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

Heiko Dietrich
(heiko.dietrich@monash.edu)
ICTS, Bangalore 2016

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: o'brien@maths.anu.edu.au

We describe the theoretical and practical details of an algorithm which can be used to determine 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.
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

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

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

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

Similarly, a std-pcp for $C_{p^d}$ is

$$Pc\langle a_1, \ldots, a_d \mid a_1^p = \ldots = a_d^p = 1 \rangle.$$
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 $\text{Pc}\langle a_1, \ldots, a_d \mid a_1^p = \ldots = a_d^p = 1 \rangle$.

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

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: $\alpha \in \text{Aut}(H)$ acts as $\alpha^* \in \text{Aut}(H^*)$ on allowable subgroups; two allowable $U/R^*$ and $V/R^*$ are in same $\text{Aut}(H)$-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 $\text{Aut}(H)$-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”.

Heiko Dietrich (heiko.dietrich@monash.edu) Computational aspects of finite $p$-groups ICTS, Bangalore 2016
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 $S_1$ be the set of relators.

- $G/P_1(G')$ has std-pcp $H = P \langle a_1, a_2 \mid a_1^3 = a_2^3 = 1 \rangle$,
  and we have an epimorphism $\theta: G \to H$ with $x, y \mapsto a_1, a_2$.
- use the $p$-quotient algorithm to construct covering
  \[ H^* = P \langle a_1, \ldots, a_5 \mid [a_2, a_1] = a_3, a_1^3 = a_4, a_2^3 = a_5, a_3^3 = a_4^3 = a_5^3 = 1 \rangle. \]
- evaluate $S_1$ 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
    \[ P \langle a_1, \ldots, a_4 \mid [a_2, a_1] = a_3, a_1^3 = a_2^3 = a_4^3 = a_3^3 = a_4^3 = 1 \rangle. \]
Isomorphism test: example of std-pcp

Recall:

\[ H = \langle a_1, a_2 \mid a_1^3 = a_2^3 = 1 \rangle; \]
\[ H^* = \langle a_1, \ldots, a_5 \mid [a_2, a_1] = a_3, a_1^3 = a_4, a_2^3 = a_5, a_3^3 = a_4 = a_5^3 = 1 \rangle, \]
with 3-multiplicator \( M = \langle a_3, a_4, a_5 \rangle \).

- A generating set for the automorphism group \( \text{Aut}(H) \cong \text{GL}_2(3) \) is
  \[
  \alpha_1 : \quad a_1 \mapsto a_1 a_2^2, \quad \alpha_2 : \quad a_1 \mapsto a_1, \quad \alpha_3 : \quad a_1 \mapsto a_1^2
  \]
  \[
  a_2 \mapsto a_1^2 a_2^2, \quad a_2 \mapsto a_1^2 a_2
  \]

- Note that
  \[
  \alpha_1^*(a_3) = \alpha_1^*(a_2, a_1) = [a_1^2 a_2^2, a_1 a_2^2] = \ldots = a_3
  \]
  \[
  \alpha_1^*(a_4) = \alpha_1^*(a_1^3) = (a_1 a_2^2)^3 = \ldots = a_4 a_5^2
  \]
  \[
  \alpha_1^*(a_5) = \alpha_1^*(a_2^3) = (a_1^2 a_2^2)^3 = \ldots = a_4^2 a_5^2
  \]

so the matrices representing the action of \( \alpha_i^* \) on \( M \) are
\[
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 2 \\
0 & 2 & 2
\end{pmatrix},
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 2 & 1
\end{pmatrix},
\begin{pmatrix}
2 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 1
\end{pmatrix}.
\]
Isomorphism test: example of std-pcp

Recall that

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

and \( G/P_2(G) \cong F/U \) for the subspace \( U/R^* = \langle a_4a_5^2 \rangle \), which is \( \langle (0, 1, 2) \rangle \)

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

  \[ \{ \langle a_5 \rangle, \langle a_4a_5 \rangle, \langle a_4^2a_5 \rangle, \langle a_4 \rangle \}. \]

- The orbit rep with label 1 is \( \ldots \bar{U}/R^* = \langle a_5 \rangle \).

- Factor \( H^* \) by \( \langle a_5 \rangle \) to obtain the std-pcp for \( G/P_2(G) \) as

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

Recall that \( U/R^* \) was found by evaluating the relations \( S_1 \) of \( G \).

But: for the std-pcp we factored out \( \bar{U}/R^* = \delta(U/R^*) \) for some \( \delta \in \text{Aut}(H^*) \).

