# CbA2Gen

## A classification of 2-generated cyclic-by-abelian finite p-groups

Version 1.0.0

22 June 2022

**Osnel Broche Cristo**
**Diego García-Lucas**
**Ángel del Río Mateos**

**Osnel Broche Cristo** Email: osnelc@gmail.com

**Diego García-Lucas** Email: diego.garcial@um.es

**Ángel del Río Mateos** Email: adelrio@um.es
Homepage: https://www.um.es/adelrio/

## Copyright



## Acknowledgements

We express our gratitude to Eamonn O'Brien for support on the ANUPQ package.

# Contents

# Chapter 1

# Cyclic-by-abelian 2-generated groups

In [BGLdR21] a complete classification of cyclic-by-abelian 2-generated finite $p$-groups is given. More concretely, to each group $G$ of this type a tuple

$$\mathrm{inv}(G) = (p, m, n_1, n_2, \sigma_1, \sigma_2, o_1, o_2, o_1', o_2', u_1, u_2)$$

is associated, verifying that if $H$ is another such group then $G \cong H$ if and only if $\mathrm{inv}(G) = \mathrm{inv}(H)$, and the set of possible values of $\mathrm{inv}(G)$ is described.

## 1.1 The vector of invariants

Let $G$ be a 2-generated non-abelian cyclic-by-abelian finite group of prime-power order. We start describing the meaning of the entries of $\mathrm{inv}(G)$. By the Burnside Basis Theorem [Rob82, 5.3.2], $G/G'$ is a 2-generated and non-cyclic, and the first four invariants $p$, $m$, $n_1$ and $n_2$ of $G$ are given by

$$|G'| = p^m \quad \text{and} \quad G/G' \cong C_{p^{n_1}} \times C_{p^{n_2}}, \quad \text{with } n_1 \geq n_2.$$

A *basis* of $G$ is a pair $b = (b_1, b_2)$ of elements of $G$ satisfying

$$G/G' = \langle b_1 G' \rangle \times \langle b_2 G' \rangle \quad \text{and} \quad |b_i G'| = p^{n_i} \ (i = 1, 2)$$

Let $\mathscr{B}$ denote the set of bases of $G$. Each basis determines a list of eight integers and our strategy consists in selecting bases for which this list satisfy a extreme condition with respect to a well order. This provides the additional eight entries of the list $\mathrm{inv}(G)$.

To define the integers associated to a basis we first define two maps $\sigma : G \to \{1, -1\}$ and $o : G \to \{0, 1, \ldots, m-1\}$ by setting:

$$\sigma(g) = \begin{cases} -1, & \text{if } a^g = a^{-1} \neq a \text{ for some } a \in G'; \\ 1, & \text{otherwise.} \end{cases}$$

$$o(g) = \begin{cases} 0, & \text{if } a^g = a^{-1} \text{ for every } a \in G'; \\ \log_p |gC_G(G')|, & \text{otherwise.} \end{cases}$$

So each basis $(b_1, b_2)$ of $G$ yields four integers $\sigma(b_i)$ and $o(b_i)$, $i = 1, 2$ and we use this to define the next four entries of $\mathrm{inv}(G)$ by setting

$$\sigma o = (\sigma_1, \sigma_2, o_1, o_2) = \min_{\text{lex}} \{(\sigma(b_1), \sigma(b_2), o(b_1), o(b_2)) : (b_1, b_2) \in \mathscr{B}\}$$

where $\min_{\text{lex}}$ denotes the minimum with respect to the lexicographical order. Let $r_1$ and $r_2$ be the unique integers $1 < r_i \leq 1 + p^m$ satisfying

$$r_1 \equiv \sigma_1(1+p^{m-o_1}) \mod p^m \quad \text{and} \quad \begin{cases} r_2 \equiv \sigma_2(1+p^{m-o_2}) \mod p^m, & \text{if } o_1 o_2 = 0; \\ r_2 \equiv \sigma_2(1+p^{m-o_1})^{p^{o_1-o_2}} \mod p^m, & \text{otherwise.} \end{cases} \tag{1.1}$$

By [BGLdR21, Proposition 2.3 ], the following set is not empty

$$\mathscr{B}_r = \{(b_1,b_2) \in \mathscr{B} : a^{b_i} = a^{r_i} \text{ for every } i = 1,2 \text{ and } a \in G'\}.$$

After this point only bases in $\mathscr{B}_r$ are used, and for each $b = (b_1,b_2) \in \mathscr{B}_r$ we denote by $t_1(b)$ and $t_2(b)$ the unique integers satisfying

$$1 \leq t_i(b) \leq p^m \quad \text{and} \quad b_i^{p^{n_i}} = [b_2,b_1]^{t_i(b)} \quad (i = 1,2).$$

Define $o'(b) = (o'_1(b), o'_2(b))$ and $u(b) = (u_2(b), u_1(b))$ by setting

$$o'_i(b) = \log_p(|b_i|) - n_i \quad \text{and} \quad t_i(b) = u_i(b) p^{m-o'_i(b)}$$

Observe that $|b_i| = p^{n_i+o'_i(b)}$ and hence $0 \leq o'_i(b) \leq m$ and $p \nmid u_i(b)$. We use this to define the next two entries of $\text{inv}(G)$ by setting

$$(o'_1, o'_2) = \max_{\text{lex}}\{o'(b) : b \in \mathscr{B}_r\}.$$

Then we define

$$\mathscr{B}'_r = \{b \in \mathscr{B}_r : o'(b) = (o'_1, o'_2)\}.$$

The two remaining entries of $\text{inv}(G)$ are given by

$$(u_2, u_1) = \min_{\text{lex}}\{u(b) : b \in \mathscr{B}'_r\}.$$

(Warning: The "unnatural" order in the $u$'s is not a typo but a convenient technicality.)

