ModIsomExt

ModIsomExt - An extension of ModIsom

Version 1.1.0

16 December 2021

Leo Margolis
Tobias Moede

Leo Margolis Email: `leo.margolis@vub.be`
Homepage: `http://homepages.vub.ac.be/~lmargoli/`

Tobias Moede Email: `t.moede@tu-braunschweig.de`
Homepage: `https://www.tu-braunschweig.de/iaa/personal/moede`

Copyright

Acknowledgements

We appreciate very much all past and future comments, suggestions and contributions to this package and its documentation provided by GAP users and developers.

Contents

Chapter 1

The modular isomorphism problem (MIP)

1.1 What is MIP?

The modular isomorphism problem (MIP) asks the following: Let p be a prime, k a field of characteristic p and let G and H be p-groups. Does $kG \cong kH$ imply $G \cong H$? To our best knowledge, the problem was first posed by R. Brauer [Bra63]. See [San85], [HS06], [EK11] for some history of the problem and an overview of most known results.

1.2 The ModIsom package

The package ModIsomExt is an extension of the ModIsom package [Eic19]. ModIsom was used to study the modular isomorphim problem for groups of order 2^8 and 3^6 [Eic08] and 2^9 [EK11]. For this purpose so-called canonical forms of the associated augmentation ideals (and their quotients) are computed, see [Eic08].

ModIsomExt is built on the same ideas, but uses more efficient methods. In particular ModIsomExt allows to compute quotients of augmentation ideals without first computing the full augmentation ideal, which is very time and memory consuming. ModIsomExt was used in [MM20] to verify that there are no counterexamples to MIP for groups of order 3^7. Moreover ModIsomExt allows an easier application of the methods directly to groups instead of the GroupId's, hence making it possible to work, in priniciple, with groups of any size.

1.3 The ModIsomExt info class

The package ModIsomExt defines an info class *`InfoModIsomExt`*. Currently, there are two info levels implemented:

`0` No additional information is printed.

`1` `MIPBinSplit` (1.5.2) and `MIPBinsByGT` (2.1.1) print additional information during the respective computations.

If one sets also *`InfoModIsom`* to be 1, then `MIPBinSplit` (1.5.2) prints further additional information.

1.4 Nilpotent tables

We recall the format of nilpotent structure constants tables or just nilpotent tables as used in the ModIsom package. Let A be a finite-dimensional nilpotent associative algebra over a field F. Furthermore let $(b_1,\ldots,b_d)$ be a weighted basis of A, i.e. a basis with associated weights $(w_1,\ldots,w_d)$ such that $A^j = \langle b_i \mid w_i \geq j\rangle$ and let $a_{i,j,k}$ be such that $b_i b_j = \sum\limits_k a_{i,j,k} b_k$. Then a nilpotent table for A is a record with the following entries.

`dim`

The dimension d of A.

`fld`

The field over which A is defined.

`rnk`

The rank of A, i.e. the dimension of A/A^2.

`wgs`

The weights $(w_1,\ldots,w_d)$.

`wds`

A list of length d with possible holes. If the i-th enty is bound, then it has the form $[k,l]$ and $w_i > 1$, $b_i = b_k b_l$ with $w_k = 1$ and $w_l = w_i - 1$ holds.

`tab`

A partial structure constants table for A. If $tab[i][j][k]$ is bound, then it equals $a_{i,j,k}$.

`com`

Optional. If bound, then it is a boolean indicating whether the algebra is assumed to be commutative.

Note that the ModIsom package provides several functions for working with and manipulating nilpotent tables; see below and the ModIsom documentation for more information.

1.4.1 MultByTable

▷ `MultByTable(T, v, w)` (operation)

Given a nilpotent table `T` and two coefficient vectors `v` and `w` representing elements of the algebra described by `T`, the function returns a coefficient vector for the product of these elements, again relative to the basis for the algebra given in `T`.

1.5 Quotients of augmentation ideals and splitting bins

The following functions provide the main functionality of the ModIsomExt package.

1.5.1 TableOfRadQuotient

▷ `TableOfRadQuotient(kG, n)` (operation)

Given a modular group algebra *kG* the function computes the class-*n* quotient of the augmentation ideal $I(kG)$, i.e. $I(kG)/I(kG)^{n+1}$. The output is a nilpotent table for this quotient. Note that in addition to the standard entries of a nilpotent table it contains further entries for computational reasons. This allows do determine the class-*n* quotient of the augmentation ideal without computing the full augmentation ideal using `NilpotentTableOfRad` as provided by the ModIsom package.

The components *dim*, *fld*, *rnk*, *tab*, *wgs*, *wds* remain unchanged from the ModIsom package. The additional components are *commwords*, *powwords* and *pre*. These new components contain additional information on precisely which basis of $I(kG)/I(kG)^{n+1}$ is used and what the result of multiplying basis elements is. We explain how users can understand how the basis looks and how they can multiply two elements in the algebra.

