DOCUMENTATION FOR `GAMMA_GRADED_CHARACTER_RINGS`

ALEXANDER ZIEGLER

1. Introduction

Suppose one wants to compute the Chow ring $\mathsf{CH}^*(\mathsf{BG})$ of the motivic classifying space of a finite group G. Possible first approximations are $\mathsf{gr}^*_{\mathrm{geom}}\mathsf{R}(\mathsf{G})$ or, a bit further removed, $\mathsf{gr}^*_\gamma\mathsf{R}(\mathsf{G})$, the associated graded algebras to the geometric and the γ-filtration on the representation ring of G denoted by $\mathsf{R}(\mathsf{G})$. We will not elaborate on the precise meaning of the term "approximation" here but refer the reader to [Zie25]. Computing each of the $\mathsf{gr}^i_\gamma\mathsf{R}(\mathsf{G})$, including the ring structure, is a task that is straightforward while still incredibly labor intensive, a task well suited for a computer algebra system. Under good circumstances, we can even compute $\mathsf{gr}^i_{\mathrm{geom}}\mathsf{R}(\mathsf{G})$ by restricting to subgroups. In the following document, we will describe the usage of the functions contained in `gamma_graded_character_rings` in order to compute $\mathsf{gr}^*_\gamma\mathsf{R}(\mathsf{G})$ or under good circumstances $\mathsf{gr}^*_{\mathrm{geom}}\mathsf{R}(\mathsf{G})$. All further details can be found in the comments contained in `gamma.sage` and `gamma-filtration.g`. The mathematical background is laid out in [Zie25].

2. Intended Usage

To access the tools presented, simply open [SAGE] and load the file `gamma.sage`. This section is a rundown on how to use the functions

- `presentation`,
- `presentation_with_classes`,
- `presentation_with_restriction` and
- `presentation_with_restriction_and_classes`

which are the intended interface to the code in `gamma_graded_character_rings`. Laying things out roughly,

`presentation`

computes all the $\mathsf{gr}^i_\gamma\mathsf{R}(\mathsf{G})$, including the ring structure, for $i \leqslant n$ where n is some natural number specified by the user.

`presentation_with_restriction`

computes $\mathsf{gr}^n_\gamma\mathsf{R}(\mathsf{G})$ where n is some natural number specified by the user, as well as non-zero terms in $\mathsf{gr}^n_\gamma\mathsf{R}(\mathsf{G})$ that restrict to zero on all members of some family of subgroups $\mathsf{H}_i \leqslant \mathsf{G}$ specified by the user. This can be helpful to identify non-zero terms that might vanish in $\mathsf{gr}^n_{\mathrm{geom}}\mathsf{R}(\mathsf{G})$, by restricting to subgroups where $\mathsf{gr}^n_{\mathrm{geom}}\mathsf{R}(\mathsf{H}_i) = \mathsf{gr}^n_\gamma\mathsf{R}(\mathsf{H}_i)$.

Date: Februar 2025.

`presentation_with_classes`

computes graded pieces of an associated graded algebra $\bigoplus_i F^i/F^{i+1}$ for $i \leqslant n$ where n is some natural number specified by the user. Here, the F^i are the coarsest filtration on $R(G)$ such that $\Gamma_i \subseteq F^i$, $F^i \cdot F^j \subseteq F^{i+j}$ and F^i contains some elements of $R(G)$ specified by the user. Ideally $F^i = F^i_{\text{geom}}$ where the latter denotes the i-th step in the geometric filtration on $R(G)$.

`presentation_with_restriction_and_classes`

computes a group F^n/F^{n+1} where n is some natural number specified by the user, as well as non-zero terms in F^n/F^{n+1} that restrict to zero in $R(H_i)/F^{n+1}_{H_i}$. Here, the F_i are a filtration on $R(G)$ as described above. Similarly, the $F^{n+1}_{H_i}$ are the smallest ideals in $R(H_i)$ that contain the image of F^{n+1} under restriction and some additional elements of $R(H_i)$ as specified by the user.

We will now go into more detail on how to use each function, including some examples.