For the next iteration we need to modify the set of relations \( S_1 \) 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

$$
\delta : \quad a_1 \mapsto a_1 a_2 a_3 a_4 = a_1 a_2 [a_2, a_1] a_1^3 \\
a_2 \mapsto a_1 a_2^2
$$

Apply $\delta$ to $S_1 = \{(xyx)^3, x^{27}, y^{27}, [x, y]^3, \ldots \}$ to obtain

$$
S_2 = \{(xy[y, x]x^3xy^2xy[y, x]x^3)^3, (xy[y, x]x^3)^{27}, (xy^2)^{27}, \ldots \};
$$

it follows that $G = \langle x, y \mid S_1 \rangle \cong \langle x, y \mid S_2 \rangle$, see O’Brien 1994.

Now iterate with $G \cong \langle x, y \mid S_2 \rangle$ 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
Resources

Constructing automorphism groups of $p$-groups
B. Eick, C. R. Leedham-Green, E. A. O’Brien
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 $\text{Aut}(G)$.

Approach

Compute $\text{Aut}(G) = \text{Aut}(G_c)$ by induction on that series:

- $\text{Aut}(G_1) = \text{Aut}(C'^d_p) \cong \text{GL}_d(q)$
- construct $\text{Aut}(G_{k+1})$ from $\text{Aut}(G_k)$.

For the induction step use ideas from $p$-group generation.
Computing automorphism groups

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

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

**Theorem**

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

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 \( U/R^* \);
- \( S \leq \text{Aut}(K) \) induced by stabiliser \( \Sigma \) of \( U/R^* \) in \( \text{Aut}(H) \);
- \( T \leq \text{Aut}(K) \) is kernel of \( \text{Aut}(K) \to \text{Aut}(H) \);
- \( \text{Aut}(K) = \langle S, T, \text{Inn}(K) \rangle \).

**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

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

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 \( \Sigma \) of \( U/R^* \) in \( \text{Aut}(H) \).
Induction step: example

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

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

Now compute:

- $H^* = \text{Pc}\langle a_1, \ldots, a_5 \mid [a_2, a_1] = a_3, a_1^5 = a_4, a_2^5 = a_5, a_3^5 = a_4^5 = a_5^5 = 1 \rangle$
- the allowable subgroup $U/R^* = \langle a_5 \rangle$ yields $G$ as a quotient of $H^*$
- $\alpha_1 : (a_1, a_2) \mapsto (a_1^2, a_2)$ and $\alpha_2 : (a_1, a_2) \mapsto (a_1^4 a_2, a_1^4)$ generate $\text{Aut}(H)$; their extensions act on the multiplicator $\langle a_3, a_4, a_5 \rangle$ as
  $$
  \begin{pmatrix}
  2 & 0 & 0 \\
  0 & 2 & 0 \\
  0 & 0 & 1
  \end{pmatrix},
  \begin{pmatrix}
  1 & 0 & 0 \\
  0 & 4 & 1 \\
  0 & 4 & 0
  \end{pmatrix}
  $$
- the stabiliser $\Sigma$ of $U/R^*$ is generated by the extensions of $\alpha_1$ 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 $\text{Aut}(G) = \langle T, S, \text{Inn}(G) \rangle$, where $S$ is induced by $\Sigma$. 
Stabiliser problem

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

Our set-up is:
- consider $M = R/R^*$ as GF($p$)-vectorspace and $V = U/R^*$ as subspace;
- represent the action of $\text{Aut}(H)$ on $M$ as a subgroup $A \leq \text{GL}_m(p)$;
- 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 $A \triangleright S \triangleright P \triangleright 1$, where
- $P$ induced by $\ker(H \to \text{Aut}(H/P_1(H)))$, a normal $p$-subgroup
- $S$ solvable radical, with $S = S_1 \triangleright \ldots \triangleright S_n \triangleright P$, each section prime order.

Schwingel Algorithm for stabiliser under $p$-group $P$

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

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

Lemma

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

If $l \in \text{Stab}_L(\omega^T)$, then $\omega^t = \omega^l$ for some $t \in T$, hence $lt^{-1} \in \text{Stab}_L(\omega)$.
Stabiliser problem: exploiting structure of $A$

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

**Lemma**

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

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

**Advantage:** Reduce the number of generators of $\text{Stab}_S(V)$ substantially.
Stabiliser problem: exploiting structure of $K$ (and $G$)

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

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

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

This approach frequently reduces to small subgroups of $\text{GL}_d(p)$ 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 “$p$” . . . ?
Automorphism Groups

Algorithm Example Stabiliser Problem

πππ

Heiko Dietrich (heiko.dietrich@monash.edu)

Computational aspects of finite $p$-groups

ICTS, Bangalore 2016
"Theorem"
We have $\pi = 4$.

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

In every iteration: circumference is $\pi$, arc length 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 $1 = e^{2\pi i}$, which yields $e = e^{2\pi i+1}$. Now observe:

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

Since $e^{4\pi i} = 1$, 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.