Isomorphism testing

▸ Go to Classifications

▸ Go to Automorphisms

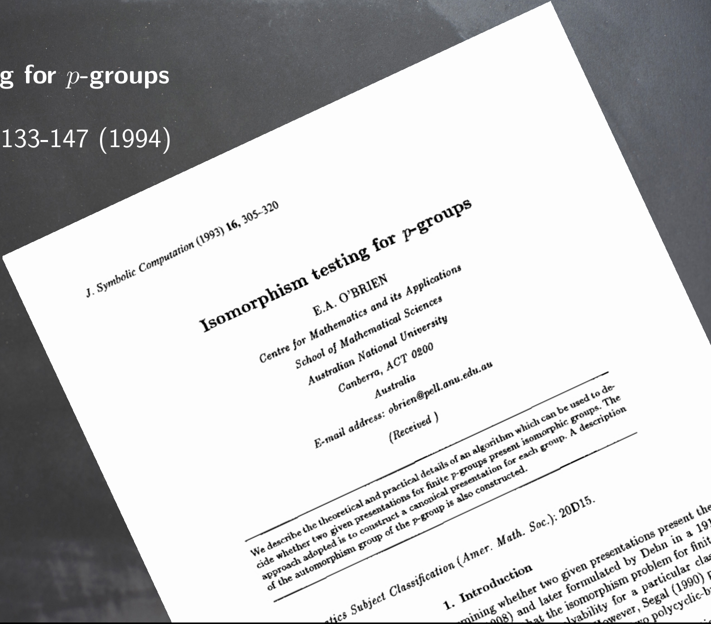
Conclusion Lecture 3

Things we have discussed in the third lecture:

- (immediate) descendants
- p-group generation algorithm
- p-cover, nucleus, multiplicator, allowable subgroups, extended auts
- automorphism groups of immediate descendants
- the group number gnu for group order p^5, p^6, p^7
- PORC conjecture

Resources

Isomorphism testing for p-groups
E. A. O'Brien
J. Symb. Comp. 17, 133-147 (1994)


J. Symbolic Computation (1993) 16, 305–320

Isomorphism testing for p-groups

E.A. O'BRIEN
Centre for Mathematics and its Applications
School of Mathematical Sciences
Australian National University
Canberra, ACT 0200
Australia
E-mail address: obrien@pell.anu.edu.au

(Received)

We describe the theoretical and practical details of an algorithm which can be used to decide whether two given presentations for finite p-groups present isomorphic groups. The approach adopted is to construct a canonical presentation for each group. A description of the automorphism group of the p-group is also constructed.

...tics Subject Classification (*Amer. Math. Soc.*): 20D15.

1. Introduction

Standard Presentations

Problem: Decide whether two p-groups are isomorphic.

Standard presentation

For a p-group G use methods from the p-quotient and p-group generation algorithms to construct a **standard pcp** (std-pcp) for G, such that $G \cong H$ if and only if G and H have the same std-pcp.

Example: For each $j = 1, \ldots, p-1$ the presentation

$$\mathrm{Pc}\langle a_1, a_2 \mid a_1^p = a_2^j,\ a_2^p = 1\rangle$$

is a wpcp describing C_{p^2}; as a std-pcp one could choose

$$\mathrm{Pc}\langle a_1, a_2 \mid a_1^p = a_2,\ a_2^p = 1\rangle.$$

Similarly, a std-pcp for C_p^d is $\mathrm{Pc}\langle a_1, \ldots, a_d \mid a_1^p = \ldots = a_d^p = 1\rangle$.

Isomorphism test: computing std-pcp's

Let G be d-generator p-group of p-class c.
Std-pcp of $G/P_1(G)$ is $\mathrm{Pc}\langle a_1, \ldots, a_d \mid a_1^p = \ldots = a_d^p = 1\rangle$.

Suppose $H \cong G/P_k(G)$ with $k < c$ is defined by std-pcp; have $\theta\colon G \to G/P_k(G)$.