Function 2.1. The function

`presentation(group, n, upper_bound, modified_strong_gens)`

takes as parameters

- `group` - a finite group given as a [GAP4]-object,
- `n` - a positive integer that is the highest degree in which relations are computed,
- `upper_bound` - a positive integer that is an upper bound (with respect to divisibility) for the additive order of elements of $\mathrm{gr}^i_\gamma R(\texttt{group})$ where $i \leqslant \texttt{n}$,
- `modified_strong_gens` - either `[false` or `[true, x]` where `x` is an ascending list of integers that correspond to generators of $\mathrm{gr}^*_\gamma R(\texttt{group})$.

Clearly this function performs better for smaller `n`, but choosing a smaller value for `upper_bound` should not make a big difference, in particular if it is a power of a prime number. A priori one would use `[false]` for the parameter `modified_strong_gens`, which means that we do not exclude any generators. After running `presentation` this way, the resulting relations might indicate redundancies in the set of generators that is computed. Take `x` to be the ascending list of positions of non-redundant generators in the list of all computed generators. Running the function with `[true, x]` for `modified_strong_gens` can increase the performance for higher `n` significantly. We will give an example later. Alternatively, one can take `x` to be any list of positions of generators. Then only the relations between the products of these generators will be computed. Thanks to [Che20, Proposition 2.6] the order of `group` is always a sensible choice for `upper_bound`.

The function returns a list whose three entries are

- if `modified_strong_gens=[false` a list of generators of $\mathrm{gr}^*_\gamma R(\texttt{group})$, if `modified_strong_gens=[true,x]` the sublist of entries at positions specified in `x`,
- a list of corresponding Chern classes given in the form `[1,[i,j]]` which represent the `j`-th Chern class of the `i`-th character in the character table of `group`,
- a generating set of the ideal containing all relations, up to degree `n`, of a presentation of $\mathrm{gr}^*_\gamma R(\texttt{group})$ with respect to the aforementioned (sub-)list of generators.

Example 2.2. In this example we'll denote in mathematical formulas

$$G := \texttt{gap.SmallGroup}(64, 138)$$

which can be identified with $\mathrm{Syl}_2(\mathrm{GL}_4(\mathbb{F}_2))$. Executing

```
presentation(gap.SmallGroup(64,138),1,64,[false])
```

returns a list of generators

```
[x4, x5, x6, x7, x9, x11, x13, x15, x17, x19, x21, x23, x25]
```

identified with a list of abstract Chern classes

```
[[1, [6, 1]], [1, [7, 1]], [1, [8, 1]], [1, [9, 1]], [1, [11, 1]],
[1, [13, 1]], [1, [15, 1]], [1, [9, 2]], [1, [11, 2]], [1, [13, 2]],
[1, [15, 2]], [1, [15, 3]], [1, [15, 4]]]
```

as well as relations

```
[2*x13, 2*x11, 2*x9, x7 + x9 + x11 + x13, x6 + x11, x5 + x11 + x13,
x4 + x9].
```

Only the first seven generators are identified with first Chern classes. The relations tell us $\mathsf{gr}^1_\gamma \mathsf{R}(\mathsf{G}) \cong \langle \mathtt{x4}, \mathtt{x5}, \mathtt{x6}\rangle_{\mathbb{F}_2}$ is just an $\mathbb{F}_2$-vector space of dimension 3. The remaining generators of degree one are redundant. Thus we can improve performance by forcibly dropping these generators. The list of positions of non-redundant generators is `[0,1,2,7,8,9,10,11,12]`. We call now

```
presentation(gap.SmallGroup(64,138),2,64,
[true,[0,1,2,7,8,9,10,11,12]])
```

and get relations

```
[4*x21, 4*x19, 4*x17, x15 + x17 + x19 + 2*x21, 2*x6, 2*x5, 2*x4,
x5^2 + x5*x6, x4*x5 + x4*x6 + x5*x6 + x6^2].
```

Notably `x19` is redundant and from the present relations we can compute

$$\mathsf{gr}^2_\gamma \mathsf{R}(\mathsf{G}) \cong (\mathbb{Z}/4\mathbb{Z})^2 \oplus (\mathbb{Z}/2\mathbb{Z})^5.$$

