ModIsom

A GAP4 Package

Version 1.0

by

Bettina Eick

Institut Computational Mathematics, TU Braunschweig

Pockelsstrasse 14, 38106 Braunschweig, Germany

email: beick@tu-bs.de

April 2009

Contents

1 Introduction

The modular isomorphism problem asks whether $\mathbb{F}G \cong \mathbb{F}H$ implies that $G \cong H$ for two p-groups G and H and $\mathbb{F}$ the field with p elements. This problem is still open, despite various efforts towards proving the claim or finding counterexamples to it. The claim has been proved, for example, for abelian p-groups [Des56], p-groups of class 2 and exponent p [PS72], metacyclic p-groups [San96] and groups of order p^n dividing 2^7 [Wur93, BKRW99] or p^5 [SS96].

The modular isomorphism problem can be considered as a special case of a wider range of problems. In general, one can ask under which conditions on a ring R and groups G and H does $RG \cong RH$ imply $G \cong H$. Important results on this type of problem are due to Dade [Dad71] who found that for every field R there exist finite groups $G \ncong H$ with $RG \cong RH$ and to Hertweck [Her01] who determined finite groups $G \ncong H$ with $\mathbb{Z}G \cong \mathbb{Z}H$. In both cases, explicit examples for G and H are known, but none of them are p-groups. On the other hand, Roggenkamp and Scott [RS87] proved that $RG \cong RH$ implies that $G \cong H$ if R is the ring of p-adic integers and G and H are p-groups.

Computational approaches have been used to investigate the modular isomorphism problem. Based on an algorithm by Roggenkamp and Scott [RS93], Wursthorn [Wur93] described an algorithm for checking the modular isomorphism problem; that is, he described an algorithm for checking whether two modular group algebras $\mathbb{F}G$ and $\mathbb{F}H$ are isomorphic. This algorithm has been implemented in C by Wursthorn and has been used applied to the groups of order dividing 2^7 without finding a counterexample, see [BKRW99].

This package contains an implementation of the new algorithm described in [Eic08] for checking isomorphism of modular group algebras. Hence it provides a new method to investigate the modular isomorphism problem. The algorithm underlying this method can also be used to check isomorphisms and to compute the automorphism group of a finite dimensional, nilpotent, associative algebra over a finite field.

1.1 Associative algebras and nilpotency

Let A be an associative algebra of dimension d over a field F of prime order and let $\{b_1, \ldots, b_d\}$ be a basis for A. We identify the element $x_1 b_1 + \ldots + x_d b_d$ of A with the element $(x_1, \ldots, x_d)$ of F^d and thus obtain an identification of A with the vector space F^d. The multiplication of A can then be described by a **structure constants table**: a 3-dimensional array with entries $a_{i,j,k} \in F$ satisfying that

$$b_i b_j = \sum_{k=1}^{d} a_{i,j,k} b_k.$$

A **canonical form** $Can(A)$ for A is a structure constants table for A which is unique for the isomorphism type of A; that is, two algebras A and B are isomorphic if and only if $Can(A) = Can(B)$ holds. Further, using the above identification of A with F^d the automorpism group of A can be defined as

$$Aut(A) = \{M \in GL(d, F) \mid (ab)M = (aM)(bM) \text{ for all } a, b \in A\}.$$

An associative algebra A is **nilpotent** if its **power series** terminates at the trivial ideal of A; that is, the algebra A has the series

$$A > A^2 > \ldots > A^n > A^{n+1} = \{0\}$$

where A^j is the ideal of A generated by all products of length i in A. Note that A is generated by $dim(A/A^2)$ elements in this case and A does not contain a multiplicative identity.

1.2 Modular group algebras

Let G be a finite p-group and F the field with p elements. Then FG is the **modular group algebra** defined by G; it is an associative algebra, contains a multiplicative identity and has dimenion $|G|$.

