Efficient Isomorphism Testing for a Class of Group Extensions

François Le Gall

Symposium on Theoretical Aspects of Computer Science 2009 (Freiburg), pp. 625–636
www.stacs-conf.org

EFFICIENT ISOMORPHISM TESTING FOR A CLASS OF GROUP EXTENSIONS

FRANÇOIS LE GALL

ERATO-SORST Quantum Computation and Information Project,
Japan Science and Technology Agency, Tokyo
E-mail address: legall@qci.jst.go.jp

Abstract. The group isomorphism problem asks whether two given groups are isomorphic or not. Whereas the case where both groups are abelian is well understood and can be solved efficiently, very little is known about the complexity of isomorphism testing for nonabelian groups. In this paper we study this problem for a class of groups corresponding to one of the simplest ways of constructing nonabelian groups from abelian groups: the groups that are extensions of an abelian group A by a cyclic group $\mathbb{Z}_m$. We present an efficient algorithm solving the group isomorphism problem for all the groups of this class such that the order of A is coprime with m. More precisely, our algorithm runs in time almost linear in the orders of the input groups and works in the general setting where the groups are given as black-boxes.

1. Introduction

The group isomorphism problem is the problem of deciding, for two given groups G and H, whether there exists an isomorphism between G and H, i.e. a one-one map preserving the group operation. This is a fundamental problem in computational group theory but little is known about its complexity. It is known that the group isomorphism problem (for groups given by their multiplication tables) reduces to the graph isomorphism problem [12], and thus the group isomorphism problem is in the complexity class $NP \cap coAM$ (since the graph isomorphism problem is in this class [2]). Miller [14] has developed a general technique to check group isomorphism in time $O(n^{\log n+O(1)})$, where n denotes the size of the input groups and Lipton, Snyder and Zalcstein [13] have given an algorithm working in $O(\log^2 n)$ space. However, no polynomial algorithm is known for the general case of this problem.

Another line of research is the design of algorithms solving the group isomorphism problem for particular classes of groups. For abelian groups polynomial time algorithms follow directly from efficient algorithms for the computation of Smith normal form of integer matrices [10, 6]. More efficient methods have been given by Vikas [22] and Kavitha [11] for groups given by their multiplication tables. The current fastest algorithm solving the abelian group isomorphism problem for groups given as black-boxes has been developed

1998 ACM Subject Classification: F.2.2 Nonnumerical Algorithms and Problems.
Key words and phrases: polynomial-time algorithms, group isomorphism, black-box groups.

by Buchmann and Schmidt [5] and works in time $O(n^{1/2}(\log n)^{O(1)})$. However, as far as nonabelian groups are concerned, very little is known. For solvable groups Arvind and Torán [1] have shown that the group isomorphism problem is in $NP \cap coNP$ under certain complexity assumptions but, to our knowledge, the only polynomial-time algorithm testing isomorphism of a nontrivial class of nonabelian groups is a result by Garzon and Zalcstein [7], and holds for a very restricted class.

In this work we focus on the worst-case complexity of the group isomorphism problem over classes of nonabelian groups. Since for abelian groups the problem can be solved efficiently, we study one of the most natural next targets: cyclic extensions of abelian groups. Loosely speaking such extensions are constructed by taking an abelian group A and adding one element y that, in general, does not commute with the elements in A. More formally the class of groups we consider in this paper, denoted $\mathscr{S}$, is the following.

Definition 1.1. Let G be a finite group. We say that G is in the class $\mathscr{S}$ if there exists a normal abelian subgroup A in G and an element $y \in G$ of order coprime with $|A|$ such that $G = \langle A, y\rangle$.

In technical words G is an extension of an abelian group A by a cyclic group $\mathbb{Z}_m$ with $gcd(|A|, m) = 1$. This class of groups includes all the abelian groups and many non-abelian groups too. For example, for $A = \mathbb{Z}_3^4$ and $m = 4$ there are exactly 9 isomorphism classes in $\mathscr{S}$.