As `x19` is a position 9 in our original list of generators we exclude 9 from the parameter `modified_strong_gens` and call on

```
presentation(gap.SmallGroup(64,138)),3,64,
[true,[0,1,2,7,8,10,11,12]])
```

to obtain relations

```
[2*x23, 4*x21, 4*x17, 4*x15, 2*x6, 2*x5, 2*x4, x5*x23, x4*x23 +
x6*x23, x5*x21 + x6*x21 + x23, x4*x17, x6*x15 + x6*x17, x4*x15 +
x5*x15, x5^2 + x5*x6, x4*x5 + x4*x6 + x5*x6 + x6^2, x5*x15*x17,
x6^2*x17, x5*x6*x17, x5*x6^2 + x5*x15 + x5*x17 + x6*x21 + x23,
x4^2*x6 + x4*x6^2, x4^3 + x6^3 + x5*x15 + x6*x17 + x4*x21 + x6*x21].
```

We see that we can drop the generator `x23` and the relation

```
x5*x6^2 + x5*x15 + x5*x17 + x6*x21 + x23.
```

This yields all relations up to degree 3 that form a presentation of $\mathsf{gr}^*_\gamma \mathsf{R}(\mathsf{G})$ as generated by the Chern classes described by `[x4, x5, x6, x15, x17, x21, x25]` and one could continue to search for relations of higher degree, given sufficient computational power. We'll omit a concrete description of the isomorphism type of $\mathsf{gr}^3_\gamma \mathsf{R}(\mathsf{G})$. Be aware that there might be redundancies among those relations. We invite the reader to compare this example to the computation of $\mathsf{CH}^*(\mathsf{BG})/2$ in [Zie25].

Function 2.3. Some additional functionality is provided by

```
presentation_with_restriction(group, n, upper_bound,
        modified_strong_gens, subgroups)
```

whose parameters are

- `group` - a finite group given as a [GAP4] object,
- `n` - a positive integer that is the degree in which relations are computed,
- `upper_bound` - a positive integer that is an upper bound (with respect to divisibility) for the additive order of elements of $\mathsf{gr}^{\mathsf{n}}_{\gamma}\mathsf{R}(\mathtt{group})$,
- `modified_strong_gens` - either `[false]` or `[true, x]` where `x` is an ascending list of integers that correspond to generators of $\mathsf{R}(\mathtt{group})$,
- `subgroups` - a list of integers that correspond to the positions of conjugacy classes of subgroups of `group` as provided by the [GAP4]-function `ConjugacyClassesSubgroups(group);` such that the intersection of the kernels of restrictions to representatives of those conjugacy classes is computed.

The parameter `modified_strong_gens` works as for the function `presentation`.

The function returns a list whose five entries are

- if `modified_strong_gens=[false]` a list of generators of $\mathsf{gr}^{*}_{\gamma}\mathsf{R}(\mathtt{group})$, if `modified_strong_gens=[true,x]` the sublist of entries at positions specified in `x`,
- a list of corresponding Chern classes given in the form `[1,[i,j]]` which represent the `j`-th Chern class of the `i`-th character in the character table of `group`,
- a generating set of the ideal containing all relations in degree `n` (sometimes also some redundant relations of higher degree) of a presentation of $\mathsf{gr}^{*}_{\gamma}\mathsf{R}(\mathtt{group})$ with respect to the aforementioned (sub-)list of generators.
- a list of all non-zero terms in $\mathsf{gr}^{\mathsf{n}}_{\gamma}\mathsf{R}(\mathtt{group})$ that vanish when mapped via restriction to the coproduct of all

  $\mathsf{gr}^{\mathsf{n}}_{\gamma}\mathsf{R}(\mathtt{Representative(ConjugacyClassesSubgroups(group)[i])}$

  where `i` is an entry in `subgroups`,
- a list of all isomorphism classes of groups determined by `subgroups` given as pairs of positive integers `[i,j]` that represent the corresponding entry in the SmallGroups library.