Find std-pcp of $G/P_{k+1}(G)$ using p-group generation:

The p-group generation algorithm constructs immediate descendants of H.
Among these immediate descendants is $K \cong G/P_{k+1}(G)$. Proceed as follows:

- let $H \cong F/R$ (defined by std-pcp) and $H^* \cong F/R^*$;
- *evaluate relations* in H^* to get allowable M/R^* with $F/M \cong G/P_{k+1}(G)$;
- recall: $\alpha \in \mathrm{Aut}(H)$ acts as $\alpha^* \in \mathrm{Aut}(H^*)$ on allowable subgroups; two allowable U/R^* and V/R^* are in same $\mathrm{Aut}(H)$-orbit iff $F/U \cong F/V$; the choice of orbit rep determines the pcp obtained, and two elements from the same orbit determine different pcp's for isomorphic groups;
- associate with each allowable subgroup a unique *label*: a positive integer which runs from one to the number of allowable subgroups;
- let $\overline{M}/R^*$ be the element in the $\mathrm{Aut}(H)$-orbit of M/R^* with label 1.

Now $K = F/\overline{M}$ is isomorphic to $G/P_{k+1}(G)$; the pcp defining K is "standard".

Isomorphism test: example of std-pcp

The group

$$G = \langle x, y \mid (xyx)^3, x^{27}, y^{27}, [x,y]^3, (xy)^{27}, [y,x^3], [y^3,x] \rangle;$$

has order 3^7, rank 2, and 3-class 3; let $\mathcal{S}_1$ be the set of relators.

- $G/P_1(G)$ has std-pcp $H = \mathrm{Pc}\langle a_1, a_2 \mid a_1^3 = a_2^3 = 1 \rangle$, and we have an epimorphism $\theta\colon G \to H$ with $x, y \mapsto a_1, a_2$.
- use the p-quotient algorithm to construct covering

$$H^* = \mathrm{Pc}\langle\, a_1, \ldots, a_5 \mid [a_2, a_1] = a_3,\ a_1^3 = a_4,\ a_2^3 = a_5,\ a_3^3 = a_4^3 = a_5^3 = 1 \,\rangle.$$

- evaluate $\mathcal{S}_1$ in H^* via $\hat{\theta}$ to determine the allowable subgroup $U/R^* = \langle a_4^2 a_5 \rangle$ which must be factored from H^* to obtain $G/P_2(G)$, that is, F/U is isomorphic to $G/P_2(G)$ with wpcp

$$\mathrm{Pc}\langle a_1, \ldots, a_4 \mid [a_2, a_1] = a_3,\ a_1^3 = a_2^3 = a_4,\ a_3^3 = a_4^3 = 1 \rangle.$$

Isomorphism test: example of std-pcp

Recall:

$$H = \mathrm{Pc}\langle a_1, a_2 \mid a_1^3 = a_2^3 = 1\rangle;$$

$$H^* = \mathrm{Pc}\langle\, a_1, \ldots, a_5 \mid [a_2, a_1] = a_3,\ a_1^3 = a_4,\ a_2^3 = a_5,\ a_3^3 = a_4^3 = a_5^3 = 1\,\rangle,$$

with 3-multiplicator $M = \langle a_3, a_4, a_5\rangle$.

- A generating set for the automorphism group $\mathrm{Aut}(H) \cong \mathrm{GL}_2(3)$ is

$$\alpha_1: \begin{array}{lll} a_1 & \longmapsto & a_1a_2^2, \\ a_2 & \longmapsto & a_1^2a_2^2 \end{array} \quad \alpha_2: \begin{array}{lll} a_1 & \longmapsto & a_1, \\ a_2 & \longmapsto & a_1^2a_2 \end{array} \quad \alpha_3: \begin{array}{lll} a_1 & \longmapsto & a_1^2 \\ a_2 & \longmapsto & a_2 \end{array}$$

- Note that

$$\alpha_1^*(a_3) = \alpha_1^*([a_2, a_1]) = [a_1^2a_2^2, a_1a_2^2] = \ \ldots = a_3$$
$$\alpha_1^*(a_4) = \alpha_1^*(a_1^3) = (a_1a_2^2)^3 = \ \ldots = a_4a_5^2$$
$$\alpha_1^*(a_5) = \alpha_1^*(a_2^3) = (a_1^2a_2^2)^3 = \ \ldots = a_4^2a_5^2$$

so the matrices representing the action of α_i^* on M are

$$\begin{pmatrix} 1&0&0\\0&1&2\\0&2&2 \end{pmatrix}, \quad \begin{pmatrix} 1&0&0\\0&1&0\\0&2&1 \end{pmatrix}, \quad \begin{pmatrix} 2&0&0\\0&2&0\\0&0&1 \end{pmatrix}.$$