A group can be represented on a computer in different ways. In this paper we use the black-box setting introduced by Babai and Szemerédi [4], which is one of the most general models for handling groups, and particularly convenient to discuss algorithms running in sublinear time. In order to state precisely the running time of our algorithm, we introduce the following definition. For any group G in the class $\mathscr{S}$, let $\gamma(G)$ be the smallest integer m such that G is an extension of an abelian group A by the cyclic group $\mathbb{Z}_m$ with $gcd(|A|, m) = 1$. The main result of this paper is the following theorem.

Theorem 1.2. *There exists a deterministic algorithm checking whether two groups G and H in the class $\mathscr{S}$ given as black-box groups are isomorphic and, if this is the case, computing an isomorphism from G to H. Its running time has for upper bound $(\sqrt{n} + \gamma)^{1+o(1)}$, where $n = min(|G|, |H|)$ and $\gamma = min(\gamma(G), \gamma(H))$.*

Notice that, for any group G in the class $\mathscr{S}$, the relation $\gamma(G) \leq |G|$ holds. Then the complexity of our algorithm has for upper bound $n^{1+o(1)}$, and is almost linear in the size of the groups. Another observation is that, if $\gamma = O(n^{1/2})$, then the complexity of our algorithm is $n^{1/2+o(1)}$ and is of the same order as the best known algorithm testing isomorphism of abelian groups [5] in the black-box setting. This case $\gamma = O(n^{1/2})$ corresponds to the rather natural problem of testing isomorphism of extensions of a large abelian group by a small cyclic group.

The outline of our algorithm is as follows. Since a group G in the class $\mathscr{S}$ may in general be written as the extension of an abelian group A_1 by a cyclic group $\mathbb{Z}_{m_1}$ and as the extension of an abelian group A_2 by a cyclic group $\mathbb{Z}_{m_2}$ with $A_1 \ncong A_2$ and $m_1 \neq m_2$, we introduce (in Section 3) the concept of a standard decomposition of G, which is an invariant for the groups in the class $\mathscr{S}$ in the sense that two isomorphic groups have similar standard decompositions (but the converse is false). We also show how to compute a standard decomposition of G efficiently. This allows us to consider only the case where H and G are two extensions of the same abelian group A by the same cyclic group $\mathbb{Z}_m$. One of the main technical contributions of this paper is an efficient algorithm that tests whether two automorphisms of order m in the automorphism group of A are conjugate or not (Section 4). Finally, we present a time-efficient reduction from the problem of testing whether G and H are isomorphic to an instance of the above conjugacy problem (Section 5).

Remark. Several algorithms for the group isomorphism problem performing relatively well in practice are known and have been implemented in computational group theory softwares (GAP, MAGMA,...). The main works in this area are the algorithms developed by Smith for solvable groups [20] and by O'Brien [15] for p-groups. However these algorithms involve computation in groups of size exponential in n, e.g. the automorphism groups or the cohomology groups, and no rigorous analysis of their time complexity is available.

2. Preliminaries

We assume that the reader is familiar with the basic notions of group theory and state without proofs basic definitions and properties of groups we will use in this paper.

Let G be a finite group (in this paper we will consider only finite groups). Given a set S of elements of G, the subgroup generated by the elements of G is written $\langle S\rangle$. For any two elements $g, h \in G$ we denote $[g,h]$ the commutator of g and h, i.e. $[g,h] = ghg^{-1}h^{-1}$. The commutator subgroup of G is defined as $G' = \langle [g,h] \,|\, g,h \in G\rangle$. The derived series of G is defined recursively as $G^{(0)} = G$ and $G^{(i+1)} = (G^{(i)})'$. The group G is said to be solvable if there exists some integer k such that $G^{(k)} = \{e\}$.