Be careful with the relations returned in the third entry of the list. This function returns only those relations witnessed in degree n which might be multiples of relations of lower degree. If you want a nice(r) description that fits into a presentation of $\mathsf{gr}^{*}_{\gamma}\mathsf{R}(\mathtt{group})$ you should use `presentation` instead.

Example 2.4. In the previous example, we have computed a generating set and all corresponding relations up to degree 3 for $\mathsf{gr}^{*}_{\gamma}\mathsf{R}(\mathsf{G})$ with

$$\mathsf{G} := \mathtt{gap.SmallGroup}(64, 138).$$

We would also like to know whether the map $\mathsf{gr}^{*}_{\gamma}\mathsf{R}(\mathsf{G}) \to \mathsf{gr}^{*}_{\mathrm{geom}}\mathsf{R}(\mathsf{G})$ is an isomorphism up to degree 3. This map is, independently of G, always an isomorphism in degrees 0 and 1. In fact, if we take Γ_{i} to be the i-th step in the γ-filtration and likewise $\mathsf{F}^{\mathsf{i}}_{\mathrm{geom}}$ the i-th step in the geometric filtration, it is known for all $\mathsf{i} \in \mathbb{N}$ that

$$\mathsf{F}^{\mathsf{i}}_{\mathrm{geom}} \subseteq \Gamma_{\mathsf{i}}$$

where this inclusion is equality for $\mathsf{i} = 0, 1, 2$. Therefore it suffices to check that $\mathsf{gr}^{*}_{\gamma}\mathsf{R}(\mathsf{G}) \to \mathsf{gr}^{*}_{\mathrm{geom}}\mathsf{R}(\mathsf{G})$ is injective up to degree 3, as this is equivalent to $\mathsf{F}^{\mathsf{i}}_{\mathrm{geom}} = \Gamma_{\mathsf{i}}$ for $\mathsf{i} = 3, 4$. In the previous example we have already determined that we may run our functions with modified_strong_gens=[true,[0,1,2,7,8,10,11,12]]

to remove redundant generators of $\mathrm{gr}^*_\gamma R(G)$. We execute the function

```
presentation_with_restriction(gap.SmallGroup(64,138), 2, 64,
        [true,[0,1,2,7,8,10,11,12]], [41,49,55])
```

which returns a list whose fourth entry is the empty list. This means

$$\ker\left(\mathrm{gr}^2_\gamma R(G) \to \coprod_{i\in[41,49,55]} \mathrm{gr}^2_\gamma R(H_i)\right) = 0$$

where we denote by H_i a representative of

gap.ConjugacyClassesSubgroups(SmallGroup(64,138))[i].

Checking the fifth entry of the returned list, the gap.IdGroup(H_i) for the given i are all isomorphic to gap.SmallGroup$(8,3)$ which is isomorphic to the dihedral group D_8 and $\mathrm{Syl}_2(\mathrm{GL}_3(\mathbb{F}_2))$. In particular their Chow rings are generated by Chern classes (see [Tot14, Chapter 13] or [Zie25]) which means $\mathrm{gr}^*_\gamma R(H_i) \cong \mathrm{gr}^*_{\mathrm{geom}} R(H_i)$. Applying the commutative square

$$\begin{tikzcd}
\mathrm{gr}^2_\gamma R(G) \arrow[r] \arrow[d, hook] & \mathrm{gr}^2_{\mathrm{geom}} R(G) \arrow[d] \\
\coprod_{i\in[41,49,55]} \mathrm{gr}^2_\gamma R(H_i) \arrow[r, "\sim"] & \coprod_{i\in[41,49,55]} \mathrm{gr}^2_{\mathrm{geom}} R(H_i)
\end{tikzcd}$$

we see that the top map injects and thus also surjects. The function

```
presentation_with_restriction(gap.SmallGroup(64,138), 3, 64,
        [true,[0,1,2,7,8,10,11,12]], [41,49,55])
```

again returns a list whose fourth entry is the empty list. By the same argument as in degree 2 we get $\mathrm{gr}^3_\gamma R(G) \cong \mathrm{gr}^3_{\mathrm{geom}}$.

