition 26 Ein 2-*Cozykel* von G in M ist eine Abbildung

$$\zeta \ : \ G \times G \ \to \ M$$

welche

$$g \cdot \zeta(g', g'') - \zeta(g \cdot g', g'') + \zeta(g, g' \cdot g'') - \zeta(g, g') \ = \ 0$$

erfüllt für g, g', $g'' \in G$.

Wir schreiben

$$\mathrm{Z}^2(G, M)$$

für die Menge der 2-Cozykel von G in M.

Es ist $\mathrm{Z}^2(G, M)$ eine Untergruppe der abelschen Gruppe aller Abbildungen von $G \times G$ nach M.

Insbesondere ist $\mathrm{Z}^2(G, M)$ eine abelsche Gruppe.

Definition 27 Sei $f : G \to M$ eine Abbildung.

Sei die Abbildung $\zeta_f^{\mathrm{inn}} : G \times G \to M$ definiert durch

$$\zeta_f^{\mathrm{inn}}(g, g') \ := \ g \cdot f(g') - f(g \cdot g') + f(g)$$

für g, $g' \in G$.

Es ist ζ_m^{inn} ein 2-Cozykel, da

$$\begin{array}{rl} & g \cdot \zeta_f^{\mathrm{inn}}(g', g'') - \zeta_f^{\mathrm{inn}}(g \cdot g', g'') + \zeta_f^{\mathrm{inn}}(g, g' \cdot g'') - \zeta_f^{\mathrm{inn}}(g, g') \\ = & g \cdot (g' \cdot f(g'') - f(g' \cdot g'') + f(g')) \\ & -(g \cdot g' \cdot f(g'') - f(g \cdot g' \cdot g'') + f(g \cdot g')) \\ & +(g \cdot f(g' \cdot g'') - f(g \cdot g' \cdot g'') + f(g)) \\ & -(g \cdot f(g') - f(g \cdot g') + f(g)) \\ = & 0 \end{array}$$

ist für g, g', $g'' \in G$. Ein Derivation der Form ζ_f^{inn} für ein f heißt auch *innerer* 2-*Cozykel*.

Wir schreiben

$$\mathrm{B}^2(G, M) \ := \ \{ \, \zeta_m^{\mathrm{inn}} : \ m \in M \, \} \ \subseteq \ \mathrm{Z}^1(G, M)$$

für die Menge der inneren Derivationen.

Es ist $\mathrm{B}^2(G, M)$ eine Untergruppe von $\mathrm{Z}^2(G, M)$.

Definition 28 Die Faktorgruppe

$$\mathrm{H}^2(G,M) \;:=\; \mathrm{Z}^2(G,M)/\mathrm{B}^2(G,M)$$

heißt *zweite Cohomologiegruppe* von G in M.

7 Zusammenhang von H^2 und H^1

Sei G eine endliche Gruppe.

Sei M ein G-Modul, der als $\mathbb{Z}$-Modul endlich erzeugt frei von Rang $\mathrm{rk}\, M$ sei. Wenn nötig, wird als abelsche Gruppe $M = \mathbb{Z}^{(\mathrm{rk}\, M)\times 1}$ identifiziert.

Operiere $g \in G$ vermittels $\gamma(g) \in \mathbb{Z}^{(\mathrm{rk}\, M)\times(\mathrm{rk}\, M)}$ auf M.

Definition 29 Wir definieren die abelsche Gruppe

$$Q \;:=\; \mathbb{Q}/\mathbb{Z}\,.$$

Sei

$$M_Q := Q^{(\mathrm{rk}\, M)\times 1}\,.$$

Operiere $g \in G$ vermittels $\gamma(g) \in \mathbb{Z}^{(\mathrm{rk}\, M)\times(\mathrm{rk}\, M)}$ auf M_Q.

Es wird so M_Q zu einem G-Modul.

Bemerkung 30 Es ist

$$\mathrm{H}^2(G,M) \;\simeq\; \mathrm{H}^1(G,M_Q)\,.$$

8 Berechnung von H^2 für Gruppen in Erzeugern und Relationen

Das Setup sei wie in §5:

Sei $n \geqslant 1$. Sei F die freie Gruppe auf $x_1\,,\, x_2\,,\, \ldots\,,\, x_n$.

Sei $r \geqslant 1$. Seien $w_1\,,\, \ldots\,,\, w_r \,\in\, F$.

Sei

$$G \;:=\; \langle\, x_1\,,\, x_2\,,\, \ldots\,,\, x_n \;:\; w_1\,,\, \ldots\,,\, w_r \,\rangle\,.$$

Sei M ein G-Modul, der als $\mathbb{Z}$-Modul endlich erzeugt frei von Rang $\mathrm{rk}\, M$ sei. Wenn nötig, wird als abelsche Gruppe $M = \mathbb{Z}^{(\mathrm{rk}\, M)\times 1}$ identifiziert.