Given a prime p, a p-group is a group of order p^r for some integer r. It is well-known that any p-group is solvable. If G is a group and $|G| = p_1^{e_i} \dots p_r^{e_r}$ for distinct prime numbers p_i such that $p_1 < \cdots < p_r$, then for each $i \in \{1,\dots,r\}$ the group G has a subgroup of order $p_i^{e_i}$ called a Sylow p_i-subgroup of G. Moreover, if G is additionally abelian, then each Sylow p_i-group is unique and G is the direct product of its Sylow subgroups. Abelian p-groups have remarkably simple structures: any abelian p-group P is isomorphic to a direct product of cyclic p-groups $\mathbb{Z}_{p^{e_1}} \times \cdots \times \mathbb{Z}_{p^{e_s}}$ for some positive integer s and positive integers $e_1 \leq \ldots \leq e_s$, and this decomposition is unique. A total order $\preceq$ over the set of prime powers can be defined as follows: for any two prime powers p^α and q^β where α and β are positive integers, we write $p^\alpha \preceq q^\beta$ if and only if $(p < q)$ or $(p = q$ and $\alpha \leq \beta)$. We say that a list $(g_1,\dots,g_t)$ of t elements in G is a basis of an abelian group G if $G = \langle g_1\rangle \times \cdots \times \langle g_t\rangle$, the order of each g_i is a prime power and $|g_i| \preceq |g_j|$ for any $1 \leq i \leq j \leq n$. It is easy to show that any (finite) abelian group has a basis and that, if $(g_1,\dots,g_t)$ and $(g'_1,\dots,g'_{t'})$ are two bases of G, then $t = t'$ and $|g_i| = |g'_i|$ for each $i \in \{1,\dots,t\}$. For example, $(g_1,\dots,g_t)$ is a basis of $G \cong \mathbb{Z}_2 \times \mathbb{Z}_4 \times \mathbb{Z}_3^2$ if and only if $t = 4$, $|g_1| = 2$, $|g_2| = 4$, $|g_3| = |g_4| = 3$ and $G = \langle g_1\rangle \times \langle g_2\rangle \times \langle g_3\rangle \times \langle g_4\rangle$.

Let n be a positive integer. A Hall divisor of n is a positive integer m dividing n such that m is coprime with n/m. A subgroup H of a finite group G is called a Hall subgroup of G if $|H|$ is a Hall divisor of $|G|$. We will use in this paper the following well-known theorem.

Theorem 2.1 (Hall's theorem)**.** *Let G be a finite solvable group and r be a Hall divisor of $|G|$. If H_1 and H_2 are two subgroups of G with $|H_1| = |H_2| = r$, then H_1 and H_2 are conjugate.*

We say that a finite group G is an extension of a group K by a group L if there exists a normal abelian subgroup $N \cong K$ of G such that $G/N \cong L$. We say that such an extension splits if there exists some subgroup M of G such that $G = NM$ and $N \cap M = \{e\}$. The

Schur-Zassenhaus theorem states that any extension of K by L such that $gcd(|K|,|L|)=1$ splits. Concretely, any such split extensions can be constructed as a semidirect product $K\rtimes L$. Thus an equivalent definition of the class $\mathscr{S}$ is the following: a group G is in $\mathscr{S}$ if and only if there exist an abelian group A and a cyclic group $\mathbb{Z}_m$ with $gcd(|A|,m)=1$ such that $G=A\rtimes\mathbb{Z}_m$.

In this paper we work in the black-box setting first introduced in [4]. A black-box group is a representation of a group where elements are represented by strings (of the same length). An oracle that performs the group product is available: given two strings representing two elements g and g', the oracle outputs the string representing $g\cdot g'$. Another oracle that, given a string representing an element g, computes a string representing the inverse g^{-1} is available as well. In this paper we assume the usual unique encoding hypothesis, i.e. any element of the group is encoded by a unique string. We say that a group G is input as a black-box if a set of strings representing generators $\{g_1,\ldots,g_s\}$ of G with $s=O(\log|G|)$ is given as input, and queries to the multiplication and inversion oracles can be done at cost 1. The hypothesis on s is natural since every group G has a generating set of size $O(\log|G|)$, and enables us to make the exposition of our results easier. Also notice that a set of generators of any size can be converted efficiently into a set of generators of size $O(\log|G|)$ if randomization is allowed [3].

3. Computing a Standard Decomposition

For a given group G in the class $\mathscr{S}$ in general many different decompositions as a semidirect product of an abelian group by a cyclic group exist. For example, the abelian group $\mathbb{Z}_6=\langle x_1,x_2\mid x_1^2=x_2^3=[x_1,x_2]=e\rangle$ can be written as $\langle x_1\rangle\times\langle x_2\rangle$, $\langle x_2\rangle\times\langle x_1\rangle$ or $\langle x_1,x_2\rangle\times\{e\}$. That is why we introduce the notion of a standard decomposition. Let us first start with a simple definition.