Every modular group algebra FG contains a unique maximal ideal $J(FG)$; this ideal has codimension 1 in FG and a basis is given by $\{g - 1 \mid g \in G, g \neq 1\}$. Thus $J(FG)$ coincides with the augementation ideal of FG. It is well-known that $J(FG)$ is nilpotent and a basis through its power series can be read off from the Jennings series of G.

By construction, the ideal $J(FG)$ is fully invariant under any isomorphism of FG. Hence we find that $Aut(FG)$ can be identified with $Aut(J(FG))$ and a canonical form $Can(J(FG))$ extends naturally to a canonical form of FG. In particular, two group algebras FG and FH are isomorphism if and only if $J(FG)$ and $J(FH)$ are isomorphic or, equivalently, if and only if $Can(J(FG)) = Can(J(FH))$ holds.

The main function of this package determines the automorphism group of $J(FG)$ and a canonical form $Can(J(FG))$.

1.3 The modular isomorphism problem

The modular isomorphism problem asks whether there exist two non-isomorphic finite p-groups G and H with $FG \cong FH$. The methods of this package can be used to check this problem for groups of small order. In fact, the implementation of this package has been used to check MIP for the groups of order dividing 2^8 and 3^6 (without finding a counterexample).

2 Methods and functions

This chapter contains all the main methods and functions of this package. Throughout, let F be a prime field. First, we note that there is an info class

1 ► `InfoModIsom`

which takes values 0 or 1. With value 1 it prints various informations about the computation of canonical forms and automorphism groups.

2.1 Structure constants tables

All algebras in this package are described by structure constants tables. In the case of this package, these are 3-dimensional arrays and thus elements of $F^{n\times n\times n}$. The functions converting between such tables and algebras are the following.

1 ► `AlgebraByTable(T)` M

returns the algebra defined by the table T.

2 ► `TableByBasis(A, b)` M

returns the table of A with respect to the basis b.

2.2 Associative algebras and nilpotency

Let A be an associative algebra over a prime field defined by a structure constants table in GAP.

1 ► `IsNilpotentAlgebra(A)` P

returns true if the algebra A is nilpotent and false otherwise.

2 ► `PowerSeries(A)` A

returns the power series of A.

3 ► `PowerBasis(A)` A

returns a basis of A exhibiting the power series provided that A is nilpotent.

4 ► `TableByPowerBasis(A)` A

returns a structure constants table for A with respect to the basis computed by PowerBasis.

5 ► `CanonicalForm(A)` A

returns a canonical form for A. This is a table T which is unique for the isomorphism type of A.

6 ► `AutomorphismGroup(A)` A

returns the automorphism group of A as a subgroup of $GL(dim(A), F)$.

2.3 Modular group algebras

The modular group algebra FG of a finite p-group G is not nilpotent, as it contains a unit. However, the augementation ideal A of FG is nilpotent and this is used in all computations with FG in this package.

1 ▶ `ModularGroupAlgebra(G)` A

returns the modular group algebra FG for G.

2 ▶ `AugmentationIdeal(FG)` A

returns the augmentation ideal of the group algebra FG.

3 ▶ `TableByPowerSeriesOfAug(FG)` A

returns a structure constants table for the augmentation ideal of FG with respect to a power basis of this augmentation ideal. Note that this method is usually significantly faster than `TableByPowerSeries(I)` where I is the augmentation ideal of FG, as the power series of the augmentation ideal of a modular group algebra can be determined from the Jennings series of G.

4 ▶ `CanoFormOfAugIdeal(FG)` F

returns the canonical form of the augmentation ideal of FG using the `TableByPowerSeriesOfAug(FG)`.

5 ▶ `AutomorphismGroupOfAugIdeal(FG)` F

returns the automorphism group of the augmentation ideal of FG using the `TableByPowerSeriesOfAug(FG)`.

6 ▶ `CanonicalForm(FG)` A

returns a canonical form for FG. This is a table T which is unique for the isomorphism type of FG as algebra over F. This function is based on `CanoFormOfAugIdeal(FG)`.