The dimension of $I(kG)/I(kG)^{n+1}$ is recorded in *T.dim*. The basis of $I(kG)/I(kG)^{n+1}$ is found as in the theory of Jennings, cf. [MM20]. The elements of G chosen to provide the basis of subsequent quotients of dimension subgroups are recorded in *T.pre.jen.pcgs*. Let us call these elements $g_1,\dots,g_m$. Note that $|G| = p^m$. If now l is an integer smaller than *T.dim+1*, then the l-th elements of the basis of $I(kG)/I(kG)^{n+1}$ is $(g_1-1)^{e_1}\cdot\ldots(g_m-1)^{e_m}$ where *[e_1,...,e_m] = T.pre.exps[l]*. The weight of this element is recorded in *T.wgs[l]* and also *T.pre.weights[l]*.

We consider the group $G = SmallGroup(3^7, 19)$. The following example shows that $I(kG)/I(kG)^9$ has dimension 135 and that the full augmentation ideal $I(kG)$ has dimension 2186.

Example

```
gap> G := SmallGroup(3^7, 19);;
gap> kG := GroupRing(GF(3), G);;
gap> T := TableOfRadQuotient(kG, 8);;
gap> T.dim;
135

gap> T := TableOfRadQuotient(kG, 38);;
gap> T.dim;
2186

gap> T := TableOfRadQuotient(kG, 39);;
gap> T.dim;
2186
```

We next consider an example how the basis used can be recognized.

Example

```
gap> G := DihedralGroup(8);;
gap> kG := GroupRing(GF(2), G);;
gap> T := TableOfRadQuotient(kG, 4);;
gap> T.dim;
7
gap> pcgs := T.pre.jen.pcgs;
Pcgs([ f1, f2, f3 ])
gap> List(pcgs, Order);
[ 2, 4, 2 ]
gap> pcgs[3] in Center(G);
true
```

```
gap> T.pre.exps{[1..7]};
[ [ 1, 0, 0 ], [ 0, 1, 0 ], [ 1, 1, 0 ], [ 0, 0, 1 ], [ 1, 0, 1 ], [ 0, 1, 1 ],
  [ 1, 1, 1 ]]
```

We conclude that $I(kG)/I(kG)^5$ is 7-dimensional and if we denote by a a reflection and by b a non-central rotation in G, then the basis used by T is: $(a-1)$, $(b-1)$, $(a-1)(b-1)$, (b^2-1), $(a-1)(b^2-1)$, $(b-1)(b^2-1)$, $(a-1)(b-1)(b^2-1)$.
Say continuing the previous example we want to multiply $(b-1)+(a-1)(b-1)+(a-1)(b^2-1)$ and $(a-1)+(b-1)+(b^2-1)$.

Example

```
gap> v := Z(2)^0*[0,1,1,0,1,0,0];
[ 0*Z(2), Z(2)^0, Z(2)^0, 0*Z(2), Z(2)^0, 0*Z(2), 0*Z(2) ]
gap> w := Z(2)^0*[1,1,0,1,0,0,0];
[ Z(2)^0, Z(2)^0, 0*Z(2), Z(2)^0, 0*Z(2), 0*Z(2), 0*Z(2) ]
gap> MultByTable(T,v,w);
[ 0*Z(2), 0*Z(2), Z(2)^0, 0*Z(2), Z(2)^0, 0*Z(2), 0*Z(2) ]
```

So the result is $(a-1)(b-1)+(a-1)(b^2-1)$.

1.5.2 MIPBinSplit

▷ `MIPBinSplit(p, n, max, start, step, L)` (operation)

Given a list `L` of small group library ids or a list of groups of order `p^n` this functions checks isomorphism of the associated modular group algebras using canonical forms for the quotients of the augmentation ideals $I(kG)$. The parameter `max` is an integer or false that determines the maximal quotients $I(kG)/I(kG)^{max}$ to be checked (if false is given as input, then the quotients are enlarged until non-isomorphic quotients are found or eventually the full augmentation ideal will be checked). The parameter `start` specifies which quotients $I(kG)/I(kG)^{start}$ are precomputed. The parameter `step` determines in which steps the quotients are enlarged if necessary during the isomorphism check.

The output is a record containing three entries: `bins` contains all the groups, for which the non-isomorphism of the associated modular group algebras could not be determined; `splits` contains all the groups, for which the associated group algebras were determined to be non-isomorphic (and the first non-isomorphic quotient); `time` contains the time used for the computation (in milliseconds).