In the sequel we will deal with modifications of the γ-filtration in order to determine the geomtric filtration. For this we need to describe elements of $R(G)$, where G is some finite group, in terms of abstract Chern classes, as this is the language the code operates in. Previously we have described the j-th Chern class of the i-th character in the character table of G as a nested list [1,[i,j]]. We want this to also be a stand-in for the canonical lift of the Chern class to $R(G)$ which is $C_j(\phi_i) := \lambda^j(\phi_i - \deg\phi_i + j - 1)$ where ϕ_i is the i-th character in the character table of G. Furthermore we will deal with integer polynomials of these (lifts of) Chern classes. We model an integer monomial

$$z\prod_{k=1}^{n} C_{j\_k}(\phi_{i\_k})$$

as a nested list [z,[i_1,j_1],...,[i_n,j_n]]. We model an integer polynomial of Chern classes just as a list of the monomials that are its summands.

Function 2.5. Take $A_i \subseteq R(G)$ to be a family of finite sets indexed over $i \in \mathbb{N}$ that is the empty set for all but finitely many i. Let F^i be the coarsest filtration on $R(G)$ such that $\Gamma_i \subseteq F^i$, $F^i \cdot F^j \subseteq F^{i+j}$ and $A_i \subseteq F^i$. Ideally $F^i = F^i_{\mathrm{geom}}$ where the latter denotes the i-th step in the geometric filtration on $R(G)$. One can compute the associated graded algebra to the F^i via

```
presentation_with_classes(group, n, upper_bound,
    modified_strong_gens, added_strong_gens, added_degrees)
```

which takes as parameters

- group - a finite group given as a [GAP4]-object,

- `n` - a positive integer that is the highest degree in which relations are computed,
- `upper_bound` - a positive integer that is an upper bound (with respect to divisibility) for the additive order of elements of F^i/F^{i+1} where $i \leqslant n$,
- `modified_strong_gens` - either `[false]` or `[true, x]` where `x` is an ascending list of integers that correspond to generators of $gr^*_\gamma R(\texttt{group})$,
- `added_strong_gens` - a list of elements of $R(\texttt{group})$ represented as polynomials of canonical lifts of Chern classes which should be understood as the elements of A_i above
- `added_degrees` - a list of positive integers whose j-th entry describes the degree of the j-th entry of `added_strong_gens`, or more precisely the largest index i such that j-th entry of `added_strong_gens` is an element of A_i.

The parameter `modified_strong_gens`, works as usual. To find suitable candidates for `upper_bound` one could use the fact, that elements in Γ_i/Γ_{i+1} are $\#G$-torsion which implies elements inΓ_i/Γ_{i+k} are $(\#G)^k$-torsion. We'll see an example later.

The function returns a list whose three entries are

- if `modified_strong_gens=[false]` a list of generators of the associated graded algebra of the F_i including a generator for each entry of the parameter `added_strong_gens`, if `modified_strong_gens=[true,x]` the sublist of entries at positions specified in `x` together with all generators corresponding to entries of `added_strong_gens`,
- a list of corresponding Chern classes given in the form `[1,[i,j]]` which represent the `j`-th Chern class of the `i`-th character in the character table of `group` or just

  `['generator that is not a Chern class', j]`

  for every generator of degree `j` that is not a polynomial of Chern classes,
- a generating set of the ideal containing all relations, up to degree `n`, of the presentation of the associated graded algebra for the F_i with respect to the aforementioned (sub-)list of generators.

Function 2.6. Take $A_i \subseteq R(G)$ to be a family of finite sets that is empty for all but finitely many i and F^i to be the coarsest filtration on $R(G)$ such that $\Gamma_i \subseteq F^i$, $F^i \cdot F^j \subseteq F^{i+j}$ and $A_i \subseteq F^i$. Ideally $F^i = F^i_{\mathrm{geom}}$. Furthermore let $H_j \leqslant G$ be a family of subgroups together with families of finite sets $A_i^{H_j} \subseteq R(H_j)$ giving rise to filtrations $F_i^{H_j}$ on $R(H_j)$ analogously. One can check how far F^i and F^i_{geom} might differ by invoking the function

```
presentation_with_restriction_and_classes(group, n, upper_bound,
modified_strong_gens, subgroups, added_strong_gens, added_degrees,
sub_added_rels, sub_added_degrees)
```