7 ▶ `AutomorphismGroup(FG)` A

returns the automorphism group of FG as a subgroup of the general linear group of dimension $dim(FG)$ over F. This function is based on `AutomorphismGroupOfAugIdeal(FG)`.

2.4 The modular isomorphism problem for groups of small order

A major application of the methods in this package has been the checking of the modular isomorphism problems for the groups of order dividing 2^8 and 3^6. This section contains the functions used for this purpose.

1 ▶ `BinsByGT( p, n )` F

returns a partion of the list $[1 \cdots NumberSmallGroups(p^n)]$ into sublists so that the modular group algebras of two groups SmallGroup(p^n, i) and SmallGroup(p^n, j) can not be isomorphic if i and j are in different lists. The function BinsByGT uses various group theoretic invariants to split the groups of order p^n in bins.

2 ▶ `CheckBin( p, n, k, bin )` F

For $i \in bin$ let G_i denote SmallGroup(p^n, i) and let A_i be the augementation ideal of FG_i. This function computes and compares the canonical forms of the algebras A_i/A_i^j for $i \in bin$ and increasing $j \in \{1, \ldots, k+1\}$.

At each level j it splits the current bins into sub-bins according to the different canonical forms of A_i/A_i^j. Bins of length 1 are then discarded.

The function returns if no further bins are available or if $j = k+1$ is reached. In the later case the function returns the remaining bins.

2.5 Examples

We compute the automorphism group and a canonical form for the modular group algebra of the dihedral group of order 8 with three different methods.

Method 1 and 2 just consider the augmentation ideal of the group algebra, with method 2 being faster for this purpose, since it uses the Jennings series of the considered group to determine a table through the power series of the augementation ideal.

Method 3 considers the full group algebra. The underlying functions are based on Method 2.

```
# take the dihedral group of order 8 and its modular group algebra
gap> G := SmallGroup(8, 3);
<pc group of size 8 with 3 generators>
gap> A := ModularGroupAlgebra(G);
<algebra-with-one over GF(2), with 3 generators>

# compute the automorphism group and a canonical form of its
# augmentation ideal -- Method 1
gap> I := AugmentationIdeal(A);
<two-sided ideal in <algebra-with-one of dimension 8 over GF(2)>,
  (3 generators)>
gap>
gap> CanonicalForm(I);
[ <an immutable 7x7 matrix over GF2>, <an immutable 7x7 matrix over GF2>,
  <an immutable 7x7 matrix over GF2>, <an immutable 7x7 matrix over GF2>,
  <an immutable 7x7 matrix over GF2>, <an immutable 7x7 matrix over GF2>,
  <an immutable 7x7 matrix over GF2> ]
gap> AutomorphismGroup(I);
<matrix group of size 512 with 9 generators>

# compute the automorphism group and a canonical form of its
# augmentation ideal -- Method 2
gap> CanoFormOfAugIdeal(A);
[ <a 7x7 matrix over GF2>, <a 7x7 matrix over GF2>, <a 7x7 matrix over GF2>,
  <a 7x7 matrix over GF2>, <a 7x7 matrix over GF2>, <a 7x7 matrix over GF2>,
  <a 7x7 matrix over GF2> ]
gap> AutomorphismGroupOfAugIdeal(A);
<matrix group of size 512 with 9 generators>

# compute the automorphism group and a canonical form of the
# full group algebra -- Method 3
gap> CanonicalForm(A);
[ <an immutable 8x8 matrix over GF2>, <an immutable 8x8 matrix over GF2>,
  <an immutable 8x8 matrix over GF2>, <an immutable 8x8 matrix over GF2>,
  <an immutable 8x8 matrix over GF2>, <an immutable 8x8 matrix over GF2>,
  <an immutable 8x8 matrix over GF2>, <an immutable 8x8 matrix over GF2> ]
gap> AutomorphismGroup(A);
<matrix group of size 512 with 8 generators>
```