The groups $G = SmallGroup(3^7, 19)$ and $H = SmallGroup(3^7, 43)$ are in the same bin after using the group theoretical invariants. The following example shows $I(kG)/I(kG)^6 \cong I(kH)/I(kH)^6$ and $I(kG)/I(kG)^9 \ncong I(kH)/I(kH)^9$.

Example

```
gap> MIPBinSplit(3, 7, 5, 10, 5, [19,43]);
rec( bins := [ [ 19, 43 ] ], splits := [  ], time := 1056 )

gap> MIPBinSplit(3, 7, false, 10, 5, [19,43]);
rec( bins := [  ], splits := [ [ 8, [ 19, 43 ] ] ], time := 78920 )
```

Chapter 2

Group theoretical invariants

We call a property of a p-group G a group-theoretical invariant, if $kG \cong kH$ implies that H has the same property. Here k denotes the field with p elements.

2.1 Computing Bins

The following function applies the group theoretical invariants included in [MM20] to split the given groups into so-called bins. Groups that are in different bins do not share a certain group-theoretical invariant. In particular, they do not provide a counterexample to the MIP. The function also checks if a group lies in a class of groups for which the MIP is known based on the list in [MM20]. In this case it does not appear in any bin.

2.1.1 MIPBinsByGT

▷ `MIPBinsByGT(p, n[, L])` (operation)

Given a list `L` of small group library ids or a list of groups of order `p^n` the function uses group theoretical invariants to split the groups into bins. If `L` is not given, then all groups of order p^n are considered.

2.2 Some group-theoretical invariants

We document some of the major group-theoretical invariants for the MIP which are not easily available as standard `GAP`-functions.

2.2.1 GroupInfo

▷ `GroupInfo(G)` (operation)

This is an auxiliary function used in other group-theoretical invariants. If `IdGroup` is available in `GAP` for the order of `G` it returns `IdGroup(G)`. Otherwise it returns `[Size(G), AbelianInvariants(G)]`. This function remains unchanged from ModIsom, but was not documented before.

2.2.2 MIPConjugacyClassInfo

▷ `MIPConjugacyClassInfo(G)` (operation)

For a given p-group `G` this function returns a list `L` containing known group-theoretical invariants associated to the conjugacy classes of `G`. The first entry of `L` is the so-called Roggenkamp parameter $\sum_{g^G} \log_p(|C_G(g)/\Phi(C_G(g))|)$ where the sum runs over conjugacy classes of `G`. The next entries contain the number of conjugacy classe which are p^ℓ-th powers, for $0 \leq \ell \leq log_p(exp(G))$ (Kuelshammer). Note that for $\ell = 0$ this is just the number of conjugacy classes in `G`. Finally, the following entries contain the number of conjugacy classes of p^ℓ-th powers which are not central and have the same order as a class which powers to them where $0 \leq \ell \leq log_p(exp(G)) - 1$ (Parmenter-Polcino Milies).

2.2.3 SubgroupsInfo

▷ `SubgroupsInfo(G)` (operation)

For a given p-group `G` this function returns a list `L` containing at the `i`-th position the number of conjugacy classes of maximal elementary-abelian subgroups of order p^i in `G` (Quillen). This function remains unchanged from ModIsom, but was not documented before.

2.2.4 MIPJenningsInfo

▷ `MIPJenningsInfo(G)` (operation)

For a given `p`-group `G` this function returns a list `L` containing information on quotients of the Jennings-Zassenhaus series $D_i(G)$ of `G`. Starting with *i=1* for increasing *i* it contatins the *GroupInfo* for $D_i(G)/D_{i+1}(G)$, $D_i(G)/D_{i+2}(G)$, and $D_i(G)/D_{2i+1}(G)$ when these are defined. The last entry describes $G/D_3(G)$ if *p=2* and $G/D_4(G)$ if *p>2*. If $D_3(G) = 1$ or $D_4(G) = 1$, respectively, and *IdGroup* is a known attribute of `G`, it is *IdGroup(G)*. Otherwise it contains the *GroupInfo* of $G/D_3(G) = 1$ or $G/D_4(G) = 1$ respectively.

2.2.5 MIPSandlingInfo

▷ `MIPSandlingInfo(G)` (operation)

For a given `p`-group `G` this function returns a list `L` containing information on the Sandling quotient $G/\gamma_2(G)^p\gamma_3(G)$. The first entry describes $Q = G/\gamma_2(G)^p\gamma_3(G)$ in the following way: If $\gamma_2(G)^p\gamma_3(G) = 1$ and *IdGroup* is a known attribute of `G`, it is *IdGroup(G)*. Otherwise it contains the *GroupInfo* of $G/\gamma_2(G)^p\gamma_3(G)$ (Sandling). Moreover, if `G` is generated by at most two elements and the length of the Jennings-Zasenhaus series of `G` is at least four, it contains a second entry describing $G/\gamma_2(G)^p\gamma_4(G)$ in a similar way (Baginski/Margolis-Moede).