Isomorphism test: example of std-pcp

Recall that

$$H^* = \mathrm{Pc}\langle\, a_1, \ldots, a_5 \mid [a_2, a_1] = a_3, a_1^3 = a_4, a_2^3 = a_5,\; a_3^3 = a_4^3 = a_5^3 = 1 \,\rangle,$$

and $G/P_2(G) \cong F/U$ for the subspace $U/R^* = \langle a_4 a_5^2 \rangle$, which is $\langle (0,1,2) \rangle$

- The Aut(H)-orbit containing U/R^* is

$$\{\langle a_5\rangle, \langle a_4 a_5\rangle, \langle a_4^2 a_5\rangle, \langle a_4\rangle\}.$$

- The orbit rep with label 1 is ... $\bar{U}/R^* = \langle a_5 \rangle$.
- Factor H^* by $\langle a_5 \rangle$ to obtain the std-pcp for $G/P_2(G)$ as

$$K = \mathrm{Pc}\langle a_1, \ldots, a_4 \mid [a_2, a_1] = a_3,\; a_1^3 = a_4,\; a_1^3 = \ldots = a_4^3 = 1\rangle.$$

Recall that U/R^* was found by evaluating the relations $\mathcal{S}_1$ of G.
But: for the std-pcp we factored out $\bar{U}/R^* = \delta(U/R^*)$ for some $\delta \in$ Aut(H^*).
For the next iteration we need to modify the set of relations $\mathcal{S}_1$ accordingly.

Isomorphism test: example of std-pcp

- An extended automorphism which maps $U/R^* = \langle a_4 a_5^2 \rangle$ to $\bar{U}/R^* = \langle a_5 \rangle$ is

$$\delta : \quad \begin{array}{lcl} a_1 & \longmapsto & a_1a_2a_3a_4 \; = \; a_1a_2[a_2,a_1]a_1^3 \\ a_2 & \longmapsto & a_1a_2^2 \end{array}$$

- Apply δ to $\mathcal{S}_1 = \{(xyx)^3, x^{27}, y^{27}, [x,y]^3, \ldots\}$ to obtain

$$\mathcal{S}_2 = \{(xy[y,x]x^3xy^2xy[y,x]x^3)^3,\ (xy[y,x]x^3)^{27},\ (xy^2)^{27}, \ldots\};$$

  it follows that $G = \langle x, y \mid \mathcal{S}_1 \rangle \cong \langle x, y \mid \mathcal{S}_2 \rangle$, see O'Brien 1994.
- Now iterate with $G \cong \langle x, y \mid \mathcal{S}_2 \rangle$ and the std-pcp of $K \cong G/P_2(G)$ to compute the std-pcp of $G/P_3(G) \cong G$.

Practical issues: need *complete orbit* to identify element with smallest label. One idea is to exploit the characteristic structure of the p-multiplicator (as before).

Note: The std-pcp is only "standard" because it has been computed by some deterministic rule. Std-pcps are a very efficient tool to partition sets of groups into isomorphism classes.

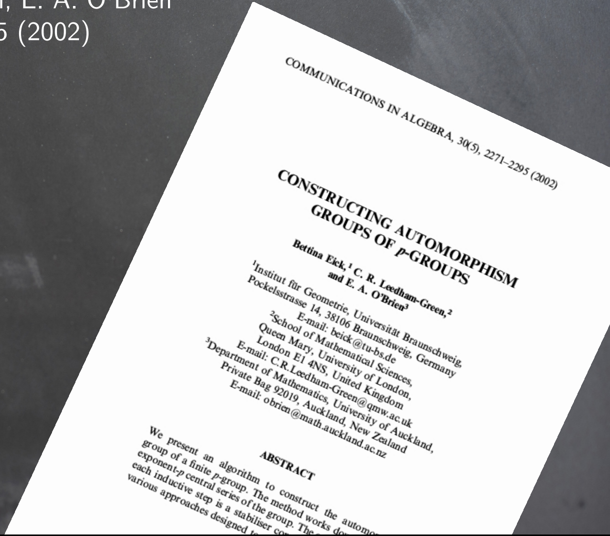
Automorphism groups

▸ Go to Isomorphisms

▸ Go to Coclass

Resources

Constructing automorphism groups of p-groups
B. Eick, C. R. Leedham-Green, E. A. O'Brien
Comm. Algebra 30, 2271-2295 (2002)