Wir wollen $\mathrm{H}^2(G,M) \simeq \mathrm{H}^1(G,M_Q)$ berechnen; cf. Bemerkung 30.

Bemerkung 31 (Skizze des Vorgehens)

Seien

$$Z \ \in \ \mathbb{Z}^{(\mathrm{rk}\, M)\cdot r \times (\mathrm{rk}\, M)\cdot n}$$

und

$$I \ \in \ \mathbb{Z}^{(\mathrm{rk}\, M)\cdot n \times (\mathrm{rk}\, M)}$$

wie in Bemerkung 22.

Seien wieder invertierbare Matrizen $S \in \mathbb{Z}^{(\mathrm{rk}\, M)\cdot r \times (\mathrm{rk}\, M)\cdot r}$ und $T \in \mathbb{Z}^{(\mathrm{rk}\, M)\cdot n \times (\mathrm{rk}\, M)\cdot n}$ derart gefunden, daß

$$D \ := \ S \cdot Z \cdot T \ \in \ \mathbb{Z}^{(\mathrm{rk}\, M)\cdot r \times (\mathrm{rk}\, M)\cdot n}$$

eine Diagonalmatrix ist. Sei $I' \ := \ T^{-1} \cdot I$.

Dies gibt

$$\begin{array}{ccccc}
Q^{(\mathrm{rk}\, M)\times 1} & \xrightarrow{I\cdot(-)} & Q^{(\mathrm{rk}\, M)\cdot n\times 1} & \xrightarrow{Z\cdot(-)} & Q^{(\mathrm{rk}\, M)\cdot r\times 1} \\
\| & & \wr\uparrow T\cdot(-) & & \wr\downarrow S\cdot(-) \\
Q^{(\mathrm{rk}\, M)\times 1} & \xrightarrow{I'\cdot(-)} & Q^{(\mathrm{rk}\, M)\cdot n\times 1} & \xrightarrow{D\cdot(-)} & Q^{(\mathrm{rk}\, M)\cdot r\times 1}
\end{array}$$

Es ist $\mathrm{Z}^1(G, M_Q)$ isomorph ersetzt durch den Kern von $D(-)$.

Es ist $\mathrm{B}^1(G, M_Q)$ isomorph ersetzt durch das Bild von $I'(-)$.

Wir schreiben $b := \mathrm{rk}\, D$.

Seien die nichtverschwindenden Diagonaleinträge von D durch $d_1, \ldots, d_b \in \mathbb{Z}$ gegeben.

Der Kern von $D(-)$ ist isomorph zu

$$\mathbb{Z}/d_1\mathbb{Z} \oplus \ldots \oplus \mathbb{Z}/d_b\mathbb{Z} \oplus Q^{(\mathrm{rk}\, M)\cdot n - b}$$

Aus der Theorie entnehmen wir, daß der Rang von I' gleich $(\mathrm{rk}\, M) \cdot n - b$ ist. Also ergibt sich das Bild von $I'(-)$ zu $Q^{(\mathrm{rk}\, M)\cdot n - b}$, welches auch auf diesen Summanden des vorstehenden Ausdrucks abgebildet wird.

Es folgt

$$\mathrm{H}^2(G, M) \ \simeq \ \mathrm{H}^1(G, M_Q) \ \simeq \ \mathbb{Z}/d_1\mathbb{Z} \oplus \ldots \oplus \mathbb{Z}/d_b\mathbb{Z}$$