Setting

$$t_i = u_i p^{m-o'_i} \quad (i = 1,2). \tag{1.2}$$

we have that $G$ is isomorphic to the group given by the following presentation where $I$ is an abbreviation of $(p,m,n_1,n_2,\sigma_1,\sigma_2,o_1,o_2,o'_1,o'_2,u_1,u_2)$:

$$\mathscr{G}_I = \left\langle b_1, b_2 \mid [b_2,b_1]^{p^m} = 1, \quad [b_2,b_1]^{b_i} = [b_2,b_1]^{r_i}, \quad b_i^{p^{n_i}} = [b_2,b_1]^{t_i}, \quad (i = 1,2) \right\rangle \tag{1.3}$$

Hence $G$ is completely determined up to isomorphism by $\text{inv}(G)$. Therefore, to obtain our classification it only remains to give the list of tuples occurring as $\text{inv}(G)$. This is done in the main theorem of [BGLdR21], whose list of conditions we implement in the functions `IsInvariantVectorCbA2Gen(L)` and `CbA2GenByOrder(p, n)`.

### 1.1.1 IsInvariantVectorCbA2Gen

▷ `IsInvariantVectorCbA2Gen(L)` (operation)

Given a list $L$, this function returns `true` if there exists a non-abelian cyclic-by-abelian 2-generated finite $p$-group $G$ such that $L = \text{inv}(G)$ (i.e., if $L$ satisfies the conditions in the main theorem of [BGLdR21]). Otherwise it returns `false`.

### 1.1.2 CbA2GenByOrder

▷ `CbA2GenByOrder(p, n)` (operation)

Given a prime $p$ and an integer $n$, this function returns the list of all the possible $\mathrm{inv}(G)$ vectors (without repetitions) for $G$ a 2-generated non-abelian cyclic-by-abelian finite $p$-groups of order $p^n$. Thus the list returned by this function is in bijection with the list of isomorphism classes of groups of this type.

Example

```
gap> l:=CbAGenByOrder(2,15);;
gap> Size(l);
1505
gap> l[1000];
[ 2, 6, 7, 2, -1, 1, 1, 0, 1, 5, 1, 15 ]
gap> l[1001];
[ 2, 6, 7, 2, -1, 1, 2, 0, 0, 5, 1, 7 ]
```

This function is based in the following two auxiliary functions, which implements separately the lists satisfying the assumption in condition (6) and in condition (7), respectively, of the main theorem of [BGLdR21].

▷ `A(p, m, n1, n2)` (operation)

Given a prime $p$ and integers $m$, $n_1$ and $n_2$, this function returns the list of all the possible $\mathrm{inv}(G)$ vectors (without repetitions), for $G$ a 2-generated non-abelian cyclic-by-abelian finite $p$-group, such that its first five entries are exactly $(p, m, n_1, n_2, 1)$.

▷ `B(p, m, n1, n2)` (operation)

Given a prime $p$ and integers $m$, $n_1$ and $n_2$, this function returns the list of all the possible $\mathrm{inv}(G)$ vectors (without repetitions), for $G$ a 2-generated non-abelian cyclic-by-abelian finite $p$-group, such that its first five entries are exactly $(p, m, n_1, n_2, -1)$.

### 1.1.3 CbA2GenPcp

▷ `CbA2GenPcp(L)` (operation)

Given a list $L$, this function firstly checks, using the function `IsInvariantVectorCbA2Gen(L)`, if there exists a non-abelian cyclic-by-abelian 2-generated finite $p$-group $G$ such that $L = \mathrm{inv}(G)$. If this is not the case, it returns `fail`. Otherwise it computes a power-commutator presentation of such group $G$, and returns $G$ as a pc-group.

### 1.1.4 InvariantsAndBasis

▷ `InvariantsAndBasis(G)` (operation)

Given a finite group $G$, this function first check that $G$ is a non-abelian cyclic-by-abelian 2-generated finite $p$-group. If this is not the case, it returns `fail`. Otherwise it returns a list with

two entries. The first entry is the list $\mathrm{inv}(G)$ (with 12 entries), and the second entry is a list with two entries `[`$b_1$`,  `$b_2$`]`, where the pair $(b_1,b_2)$ belongs to $\mathscr{B}_{rt}$.