COMMUNICATIONS IN ALGEBRA, 30(5), 2271–2295 (2002)

CONSTRUCTING AUTOMORPHISM GROUPS OF p-GROUPS

Bettina Eick,[1] C. R. Leedham-Green,[2] and E. A. O'Brien[3]

[1]Institut für Geometrie, Universität Braunschweig, Pockelsstrasse 14, 38106 Braunschweig, Germany
E-mail: beick@tu-bs.de
[2]School of Mathematical Sciences, Queen Mary, University of London, London E1 4NS, United Kingdom
E-mail: C.R.Leedham-Green@qmw.ac.uk
[3]Department of Mathematics, University of Auckland, Private Bag 92019, Auckland, New Zealand
E-mail: obrien@math.auckland.ac.nz

ABSTRACT

We present an algorithm to construct the automo… group of a finite p-group. The method works do… exponent-p central series of the group. The… each inductive step is a stabiliser co… various approaches designed…

Computing automorphism groups

Let G be a d-generator p-group with lower p-central series

$$G = P_0(G) > P_1(G) > \ldots > P_c(G) = 1.$$

In the following write $G_i = G/P_i(G)$.

We want to construct Aut(G).

Approach

Compute Aut(G) = Aut(G_c) by induction on that series:

- Aut(G_1) = Aut$(C_p^d) \cong \mathrm{GL}_d(q)$
- construct Aut(G_{k+1}) from Aut(G_k).

For the induction step use ideas from p-group generation.

Computing automorphism groups

Let $H = G_k$ and $K = G_{k+1}$; given Aut(H), compute Aut(K).

Recall from p-group generation:

- compute $H^* = F/R^*$ and the multiplicator $M = R/R^*$;
- determine allowable subgroup $U/R^* \leq M$ defining K, that is, $K \cong F/U$;
- each $\alpha \in$ Aut(H) extends to $\alpha^* \in$ Aut(H^*) which leaves M invariant; via this construction, Aut(H) acts on the set of allowable subgroups;
- let Σ be the stabiliser of U/R^* in Aut(H) under this action;
- every $\alpha \in \Sigma$ defines an automorphism of $F/U \cong K$; let $S \leq$ Aut(K) be the subgroup induced by Σ;
- let $T \leq$ Aut(K) be the kernel of Aut$(K) \to$ Aut(H).

Theorem

With the previous notation, Aut$(K) = \langle S, T, \text{Inn}(K) \rangle$.

For a proof see O'Brien (1999).

Computing automorphism groups

Recall from p-group generation:

- $H = G/P_k(G)$ and $K = G/P_{k+1}(G)$; we have $K/P_k(K) \cong H$;
- K is quotient of H^* by allowable subgroup U/R^*;
- $S \leq \mathrm{Aut}(K)$ induced by stabiliser Σ of U/R^* in $\mathrm{Aut}(H)$
- $T \leq \mathrm{Aut}(K)$ is kernel of $\mathrm{Aut}(K) \to \mathrm{Aut}(H)$;
- $\mathrm{Aut}(K) = \langle S, T, \mathrm{Inn}(K) \rangle$.

Problem: how to determine S and T efficiently?

Lemma

Let $\{g_1, \ldots, g_d\}$ and $\{x_1, \ldots, x_l\}$ be minimal generating sets for K and $P_k(K)$, respectively. Define

$$\beta_{i,j} : K \to K, \quad \begin{cases} g_i \mapsto g_i x_j \\ g_n \mapsto g_n & (n \neq i). \end{cases}$$

Then $T = \langle \{\beta_{i,j} : 1 \leq i \leq d,\ 1 \leq j \leq l\} \rangle$, an elementary abelian p-group.

Main problem: Compute S, that is, the stabiliser Σ of U/R^* in $\mathrm{Aut}(H)$.