Definition 3.1. Let G be a finite group. For any positive integer m denote by $\mathscr{D}_G^m$ the set (possibly empty) of pairs (A,B) such that the following three conditions hold: (i) A is a normal abelian subgroup of G of order coprime with m; and (ii) B is a cyclic subgroup of G of order m; and (iii) $G=AB$.

Notice that if for some m the set $\mathscr{D}_G^m$ is not empty, then G is in the class $\mathscr{S}$. Conversely, if G is in $\mathscr{S}$, then there exists at least one integer m such that $\mathscr{D}_G^m$ is not empty. Also notice that $\gamma(G)$ is the smallest positive integer such that $\mathscr{D}_G^{\gamma(G)}\neq\varnothing$. We now define the concept of a standard decomposition.

Definition 3.2. Let G be a group in the class $\mathscr{S}$. A standard decomposition of G is an element of $\mathscr{D}_G^{\gamma(G)}$.

Before explaining how to compute a standard definition for a group in $\mathscr{S}$, let us mention that it is well known that the order of an element g of any finite group G can be computed deterministically in time $\tilde{O}(|G|^{1/2})$ using Shanks' baby-step/giant-step method [18] or its variants [19]. In the following proposition we show that the decomposition of an element in an abelian group can be found efficiently by a very similar approach (we will need this in Section 5).

Proposition 3.3. *Let A be an abelian group and $(g_1,\ldots,g_s)$ be a basis of A. There exists a deterministic algorithm with time complexity $\tilde{O}(|A|^{1/2})$ that, given any element $g\in A$, outputs integers $a_1,\ldots,a_s$ such that $g=g_1^{a_1}\cdots g_s^{a_s}$.*

Proof. Denote $r_i = \sqrt{|g_i|}$ for each $i \in \{1,\dots,s\}$ and, for simplicity, suppose that r_i is an integer. The case where r_i is not an integer is similar. The algorithm first computes the set $S = \{g_1^{c_1}\cdots g_s^{c_s} \mid c_i \in \{0,\dots,r_i-1\}\}$. Then the algorithm tries all the elements $(b_1,\dots,b_s)$ with $b_i \in \{0,\dots,r_i-1\}$ until finding an element $(\bar{b}_1,\dots,\bar{b}_s)$ such that $gg_1^{-\bar{b}_1 r_1}\cdots g_s^{-\bar{b}_s r_s} \in S$. Denote $gg_1^{-\bar{b}_1 r_1}\cdots g_s^{-\bar{b}_s r_s} = g_1^{c_1}\cdots g_s^{c_s}$, where each c_i is an element of $\{1,\dots,r_i-1\}$. A clever way for finding the c_i's is to use an appropriate data structure for storing S. Then the algorithm outputs $(r_1\bar{b}_1 + c_1,\dots,r_s\bar{b}_s + c_s)$. The correctness of this algorithm follows immediately from the fact that, if $g = g_1^{a_1}\cdots g_s^{a_s}$, then each a_i can be written as $a_i = \bar{b}_i r_i + c_i$ for some $\bar{b}_i$ and c_i in $\{0,\dots,r_i-1\}$. Its complexity is $\tilde{O}(|A|^{1/2})$. ■

