The Cohomology of finite p–Groups

Simon King (joint work with David J. Green)

DFG project GR 1585/4–1
Friedrich–Schiller–Universität Jena

January 22, 2009

Outline

What is a Cohomology ring?

Let G be a finite p-group (i.e., p prime, $|G| = p^n$). We study the modular cohomology ring $H^*(G;\mathbb{F}_p)$.

- $H^*(G;\mathbb{F}_p)$ is a graded commutative finitely presented $\mathbb{F}_p$–algebra ($x \cdot y = (-1)^{\deg(x)\cdot\deg(y)} y \cdot x$).
- $H^*(G;\mathbb{F}_p)$ is determined by G up to isomorphism.
- Any group homomomorphism $\varphi\colon G_1 \longrightarrow G_2$ gives rise to an algebra homomorphism $\varphi^*\colon H^*(G_2;\mathbb{F}_p) \to H^*(G_1;\mathbb{F}_p)$
- Any subgroup $U \leq G$ gives rise to a restriction map $H^*(G;\mathbb{F}_p) \to H^*(U;\mathbb{F}_p)$

J. F. Carlson, 1997-2001(?)

Groups of order 64, long computation time, using MAGMA

DFG Project "Computational Group Cohomology"

Aim

- Compute the cohomology for many groups, including all 2328 groups of order 128 (provide minimal generating sets and relations)
- Test the Strong Benson Conjecture.
- Collect results in a data base.
- Present the results on Web pages

Web: `http://users.minet.uni-jena.de/~king/cohomology/`

Performance of our software

2-groups:

- For all 267 groups of order 64:
  $\sim$ 27 CPU-min, $\sim$ 38 clock-min (Intel Pentium M, 1.73 GHz)
- Very roughly 10 months for the 2328 groups of order 128:
  - 21 days for all but four groups of order 128 (parallel on two Dual Core AMD Opteron Processor 270 with 2 GHz, 16 Gb)
  - 2 months for the four exceptional cases (parallely)

We also have all but 8 cohomology rings for 3-, 5-, and 7-groups of order at most 625,
(known) Sylow–2 of the Higman-Sims group (order 512),
(new) Sylow–2 of the third Conway group (order 1024)

Approaches for computing Cohomology

Spectral Sequences

- Lyndon–Hochschild–Serre
  $\rightsquigarrow$ extraspecial 2–groups [D. Quillen, 1971]
- Eilenberg–Moore $\rightsquigarrow$ groups of order 32 [D. J. Rusin, 1989]

Projective resolutions

Yields approximation in increasing degree

Main problem: When is the computation finished?

- Carlson's Completeness Criterion (depends on a conjecture)
- Use spectral sequences ($\rightsquigarrow$ HAP, G. Ellis, P. Smith 2008)
- Benson's Completeness Criterion (see below)

Constructing minimal projective resolutions

D. Green 2001

Use n. c. Gröbner basis techniques for modules over $\mathbb{F}_p G$

- *Negative* monomial orders (for minimality)
- *Two-speed* replacement rules: *Type I* precedes *Type II*

$\mathbb{F}_3 C_3 \cong \mathbb{F}_3[t]/\langle t^3\rangle$, $M = (F_3 C_3 \cdot a \otimes F_3 C_3 \cdot b)/\langle t \cdot a - t^2 \cdot a + t \cdot b\rangle$

Type I rule: $t^3 \rightsquigarrow 0$. Type II rule: $t \cdot a \rightsquigarrow t^2 \cdot a - t \cdot b$.
Reduce $t \cdot a + t \cdot b$: $\rightsquigarrow t^2 \cdot a - t \cdot b + t \cdot b = t^2 \cdot a$
$\rightsquigarrow t^3 \cdot a - t^2 \cdot b \rightsquigarrow -t^2 \cdot b$ (Type I precedes Type II!).

Existence and uniqueness of reductions, Gröbner bases, computing kernels of homomorphisms...

Finding relations

Assume we know the cohomology out to degree n; hence:

- R_n: Free graded–commutative algebra over $\mathbb{F}_p$, given by minimal generators of $H^*(G;\mathbb{F}_p)$ of degree $\leq n$.
- $I_n \subset R_n$, generated by degree–$\leq n$–part of $\ker(R_n \to H^*(G))$.