Induction step: example

Consider $G = \mathrm{Pc}\langle a_1, \ldots, a_4 \mid [a_2, a_1] = a_3, a_1^5 = a_4, a_2^5 = a_3^5 = a_4^5 = 1\rangle$;
this group has 5-class 2 with $P_1(G) = \langle a_3, a_4\rangle$.

Clearly, $H = G/P_1(G) = \mathrm{Pc}\langle a_1, a_2 \mid a_1^5 = a_2^5 = 1\rangle$ with $\mathrm{Aut}(H) \cong \mathrm{GL}_2(5)$.

Now compute:

- $H^* = \mathrm{Pc}\langle a_1, \ldots, a_5 \mid [a_2, a_1] = a_3, a_1^5 = a_4, a_2^5 = a_5, a_3^5 = a_4^5 = a_5^5 = 1\rangle$
- the allowable subgroup $U/R^* = \langle a_5\rangle$ yields G as a quotient of H^*
- $\alpha_1 \colon (a_1, a_2) \mapsto (a_1^2, a_2)$ and $\alpha_2 \colon (a_1, a_2) \mapsto (a_1^4 a_2, a_1^4)$ generate $\mathrm{Aut}(H)$; their extensions act on the multiplicator $\langle a_3, a_4, a_5\rangle$ as

$$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 1 \\ 0 & 4 & 0 \end{pmatrix}$$

- the stabiliser Σ of U/R^* is generated by the extensions of α_1 and $\alpha_2\alpha_1\alpha_2^2$
- a generating set for T is $\{\beta_{1,4}, \beta_{2,4}, \beta_{1,3}, \beta_{2,3}\}$

This yields indeed $\mathrm{Aut}(G) = \langle T, S, \mathrm{Inn}(G)\rangle$, where S is induced by Σ

Stabiliser problem

To do: Compute stabiliser of allowable subgroup U/R^* under action of Aut(H).

Our set-up is:

- consider $M = R/R^*$ as GF(p)-vectorspace and $V = U/R^*$ as subspace;
- represent the action of Aut(H) on M as a subgroup $A \leq \mathrm{GL}_m(p)$;
- compute the stabiliser of V in A.

Simple Approach: Orbit-Stabiliser Algorithm – constructs the whole orbit!

We'll briefly discuss the following ideas:

1. exploiting structure of M
2. exploiting structure of A
3. exploiting structure of K (and G)

Stabiliser problem: exploiting structure of M

Task: compute stabiliser of allowable subspace $V \leq M$ under A.

Idea: exploit the fact that $N = P_{k+1}(H^*) \leq M$ is characteristic in H^*, and that $M = NV$ (since V is allowable)

Use this to split stabiliser computation in two steps:

- compute the stabiliser of $V \cap N$ as subspace of N:

  use `MeatAxe` to compute composition series of N as A-module; then compute orbit and stabiliser of $V \cap N$ stepwise[7]

- compute orbit of $V/(V \cap N)$ as subspace of $M/(V \cap N)$:

  $V/(V \cap N)$ is complement to $N/(V \cap N)$ in $M/(V \cap N)$, and $N/(V \cap N)$ is A-invariant; compute A-module composition series of M/N and $N/(V \cap N)$ and break computation up in smaller steps

[7] see Eick, Leedham-Green, O'Brien (2002) for details

Stabiliser problem: exploiting structure of A

Task: compute stabiliser of allowable subspace $V \leq M$ under A.

Idea: Consider series $A \trianglerighteq S \trianglerighteq P \trianglerighteq 1$, where

- P induced by $\ker(H \to \mathrm{Aut}(H/P_1(H)))$, a normal p-subgroup
- S solvable radical, with $S = S_1 \rhd \ldots \rhd S_n \rhd P$, each section prime order.

Schwingel Algorithm for stabiliser under p-group P

One can compute a "canonical" representative of V^P and generators for $\mathrm{Stab}_P(V)$ **without** enumerating the orbit; see E-LG-O'B (2002).