Beispiel 32 Vorgehen wie in Beispiel 23 :

```
G := SymmetricGroup(3);
G_fp<a,b>,phi := FPGroup(G);
print phi(a), phi(b); // gibt (1,2,3) und (1,2)
F<A,B> := FreeGroup(2);
rho := hom< F -> G_fp | [<A,a>,<B,b>] >; // Restklassenmorphismus
```

```
n := #Generators(G_fp);
IM_rat := IrreducibleModules(G,Rationals());
M_rat := IM_rat[3];
rk := Dimension(M_rat);
MR := MatrixRing(Integers(),rk);
rep_mat := [MR!ActionGenerator(M_rat,i) : i in [1..#Generators(G)]];
M := GModule(G,rep_mat);

prod := function(w,m)
 w_seq := ElementToSequence(w);
 t := #w_seq;
 index_seq := [Abs(x) : x in w_seq];
 sign_seq := [Sign(x) : x in w_seq];
 return &*([MR!1] cat [rep_mat[index_seq[i]]^sign_seq[i] : i in [1..t]]) * m;
end function;

umnum := map<{-1,+1} -> {1,2} | [<1,1>, <-1,2>]>;
w := [x[1] : x in Relations(G_fp)];
r := #w;

U_elem := function(i,j,rk)
 U := RMatrixSpace(Integers(),rk,n)!0;
 U[i,j] := 1;
 return U;
end function;



derivation := function(U,w)
 w_seq := ElementToSequence(w);
 t := #w_seq;
 index_seq := [Abs(x) : x in w_seq];
 sign_seq  := [Sign(x) : x in w_seq];
 elt_seq := [[ColumnSubmatrix(U,index_seq[i],1) : i in [1..t]],
             [-rep_mat[index_seq[i]]^-1 * ColumnSubmatrix(U,index_seq[i],1) : i in [1..t]]];
 return &+([RMatrixSpace(Integers(),rk,1)!0] cat
           [&*([MR!1] cat [rep_mat[index_seq[j]]^sign_seq[j] : j in [1..i-1]])
            * elt_seq[umnum(sign_seq[i])][i] : i in [1..#w_seq]]);
end function;



Z := RMatrixSpace(Integers(),rk*r,rk*n)!0;
for k in [1..r] do
 for i in [1..rk] do
  for j in [1..n] do
   InsertBlock(~Z,derivation(U_elem(i,j,rk),w[k]), 1 + (k-1)*rk, (i-1) * n + j);
  end for;
 end for;
end for;
```

```
print Z;
/* gibt wieder
[ 0  1  0  1]
[ 0  1  0  1]
[ 0  0  0  0]
[ 0  0  0  0]
[ 0  0  0  0]
[-2  2  1 -1]
*/



D,S,T := SmithForm(Z);
print D;
/*
[1 0 0 0]
[0 1 0 0]
[0 0 0 0]
[0 0 0 0]
[0 0 0 0]
[0 0 0 0]
*/
```

Folglich ist

$$\mathrm{H}^2(G,M) \;\simeq\; \mathrm{H}^1(G,M_Q) \;\simeq\; \mathbb{Z}/1\mathbb{Z} \oplus \mathbb{Z}/1\mathbb{Z} \;=\; 0\,.$$

Wir betrachten noch einen anderen Modul $M' =$ `MM`.

```
MM_rat := IM_rat[2];
rk := Dimension(MM_rat);
MR := MatrixRing(Integers(),rk);
rep_mat := [MR!ActionGenerator(MM_rat,i) : i in [1..#Generators(G)]];
MM := GModule(G,rep_mat);



derivation := function(U,w)
 w_seq := ElementToSequence(w);
 t := #w_seq;
 index_seq := [Abs(x) : x in w_seq];
 sign_seq  := [Sign(x) : x in w_seq];
 elt_seq := [[ColumnSubmatrix(U,index_seq[i],1) : i in [1..t]],
             [-rep_mat[index_seq[i]]^-1 * ColumnSubmatrix(U,index_seq[i],1) : i in [1..t]]];
 return &+([RMatrixSpace(Integers(),rk,1)!0] cat
           [&*([MR!1] cat [rep_mat[index_seq[j]]^sign_seq[j] : j in [1..i-1]])
            * elt_seq[umnum(sign_seq[i])][i] : i in [1..#w_seq]]);
end function;
```

```
Z := RMatrixSpace(Integers(),rk*r,rk*n)!0;
for k in [1..r] do
 for i in [1..rk] do
  for j in [1..n] do
   InsertBlock(~Z,derivation(U_elem(i,j,rk),w[k]), 1 + (k-1)*rk, (i-1) * n + j);
  end for;
 end for;
end for;
print Z;
/* gibt
[ 0  0]
[-3  0]
[ 0  0]

*/


D,S,T := SmithForm(Z);
print D;
/*
[3 0]
[0 0]
[0 0]
*/
```

Folglich ist

$$\mathrm{H}^2(G, M') \;\simeq\; \mathbb{Z}/3\mathbb{Z}\,.$$

Literatur

[1] Bosma, W.; Cannon, J.J.; Fieker, C.; Steel, A. (eds.), *Handbook of Magma functions,* Edition 2.16, 2010; cf. magma.maths.usyd.edu.au, magma.maths.usyd.edu.au/calc.

[2] Eick, B.; Holt, D.; O'Brien, E., *Handbook of Computational Group Theory*, Chapman & Hall, 2005.

[3] Künzer, M., *Cohomologie von Gruppen*, Skript, `pnp.mathematik.uni-stuttgart.de/lexmath/kuenzer/manuscripts.html`, 2006.

[4] Weber, C., *Kohomologie von Spechtmoduln*, Diplomarbeit, RWTH Aachen, 2006.

[5] Weber, C., *Low-degree Cohomology of Integral Specht Modules*, arxiv:0905.4209v1, 2009.

[6] Zassenhaus, H., *Über einen Algorithmus zur Bestimmung der Raumgruppen,* Comm. Math. Helv. 21, p. 117–141, 1948.