Compute the next degree as follows:

- Standard monomials of I_n of degree $n+1$ $\rightsquigarrow$ decomposable $(n+1)$–classes in cohomology.
- Find new relations in degree $n+1$ $\rightsquigarrow$ I_{n+1}.
- Indecomposable classes $\rightsquigarrow$ new generators $\rightsquigarrow$ R_{n+1}

Special choice of generators

1. Nilpotent generators $\iff$ Restrictions to all maximal elem. ab. subgroups of G are nilpotent (easy to test!)
2. "Boring" generators: Not nilpotent, but restriction to the greatest central elem. ab. subgroup is nilpotent.
3. Remaining: *Duflot regular* generators

Reason for that choice of generators

David Green's monomial order on R_n relies on the generator types.
It simplifies the computations by magic!
Also, it is used in the completeness criterion below.

Benson's Completeness Criterion

G abelian $\Rightarrow$ degree 2 suffices. Otherwise:

- Let r be the p-Rank of G and let R_n/I_n approximate $H^*(G)$
- Let $P_1,\dots,P_r \in R_n/I_n$ be a **filter-regular** HSOP, $\deg(P_i) \geq 2$
  I.e., the multiplication by P_i on $R_n/\left(I_n + \langle P_1,\dots,P_{i-1}\rangle\right)$
  has finite kernel, for $i = 1,\dots,r$.
- Maximal degrees of the kernels and of $R_n/\left(I_n + \langle P_1,\dots,P_r\rangle\right)$
  $\rightsquigarrow$ "filter degree type" $(d_1,\dots,d_{r+1})$ (after easy computation)

Theorem [D. J. Benson, 2004]

$n > \max\left(0, d_i + i - 1\right)_{i=1,\dots,r} + \sum_i \deg(P_i) - r \quad \Rightarrow \quad \boxed{R_n/I_n \cong H^*(G)}$

Remark "$n \geq \dots$" suffices if $\mathrm{rk}(Z(G)) \geq 2$

Conj. If $R_n/I_n \cong H^*(G)$ then $(d_1,\dots,d_{r+1}) = (-1,-2,\dots,-r,-r)$

Constructing a filter-regular HSOP

- Let $r = p\text{-rk}(G)$ and $z = \text{rk}(Z(G))$.
  Find **Duflot regular** generators $g_1, \ldots, g_z \in H^*(G)$.
- Let $U_1, \ldots, U_m \subset G$ be the maximal elementary abelian subgroups.
  Using **Dickson invariants**: Compute classes $D_{i,j}$ in the polynomial part of $H^*(U_j)$, for $i = 1, \ldots, r - z$ and $j = 1, \ldots, m$.
- There are classes $\Delta_i \in H^*(G)$ for $i = 1, \ldots, r - z$, simultaneously restricting to the p^{k_i}-th power of $D_{i,j}$ for $j = 1, \ldots, m$.
  Very often, $k_i = 0$.
- $g_1, \ldots, g_z, \Delta_1, \ldots, \Delta_{r-z}$ is a filter-regular HSOP of $H^*(G)$ [D. J. Benson]. Computable in R_n/I_n !!

Improvement by existence proof

Problem: Dickson invariants may be of large degree

- We take minimal factors of the Δ_i, for decreasing the degrees.
- We use the Dickson classes *only* for computing the filter degree type (which is the same for any f. r. HSOP)!
- We prove the presence of a small–degree filter–regular HSOP

non-constructively

In R_n/I_n, mod out $g_1, \ldots, g_z$, possibly some $\Delta_1, \ldots, \Delta_{i_0}$, and all monomials of some degree d. If the quotient is finite, then there exist parameters in degree d that extend $g_1, \ldots, g_z, \Delta_1, \ldots, \Delta_{i_0}$ to a filter-regular HSOP [D. Green, S. K. 2008].

Conway(3) and others would be unfeasible without that trick!

Important SAGE features

GAP, SMALLGROUPS library

GAP functions and C-executables of David Green yield data on the group and its elementary abelian subgroups.

CYTHON

- Wrapper MTX for C-MEATAXE matrices (see below)
- Compute resolutions (wrapping C-programs of David Green)
- Provide new extension classes for Cochains, Chain Maps, Cohomology Rings
  and methods for cup product, restriction, degree–wise approximation, Benson's test, creating Web pages, etc.