Next, compute $\mathrm{Stab}_A(V)$ along $S = S_1 \rhd \ldots \rhd S_n \rhd P$, using the next lemma:

Lemma

Let L be a group acting on Ω; let $T \trianglelefteq L$ and let $\omega \in \Omega$.
Then ω^T is an L-block in Ω, and $\mathrm{Stab}_L(\omega^T) = T\mathrm{Stab}_L(\omega)$.

If $l \in \mathrm{Stab}_L(\omega^T)$, then $\omega^l = \omega^t$ for some $t \in T$, hence $lt^{-1} \in \mathrm{Stab}_L(\omega)$.

Stabiliser problem: exploiting structure of A

Compute $\mathrm{Stab}_A(V)$ along $S = S_1 \rhd \ldots \rhd S_n \rhd P$, using the next lemma:

Lemma

Let L be a group acting on Ω; let $T \trianglelefteq L$ and $\omega \in \Omega$.
Then ω^T is an L-block in Ω, and $\mathrm{Stab}_L(\omega^T) = T\mathrm{Stab}_L(\omega)$.

If orbit V^{S_i} and stabiliser $\mathrm{Stab}_{S_i}(V)$ are known, compute $\mathrm{Stab}_{S_{i-1}}(V^{S_i})$, and extend each generator to an element in $\mathrm{Stab}_{S_{i-1}}(V)$.

Advantage: Reduce the number of generators of $\mathrm{Stab}_S(V)$ substantially

Stabiliser problem: exploiting structure of K (and G)

Recall: we aim to construct Aut(G) by induction on lower p-central series with terms $G_i = G/P_i(G)$; initial step is Aut$(G_1) \cong \mathrm{GL}_d(p)$

Idea: Aut(G) induces a subgroup $R \leq$ Aut(G_1); instead of starting with Aut(G_1), start with $L \leq \mathrm{GL}_d(p)$ such that $R \leq L$ and $[L : R]$ is small.

Approach:

- construct a collection of characteristic subgroups of G, such as: centre, derived group, Ω, 2-step centralisers,...
- restrict this collection to $G_1 = G/P_1(G)$
- Schwingel has developed an algorithm to construct the subgroup $R \leq$ Aut$(G_1) \cong \mathrm{GL}_d(p)$ stabilising this lattice of subspaces of G_1

This aproach frequently reduces to small subgroups of $\mathrm{GL}_d(p)$ as initial group.

Conclusion Lecture 4

Things we have discussed in the forth lecture:

- std-pcp, isomorphism test for p-groups
- automorphism group computation

Lecture 4 is also the last lecture on the ANUPQ algorithms:

ANUPQ (ANU-p-Quotient program), 22,000 lines of C code developed by O'Brien; providing implementations of

- p-quotient algorithm
- p-group generation algorithm
- isomorphism test for p-groups
- automorphisms of p-groups

Implementations are also available in GAP and Magma; various papers discuss the theory and efficiency of these algorithms.

What's the Greek letter for "p" ...?

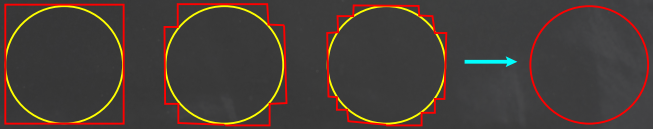
π

"Theorem"

We have $\pi = 4$.

Proof.

We take a unit circle with diameter 1 and approximate its circumference (which is defined to be π) by computing its arc-length. Remember how arc-length is defined? Use a polygonal approximation!



In every iteration: cirumference is π, arc lenght of red curve is 4.
So in the limit: $\pi = 4$, as claimed.

Well . . . obviously that is wrong!

Everyone knows that the following is true . . .

"Theorem"

We have $\pi = 0$.

Proof.

We start with Euler's Identity $1 = e^{2\pi\imath}$, which yields $e = e^{2\pi\imath+1}$. Now observe:

$$e = e^{2\pi\imath+1} = (e^{2\pi\imath+1})^{2\pi\imath+1} = e^{(2\pi\imath+1)^2} = e^{-4\pi^2} e e^{4\pi\imath}.$$

Since $e^{4\pi\imath} = 1$, this yields $1 = e^{-4\pi^2}$. Since $-4\pi^2 \in \mathbb{R}$, this forces $0 = -4\pi^2$. Since $-4 \neq 0$, we must have $\pi = 0$, as claimed.