Chapter 3

Jennings bound

3.1 Jennings bound

For a pair of groups G and H the Jennings bound is defined as the maximal integer s such that $G/D_s(G) \cong H/D_s(H)$, where D_i is the Jennings-Zassenhaus series (see [MM20]). If s is the Jennings bound for G and H, then it follows that $I(kG)/I(kG)^s \cong I(kH)/I(kH)^s$. Thus s is a minimum for the level until which `MIPBinSplit` needs to run to be able to split the groups.

3.1.1 JenningsBound

▷ `JenningsBound(p, n, L)` (operation)

Given a list `L` of small group library ids or a list of groups of order `p^n` the function computes an integer b such that the quotients of the associated augmentation ideals are guaranteed to be isomorphic up to class $b-1$.

More precisely b is the biggest integer such that for any $G,H \in L$ one has $G/D_b(G) \cong H/D_b(H)$.

3.1.2 JenningsBoundPairwise

▷ `JenningsBoundPairwise(p, n, L)` (operation)

Given a list `L` of small group library ids or a list of groups of order `p^n` the function computes for all pairs (G,H) of groups in the list an integer b such that the quotients of the associated augmentation ideals are guaranteed to be isomorphic up to class $b-1$.

More precisely the return is a list of triples. The first two entries of each triple are two groups G and H, or their ids if they are available, and the last entry contains *`JenningsBound(p,n,[G,H])`*.

3.1.3 JenningsBoundConjecture

▷ `JenningsBoundConjecture(p, n, L)` (operation)

Given a list `L` in the format returned by `MIPBinSplit` for some groups of order p^n this function checks if the groups violate the bound conjectured in Question 2.7 [MM20] on the maximal quotients of the corresponding augmentation ideals which need to be checked to decide MIP.

For elements of `MIPResults(p,n)` which are solved by theoretical results, or which remain open, it returns `fail`.

3.1.4 JenningsBoundConjectureIsStrict

▷ `JenningsBoundConjectureIsStrict(p, n, L)` (operation)

Given a list `L` in the format returned by `MIPBinSplit` for some groups of order p^n this function checks if the groups attain the bound conjectured in Question 2.7 [MM20] on the maximal quotients of the corresponding augmentation ideals which need to be checked to decide MIP. This function currently only works if `L` contains the information for a pair of groups.

Chapter 4

Computational results

The ModIsomExt package contains precomputed bins and results for certain orders of groups. These can be accessed using the following functions.

4.1 Bins

4.1.1 MIPBins

▷ `MIPBins(p, n)` (operation)

Given a prime `p` and a positive integer `n` the function returns the bins resulting from using group theoretical invariants. Precomputed bins are only available for orders 2^6, 2^7, 2^8, 2^9, 3^6, 3^7 and 5^6. Note that in the case 2^9 only 2- and 3-generated groups of this order are considered.

4.2 Results

4.2.1 MIPResults

▷ `MIPResults(p, n)` (operation)

Given a prime `p` and a positive integer `n` the function returns the results of verifying MIP. This is only available for orders 2^6, 2^7, 2^8, 2^9, 3^6, 3^7 and 5^6. Note that in the case 2^9 only 2- and 3-generated groups of this order are considered.

References

[Bra63] R. Brauer. Representations of finite groups. In *Lectures on Modern Mathematics, Vol. I*, pages 133–175. Wiley, New York, 1963. 4

[Eic08] Bettina Eick. Computing automorphism groups and testing isomorphisms for modular group algebras. *J. Algebra*, 320(11):3895–3910, 2008. 4

[Eic19] B. Eick. Modisom 2.3.3., a gap package for computing isomorphisms of modular group algebras. *accepted*, 2019. 4

[EK11] B. Eick and A. Konovalov. The modular isomorphism problem for the groups of order 512. In *Groups St Andrews 2009 in Bath. Volume 2*, volume 388 of *London Math. Soc. Lecture Note Ser.*, pages 375–383. Cambridge Univ. Press, Cambridge, 2011. 4

[HS06] M. Hertweck and M. Soriano. On the modular isomorphism problem: groups of order 2^6. In *Groups, rings and algebras*, volume 420 of *Contemp. Math.*, pages 177–213. Amer. Math. Soc., Providence, RI, 2006. 4

[MM20] Leo Margolis and Tobias Moede. Known invariants of the modular group algebra of a p-group. *in preparation*, 2020. 4, 6, 8, 10, 11

[San85] R. Sandling. The isomorphism problem for group rings: a survey. In *Orders and their applications (Oberwolfach, 1984)*, volume 1142 of *Lecture Notes in Math.*, pages 256–288. Springer, Berlin, 1985. 4

Index