Cython yields very good speed!

Important SAGE features

SINGULAR

Used for all graded commutative stuff. It rocks!

- Gröbner basis of the relation ideal of group 836 of order 128:
  - > 1 month with self-made implementation (D. Green), but
  - only few hours with SINGULAR.
- Lift Dickson classes: Either by linear algebra (MTX), or by elimination (SINGULAR).
- Detection of filter regular HSOPs.

YOU

Your comments led to huge speed-ups by making better use of SINGULAR interface and CYTHON — Thank you!

Why C-MEATAXE?

MTX matrices

Purpose: Linear Algebra over fields of order < 256
Wraps a modified version of C-MEATAXE 2.2.3.

Reasons:

- David Green's programs for computing resolutions rely on C-MEATAXE 2.2.3. Re-implementation or conversion sucks.
- We need the following operations to be fast:
  - Copying, pickling, hash, equality test, element access, conversion into lists
  - Sum, difference, skalar multiplication
  - Nullspace
- Almost no need for matrix inversion or multiplication.

MTX vs. usual SAGE-3.2.3 matrices

On AMD Athlon 64 Processor 3700+ with 2.2 GHz, 1 GB RAM, and 2 GB Swap

$\mathbb{F}_7$, random 500×500

Hash, sum and difference was slow in SAGE, is now ok.

	SAGE	MTX	
copy(M)	$1.29ms$	$0.27ms$	
loads(dumps(M))	$199ms$	$30.3ms$	
==	$571ns$	$795ns$	(different matrices)
	$1.16ms$	$0.27ms$	(equal matrices)
M[i,j]	$5.69\mu s$	$2.24\mu s$	
M.list()	$529ms$	$15.4ms$	
skalar mult.	$9.1ms$	$1.6ms$	
nullspace	$23.5s$	$4.2s$	(1000×500)

MTX vs. usual SAGE-3.2.3 matrices

$\mathbb{F}_2$, random 5000×5000

	SAGE	MTX	
copy(M)	$4ms$	$4.67ms$	
loads(dumps(M))	$10.6s$	$623ms$	
$==$	$608ns$	$869ns$	(different matrices)
	$2.01ms$	$3.13ms$	(equal matrices)
M[i,j]	$1.95\mu s$	$2.28\mu s$	
M.list()	$1.3s$	$1.4s$	
nullspace	fails!	$55.2s$	(10000×5000)

M.kernel().basis() was running out of memory.

Conclusion

Conversion would not pay off (at least for now).

Typical generating sets

Minimal number of generators

#Gen	#gps
34	3
36	1
39	1
65	1

2322 groups have less than 34 minimal generators

The winner is:
Group 836

Maximal degree of minimal generators

Deg	#gps
13	1
14	3
16	22
17	1

2301 groups have generator degree less than 12

The winner is:
Group 562

Typical relation ideals

Minimal number of relations

#Rel	#gps
461	1
486	1
526	1
626	1
1859	1

2323 groups have less than 440 minimal relations

The winner is:
Group 836

Maximal degree of minimal relations

Deg	#gps
28	4
30	3
32	1
34	1

2319 groups have relation degree less than 27

The winner is:
Group 562
But 2298 and 2300 are hardest to obtain.

How good is Benson's criterion?

Failure of Improved Benson's Completeness Criterion

Excess	#gps
0	1779
1	341
2	168
3	39
4	1

The loser is: Group 2320,
which is the direct product of D_8
with an elementary abelian group.

Failure of Duflot's Depth Bound (2313 non-abelian groups)

depth - $\mathrm{rk}(Z(G))$	#gps
0	1767
1	508
2	37
3	1

The rank of the center is a lower bound for the depth of $H^*(G)$.
The winner is: Group 2326, extraspecial of type $+$

To Do

- Search counterexamples for Strong Benson Conjecture. Currently: Group 299 of order 256, which is of defect 4. It shows unexpected behaviour: 77 generators up to degree 11, $\sim$ 3000 relations up to degree 22, but still incomplete — and blocks 158 Gb hard disk...
- Add more features: Interesting invariants, Steenrod actions, $\mathbb{F}_p$ cohomology of general finite groups
- Build a data base and create a package

Status

- Various CYTHON modules, almost full doctest coverage
- A lot of C-code, various executables

— please help! THANK YOU!