which takes as parameters

- `group` - a finite group given as a [GAP4]-object,
- `n` - a positive integer that is the degree in which relations are computed,
- `upper_bound` - a positive integer that is an upper bound (with respect to divisibility) for the additive order of elements of F^i/F^{i+1} where $i \leqslant n$,
- `modified_strong_gens` - either `[false]` or `[true, x]` where `x` is an ascending list of integers that correspond to generators of $gr^*_\gamma R(\texttt{group})$,
- `subgroups` - a list of integers that correspond to the positions of conjugacy classes of subgroups of `group` as provided by the [GAP4]-function `ConjugacyClassesSubgroups(group);` such that the intersection of the

kernels of restrictions to representatives of those conjugacy classes is computed.

- `added_strong_gens` - a list of elements of $R(\texttt{group})$ represented as polynomials of canonical lifts of Chern classes which should be understood as the elements of A_i above
- `added_degrees` - a list of positive integers whose j-th entry describes the degree of the j-th entry of `added_strong_gens`, or in the language above the largest i such that j-th entry of `added_strong_gens` is an element ofA_is,
- `sub_added_rels` - a list whose j-th entry is a list of elements of $R(H_j)$ represented as polynomials of lifted Chern classes for

$$H_j := \texttt{Representative}(\texttt{ConjugacyClassesSubgroups}(\texttt{group})[\texttt{subgroup}[\texttt{j}]])$$

  which should be understood as the elements of $A_i^{H_j}$ above (alternatively to a list of empty lists this can be chosen to be just `[]`, the empty list),
- `sub_added_degrees` - a list of lists of positive integers whose j-th entry describes the degrees of the j-th entry of `sub_added_rels` analogously to how `added_degrees` describes the degrees of entries of `added_strong_gens` (alternatively to a list of empty lists this can be chosen to be just `[]`, the empty list).

The parameter `modified_strong_gens`, works as usual. To find suitable candidates for `upper_bound` one could use the fact, that elements in Γ_i/Γ_{i+1} are $\#G$-torsion which implies elements in Γ_i/Γ_{i+k} are $(\#G)^k$-torsion. We'll see an example later.

The function returns a list whose five entries are

- if `modified_strong_gens=[false]` a list of generators of the associated graded algebra of the F_i including a generator for each entry of the parameter `added_strong_gens`, if `modified_strong_gens=[true,x]` the sublist of entries at positions specified in `x` together with all generators corresponding to entries of `added_strong_gens`,
- a list of corresponding Chern classes given in the form `[1,[i,j]]` which represent the `j`-th Chern class of the `i`-th character in the character table of `group` or just

```
['generator that is not a Chern class', j]
```

  for every generator of degree j that is not a polynomial of Chern classes,
- a generating set of the ideal containing all relations in degree `n` (sometimes also some redundant relations of higher degree) of the presentation of the associated graded algebra for the F_i with respect to the aforementioned (sub-)list of generators,
- a list of all non-zero terms in F_n/F_{n+1} that vanish when restricted to all $F_n^{H_j}/F_{n+1}^{H_j}$ where `j` is an entry in `subgroups`,
- a list of all isomorphism classes of groups determined by `subgroups` given as pairs of positive integers `[i,j]` that represent the corresponding entry in the SmallGroups library.

Be careful with the relations returned in the third entry of the list. This function returns only those relations witnessed in degree n and determined by the parameter `added_strong_gens`, which might be multiples of relations of lower degree. If you want a nice(r) description that fits into a presentation of $\bigoplus_i F_i/F_{i+1}$ you should use `presentation_with_classes` instead.