Example

```
gap> l:=CbAGenByOrder(2,15);;
gap> l[1000];
 [ 2, 6, 7, 2, -1, 1, 1, 0, 1, 5, 1, 15 ]
gap> l[1001];
[ 2, 6, 7, 2, -1, 1, 2, 0, 0, 5, 1, 7 ]
gap> G:=CycByAbelPcp(l[1000]);
<pc group of size 32768 with 15 generators>
gap> H:=CycByAbelPcp(l[1001]);
<pc group of size 32768 with 15 generators>
gap> IBG:=InvariantsAndBasis(G);
 [ [ 2, 6, 7, 2, -1, 1, 1, 0, 1, 5, 1, 15 ], [ f1, f8 ] ]
gap> IBH:=InvariantsAndBasis(H);
[ [ 2, 6, 7, 2, -1, 1, 2, 0, 0, 5, 1, 7 ], [ f1, f8 ] ]
```

▷ `Invariants(`*G*`)` (operation)

Given a finite group *G*, this function first check that $G$ is a non-abelian cyclic-by-abelian 2-generated finite $p$-group. If this is not the case, it returns `fail`. Otherwise it returns just the list with twelve entries $\mathrm{inv}(G)$.

▷ `AreIsomorphicGroups(`*G*`,`*H*`)` (operation)

Given a pair of finite groups *G* and *H*, this function first check if both groups are non-abelian cyclic-by-abelian 2-generated finite $p$-groups. If this is not the case, it returns `fail`. Otherwise it returns `true` if *G* and *H* are isomorphic, and it returns `false` if they are not. In order to do so, this function just computes `Invariants(`*G*`)` and `Invariants(`*H*`)` and compares the two resulting lists.

▷ `IsomorphismCbAGroups(`*G*`,`*H*`)` (operation)

Given a pair of finite groups *G* and *H*, this function first check if both groups are non-abelian cyclic-by-abelian 2-generated finite $p$-groups, and if that is the case it checks if they are isomorphic. If any of this conditions fails, it returns `fail`. Otherwise it returns an isomorphism between *G* and *H*. In order to build such isomorphism the bases of the groups obtained via the function `InvariantsAndBasis(`*G*`)` are used.

## 1.2 Comparison with ANUPQ and other checkings

### 1.2.1 Checking the results with ANUPQ

The functions in this subsection uses the $p$-group generating algorithm [O'B90] implemented in the GAP package ANUPQ-3.2.1 [GNOH22] check the accuracy of the main theorem of [BGLdR21] for

groups of small order.

▷ `DescendantsCbA2Gen(p,n)` (operation)

Given a prime $p$ and an integer $n$, this function returns the list of non-abelian cyclic-by-abelian 2-generated groups of order $p^n$ using the function *PqDescendants* of GAP package ANUPQ-3.2.1.

▷ `CheckNumber(p, n)` (operation)

Given a prime $p$ and an integer $n$, this function returns *true* if the number of non-abelian cyclic-by abelian 2-generated groups obtained via the function `DescendantsCbA2Gen(p,n)` is the same as the number of lists of invariants $\mathrm{inv}(G)$ allowed by the main theorem of [BGLdR21] (i.e., the number of list returned by the function `CbA2GenByOrder(p,n)`). Otherwise it returns `false`.

▷ `CheckIsoClasses(p, n)` (operation)

Given a prime $p$ and an integer $n$, this function returns `true` if the sets of list of invariants (obtained via the function `Invariants(G)`) of the non-abelian cyclic-by abelian 2-generated groups obtained via the function `DescendantsCbA2Gen(p,n)` is the same as the set of lists of invariants $\mathrm{inv}(G)$ allowed by the main theorem of [BGLdR21] (i.e., the number of list returned by the function `CbA2GenByOrder(p,n)`). Otherwise it returns `false`.

The authors have used the last two function to check that the results of [BGLdR21] are consistent with the lists of groups obtained via the $p$-group generating algorithm for $p$-groups of orders dividing $2^{12}$, $3^{11}$, $5^{10}$, $7^9$, $11^8$, $13^8$, $17^8$ and $23^8$.

### 1.2.2 Other checkings

▷ `CheckBasis(L)` (operation)

Given a list $L$ such that $L = \mathrm{inv}(G)$ for some non-abelian cyclic-by-abelian finite $p$-group $G$ (otherwise anything might happen), it returns `true` if the the vector of invariants returned by the function `InvariantsAndBasis(G)` applied to the group $G$ returned by the function `CbA2GenPcp(L)` coincides with the list $L$, and additionally the basis obtained from the same composition of functions belongs to $\mathscr{B}_{rt}$. Otherwise it returns `false`.

# References


[BGLdR21] O. Broche, D. García-Lucas, and Á. del Río. A classification of the finite two-generated cyclic-by-abelian groups of prime power order. 2021. http://arxiv.org/abs/2106.06449. 4, 5, 6, 7, 8

[GNOH22] G. Gamble, W. Nickel, E. O'Brien, and M. Horn. *ANUPQ, ANU p-Quotient, Version 3.2.6*, 2022. 7

[O'B90] E.A. O'Brien. The p-group generation algorithm. *Journal of Symbolic Computation*, 9(5):677–698, 1990. 7

[Rob82] D. J. S. Robinson. *A course in the theory of groups*, volume 80 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1982. 4