We now show how to compute a standard decomposition of any group in the class $\mathscr{S}$ in time polynomial in the order of the group. The key part of the algorithm is the following procedure FIND-DECOMPOSITION that, given a group G in $\mathscr{S}$ and an integer m, computes an element of $\mathscr{D}_G^m$ if this set is not empty. The description is given in metacode, followed by more details.

```
Procedure FIND-DECOMPOSITION($G, m$)
      INPUT: a set of generators $\{g_1,\dots,g_s\}$ of a group $G$ in $\mathscr{S}$ with $s = O(\log|G|)$
             a positive integer $m$ dividing $|G|$
      OUTPUT: an error message or a pair $(M, z)$ where $z \in G$ and $M$ is a subset of $G$
 1    compute a set of generators $\{x_1,\dots,x_t\}$ of $G'$ with $t = O(\log|G|)$;
 2    factorize $m$ and write $m = p_1^{e_1}\cdots p_r^{e_r}$;
 3    search indexes $k_1,\dots,k_r \in \{1,\dots,s\}$ such that $p_\ell^{e_\ell}$ divides $|g_{k_\ell}|$ for each $1 \le \ell \le r$;
 4    if no such $r$-uple $(k_1,\dots,k_r)$ exists
 5       then return ERROR;
 6       else
 7             $g \leftarrow \Pi_{\ell=1}^{r} g_{k_\ell}^{|g_{k_\ell}|/p_\ell^{e_\ell}}$;
 8             if $m$ does not divide $|g|$
 9                then return ERROR;
10                else
11                      $z \leftarrow g^{|g|/m}$;
12                      for $j = 1$ to $s$ do $h_j \leftarrow g_j^m$;
13                      if $\langle x_1,\dots,x_t,h_1,\dots,h_s\rangle$ is abelian
                                  and $gcd(|x_i|, m) = 1$ for each $i \in \{1,\dots,t\}$
                                  and $gcd(|h_\ell|, m) = 1$ for each $\ell \in \{1,\dots,s\}$
14                         then return $(\{x_1,\dots,x_t,h_1,\dots,h_s\}, z)$;
15                         else return ERROR;
16                 endelse
17        endelse
```

At Step 1 a set of generators $\{z_1,\dots,z_{t'}\}$ of G' with $t' = O(s^3)$ can be computed using $O(s^3)$ group operations by noticing that $G' = \langle g_k[g_i,g_j]g_k^{-1} \mid i,j,k \in \{1,\dots,s\}\rangle$ (we refer to [9] for a proof of this simple fact). Since G' is abelian for any group G in the class $\mathscr{S}$, a generating set $\{x_1,\dots,x_t\}$ of G' with $t = O(\log|G|)$ can then be obtained in time $\tilde{O}(|G|^{1/2})$ using the deterministic algorithm by Buchmann and Schmidt [5] that computes a basis of any abelian group K in time $\tilde{O}(|K|^{1/2})$. At Step 2 the naive technique for factoring m (trying all the integers up to $\sqrt{m}$) is sufficient. This takes $\tilde{O}(|G|^{1/2})$ time. At Steps 3, 7 and 13 we use Shanks' method [18] to compute orders of elements of G in time $\tilde{O}(|G|^{1/2})$. At step 13, commutativity is tested by checking that every two generators commute: this can be done in $O(s^2+t^2)$ group operations. Proposition 3.6 below summarizes the time complexity of the procedure and prove its correctness. We state first two simple lemmas.

Lemma 3.4. *Let G be a group in $\mathscr{S}$ and m be any positive integer. If (A_1, B_1) and (A_2, B_2) are two elements of $\mathscr{D}_G^m$, then $A_1 = A_2$.*

Proof. Let us write $B_1 = \langle y_1\rangle$. Any element g of A_2 can be written as $g = hy_1^c$ with $h \in A_1$ and some integer c. If $c \not\equiv 0 \bmod m$, then $gcd(m, |g|) \neq 1$, which is excluded since $|A_2|$ and m are coprime. Then $A_2 \subseteq A_1$. By symmetry $A_1 \subseteq A_2$ and $A_1 = A_2$. ■

Lemma 3.5. *Let G be a group in $\mathscr{S}$ and (A, B) be a standard decomposition of G. Denote $|B| = m$. Let $\{g_1, \ldots, g_s\}$ be a set of generators of G. Then $A = \langle G', g_1^m, \ldots, g_s^m\rangle$, where G' is the derived subgroup of G.*