Example 2.7. `presentation(gap.SmallGroup(32,8),1,32,[false])` returns a list of generators

```
[x6, x4, x7, x9, x10, x12, x13, x14]
```

identified with a list of abstract Chern classes

```
[[1, [8, 1]], [1, [6, 1]], [1, [9, 1]], [1, [11, 1]], [1, [9, 2]],
    [1, [11, 2]], [1, [11, 3]], [1, [11, 4]]]
```

and returns an ideal of relations

```
(x9, 2*x7, 4*x6, x4 + x7).
```

We see that x9 and x7 are redundant and

$$\mathsf{gr}^1_\gamma \mathsf{R}(\mathsf{G}) \cong \mathsf{gr}^1_{\text{geom}} \mathsf{R}(\mathsf{G}) \cong \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/4\mathbb{Z}.$$

We execute

```
presentation(gap.SmallGroup(32,8), 2, 32, [true, [0, 1, 4, 5, 6,
    7]])
```

and get an ideal of relations

```
(4*x12, 4*x10, 4*x6, 2*x4, 2*x6^2, x4^2 + x4*x6 + x6^2 + 2*x10 +
    2*x12).
```

Thus $\mathsf{gr}^2_\gamma \cong (\mathbb{Z}/2\mathbb{Z})^2 \oplus (\mathbb{Z}/4\mathbb{Z})^2$. By invoking

```
presentation_with_restriction(gap.SmallGroup(32,8),2,32,
    [true,[0,1,4,5,6,7]],[9,10,12,13]])
```

we get a list of possible additional relations in $\mathsf{gr}^2_{\text{geom}} \mathsf{R}(\mathsf{G})$

```
[[2*x10, x4^2, x4^2 + 2*x10, x4*x6 + x6^2 + 2*x12, x4*x6 + x6^2 +
    2*x10 + 2*x12, x4^2 + x4*x6 + x6^2 + 2*x12]].
```

As all checked subgroups are isomorphic to $\mathbb{Z}/8\mathbb{Z}$ or $\mathbb{Z}/4\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z}$ whose Chow rings are generated by Chern classes, the γ- and geometric filtrations coincide. We know $\mathsf{gr}^1_{\text{geom}} \mathsf{R}(\mathsf{G}) \cong \mathsf{CH}^1(\mathsf{BG}) \cong \mathsf{H}^2(\mathsf{BG}, \mathbb{Z})$ and that

$$\mathsf{gr}^2_{\text{geom}} \mathsf{R}(\mathsf{G}) \cong \mathsf{CH}^2(\mathsf{BG}) \to \mathsf{H}^4(\mathsf{BG}, \mathbb{Z})$$

injects. Using [Ell24] we can compute all squares of 2-torsion elements of $\mathsf{H}^2(\mathsf{BG}, \mathbb{Z})$, only one of which vanishes. This element has to be the image of `2*x6^2`. We can conclude, that `x4^2` can't vanish. Suppose `x4^2+2*x10` does not vanish, then $\mathsf{gr}^2_{\text{geom}} \mathsf{R}(\mathsf{G})$ contains a subgroup isomorphic to $(\mathbb{Z}/2\mathbb{Z})^4$ while [Ell24] tells us that $\mathsf{H}^2(\mathsf{BG}, \mathbb{Z}) \cong \mathbb{Z}/2\mathbb{Z} \oplus (\mathbb{Z}/4\mathbb{Z})^2$ does not. Inspecting now all previously computed possible relations, the only ones that can actually vanish are `x4^2+2*x10` and `x4*x6 + x6^2 + 2*x12` which are equivalent and must vanish. The only conclusion is that the map $\mathsf{gr}^2_\gamma \mathsf{R}(\mathsf{G}) \to \mathsf{gr}^2_{\text{geom}} \mathsf{R}(\mathsf{G})$ has non-trivial kernel of cardinality 2 generated by `x4^2+2*x10`. From this we can deduce $\mathsf{gr}^2_{\text{geom}} \mathsf{R}(\mathsf{G}) \cong \mathbb{Z}/2\mathbb{Z} \oplus (\mathbb{Z}/4\mathbb{Z})^2$.