We show how to check the modular isomorphism problem for the groups of order 64. We first use BinsByGT to determine bins and we then check the first of the resulting bins with CheckBin. The fact that CheckBin ends with an empty list of bins shows that all groups are splitted.

```
gap> bins := BinsByGT(2,6);
refine by abelian invariants of group (Sehgal/Ward)
13 bins with 256 groups
refine by abelian invariants of center (Sehgal/Ward)
30 bins with 237 groups
refine by lower central series (Sandling)
32 bins with 127 groups
refine by jennings series (Passi+Sehgal/Ritter+Sehgal)
36 bins with 123 groups
refine by conjugacy classes (Roggenkamp/Wursthorn)
16 bins with 36 groups
refine by elem-ab subgroups (Quillen)
  start bin 1 of 16
  start bin 2 of 16
  start bin 3 of 16
  start bin 4 of 16
  start bin 5 of 16
  start bin 6 of 16
  start bin 7 of 16
  start bin 8 of 16
  start bin 9 of 16
  start bin 10 of 16
  start bin 11 of 16
  start bin 12 of 16
  start bin 13 of 16
  start bin 14 of 16
  start bin 15 of 16
  start bin 16 of 16
9 bins with 21 groups
[ [ 13, 14 ], [ 18, 19 ], [ 20, 22 ], [ 97, 101 ], [ 108, 110 ],
  [ 155, 157, 159 ], [ 156, 158, 160 ], [ 173, 176 ], [ 179, 180, 181 ] ]

gap> CheckBin(2,6,bins[1]);
compute tables through power series
  determined table for 1
  determined table for 2

refine bin
  weights yields bins [ [ 1, 2 ] ]
  layer 1 yields bins [ [ 1, 2 ] ]
  layer 2 yields bins [ [ 1, 2 ] ]
  layer 3 yields bins [ [ 1, 2 ] ]
  layer 4 yields bins [  ]
```

Bibliography

[BKRW99] Frauke M. Bleher, Wolfgang Kimmerle, Klaus W. Roggenkamp, and Martin Wursthorn. Computational aspects of the isomorphism problem. In *Algorithmic algebra and number theory (Heidelberg, 1997)*, pages 313–329. Springer, Berlin, 1999.

[Des56] W. E. Deskins. Finite Abelian groups with isomorphic group algebras. *Duke Math. J.*, 23:35–40, 1956.

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

[Her01] Martin Hertweck. A solution to the integral isomorphism problem. *Ann. of Math.*, 154:115 – 138, 2001.

[PS72] Inder Bir S. Passi and Sudarshan K. Sehgal. Isomorphism of modular group algebras. *Math. Z.*, 129:65–73, 1972.

[RS87] Klaus Roggenkamp and Leonard Scott. Isomorphisms of p-adic group rings. *Ann. of Math. (2)*, 126(3):593–647, 1987.

[RS93] K. W. Roggenkamp and L. L. Scott. Automorphisms and nonabelian cohomology: an algorithm. *Linear Algebra Appl.*, 192:355–382, 1993.

[San96] Robert Sandling. The modular group algebra problem for metacyclic p-groups. *Proc. Amer. Math. Soc.*, 124(5):1347–1350, 1996.

[SS96] Mohamed A. M. Salim and Robert Sandling. The modular group algebra problem for groups of order p^5. *J. Austral. Math. Soc. Ser. A*, 61(2):229–237, 1996.

[Wur93] Martin Wursthorn. Isomorphisms of modular group algebras: an algorithm and its application to groups of order 2^6. *J. Symbolic Comput.*, 15(2):211–227, 1993.

Index

This index covers only this manual. A page number in *italics* refers to a whole section which is devoted to the indexed subject. Keywords are sorted with case and spaces ignored, e.g., "`PermutationCharacter`" comes before "permutation group".