Furthermore we know $\mathsf{gr}^3_{\text{geom}} \mathsf{R}(\mathsf{G})$ contains a generator given by the canonical lift of `x4^2+2*x10` to $\mathsf{R}(\mathsf{G})$, which can not be expressed in terms of Chern classes. We know for all i that $\mathsf{gr}^{\mathsf{i}}_\gamma \mathsf{R}(\mathsf{G})$ is $\#\mathsf{G}$-torsion meaning `x4^2+2*x10` is $(\#\mathsf{G})^2$-torsion modulo Γ_4. To compute $\mathsf{F}^3_{\text{geom}}/(\Gamma_4 + \Gamma_1(\texttt{x4^2+2*x10}))$ we run

```
presentation_with_classes(gap.SmallGroup(32,8),3,32^2,
    [true,[0,1,4,5,6,7]],[[[1,[6,1],[6,1]],[2,[9,2]]]],[],[3])
```

and obtain a list of relations

```
(2*x15, x13, 4*x12, 4*x10, 4*x6, 2*x4, 2*x10*x12, 2*x6*x12, x4*x12,
    2*x10^2, 2*x6*x10, x4*x10, 2*x6^2, x4*x6 + x6^2 + 2*x12, x4^2 +
    2*x10, x6^2*x12 + 2*x12^2, x6^2*x10, x6^3)
```

where x15 corresponds to the new generator given by the lift of `x4^2+2*x10`. Notably x13 is redundant. To compute $\mathsf{F}^3_{\text{geom}}/\mathsf{F}^4_{\text{geom}}$ we execute

```
presentation_with_restriction_and_classes(gap.SmallGroup(32,8), 3,
    32^2, [true,[0,1,4,5,7]],[9,10,12,13],
    [[[1,[6,1],[6,1]],[2,[9,2]]]], [3], [], [])
```

and get as possible additional relations

```
[x6*x10, x4^2*x6 + x6*x10, x4*x6^2 + x6*x10, x4^2*x6 + x4*x6^2 +
x6*x10].
```

All of these relations are equivalent.

It is known that the kernel of the surjection $\mathsf{CHs}(\mathsf{BG}) \to \mathsf{gr}^3_{\text{geom}}\mathsf{R}(\mathsf{G})$ is the image of $\mathsf{H}^3(\mathsf{BG}, \mathbb{Z})$ under the motivic power operation $\beta\mathrm{Sq}^1$ (for details see [Tot14, Lemma 15.11]. We can compute in HAP $\mathsf{H}^3(\mathsf{BG}, \mathbb{Z}) \cong \mathbb{Z}/2\mathbb{Z}$ telling us that $\mathsf{CH}^3(\mathsf{BG}) \to \mathsf{gr}^3_{\text{geom}}\mathsf{R}(\mathsf{G})$ is either an isomorphism or an extension with $\mathbb{Z}/2\mathbb{Z}$. Starting from this, one could use the cycle class map and computations of cup products in HAP to determine whether `x6*x10` vanishes in $\mathsf{gr}^3_{\text{geom}}\mathsf{R}(\mathsf{G})$. With our available computational power we were sadly not able to compute the necessary terms in the required resolution.

References

[Che20] Béatrice I. Chetard. "Graded character rings of finite groups". In: *Journal of Algebra* 549 (2020), pp. 291–318. ISSN: 0021-8693. DOI: `https://doi.org/10.1016/j.jalgebra.2019.11.041`.

[Ell24] G. Ellis. *HAP, Homological Algebra Programming, Version 1.66.* `https://gap-packages.github.io/hap`. GAP package. Oct. 2024.

[GAP4] *GAP – Groups, Algorithms, and Programming, Version 4.13.1.* The GAP Group. 2024. URL: `%5Curl%7Bhttps://www.gap-system.org%7D`.

[SAGE] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 3.11.1).* `https://www.sagemath.org`. 2024.

[Tot14] Burt Totaro. *Group Cohomology and Algebraic Cycles.* Cambridge Tracts in Mathematics. Cambridge University Press, 2014.

[Zie25] Alexander Ziegler. "Computing $\mathsf{CH}^*(\mathsf{B}(\mathsf{Syl}_2(\mathsf{GL}_4(\mathsf{F}_2))))/2$". In: *unpublished* (2025).