Proof. Let $B = \langle y\rangle$ and, for each $i \in \{1, \ldots, s\}$, write g_i as $z_i y^{k_i}$ for some $z_i \in A$ and $k_i \in \{1, \ldots, m\}$. Then $A = \langle G', z_1, \ldots, z_s\rangle$. Notice that G' has to be included since in general $A \neq \langle z_1, \ldots, z_s\rangle$, e.g. $G = \langle x_1, x_2, y \mid x_1^3 = x_2^3 = y^2 = e, yx_1 = x_2y, yx_2 = x_1y\rangle$ with the generating set $g_1 = x_1y$ and $g_2 = y$. A simple computation shows that $g_i^m = u_i z_i^m y^{mk_i} = u_i z_i^m$ for some element $u_i \in G'$. Since m is coprime with the order of z_i, we conclude that $A = \langle G', g_1^m, \ldots, g_s^m\rangle$. ■

Proposition 3.6. *The time complexity of the procedure* Find-Decomposition(G, m) *is $\tilde{O}(|G|^{1/2})$. If $\mathscr{D}_G^m \neq \varnothing$, then* Find-Decomposition(G, m) *outputs a pair (M, z) such that $(\langle M\rangle, \langle z\rangle)) \in \mathscr{D}_G^m$. Conversely, if* Find-Decomposition(G, m) *does not output an error message, then its output (M, z) is such that $\langle M, z\rangle \in \mathscr{S}$ and $(\langle M\rangle, \langle z\rangle) \in \mathscr{D}^m_{\langle M,z\rangle}$.*

Proof. It is clear that the procedure always terminates since no loop is used. The time complexity follows from the analysis of Steps 1, 2, 3, 7 and 13 already done, and from the fact that $s = O(\log|G|)$.

Suppose that $\mathscr{D}_G^m \neq \varnothing$ and take a decomposition $(A, \langle y\rangle) \in \mathscr{D}_G^m$. Write $m = p_1^{e_1} \cdots p_r^{e_r}$ for primes $p_1 < \cdots < p_r$ and denote $q_\ell = p_\ell^{e_\ell}$ for each $\ell \in \{1, \ldots, r\}$. Notice that for any generating set $\{g_1, \ldots, g_s\}$ of G, and for each $\ell \in \{1, \ldots, r\}$, there should be some index k_ℓ for which g_{k_ℓ} is of the form $u_\ell y^{c_\ell}$, where $u_\ell \in A$ and c_ℓ is such that q_ℓ divides the order of y^{c_ℓ}, i.e. q_ℓ divides $m/gcd(m, c_\ell)$. Also notice that in this case q_ℓ divides the order of g_{k_ℓ} as well. Then the element $\bar{g}_{k_\ell} = g_{k_\ell}^{|g_{k_\ell}|/q_\ell}$ has order q_ℓ and, more precisely, is of the form $v_\ell y^{d_\ell}$ for some $v_\ell \in A$ and some $d_\ell = \gamma_\ell m/q_\ell$ with γ_ℓ coprime with m. Then the element $g = \Pi_{\ell=1}^r \bar{g}_{k_\ell}$ is of the form wy^d where $w \in A$ and $d = d_1 + \cdots + d_r$ is coprime with m. Thus m divides $|g|$ and $z = g^{|g|/m}$ is an element of order m of the form $w'y^e$ with e coprime with m. From Lemma 3.5 we know that $\langle x_1, \ldots, x_t, h_1, \ldots, h_s\rangle = A$ and conclude that $(\langle x_1, \ldots, x_t, h_1, \ldots, h_s\rangle, \langle z\rangle) \in \mathscr{D}_G^m$.

We now prove the last part of the proposition. Suppose that the algorithm does not err and denote (M, z) its output. Then z has order m and $\langle M\rangle$ is an abelian subgroup of G of order coprime with m, since the tests at steps 8 and 13 succeeded. Moreover $\langle M\rangle$ is normal in G since $G' \leq \langle M\rangle$. We conclude that $\langle M, z\rangle \in \mathscr{S}$ and $(\langle M\rangle, \langle z\rangle) \in \mathscr{D}^m_{\langle M,z\rangle}$. ■

We now present an algorithm computing a standard decomposition of any group in $\mathscr{S}$.

Theorem 3.7. *There exists a deterministic algorithm that, on an input G in the class $\mathscr{S}$ given as a black box, outputs an element $z \in G$ and a set M of elements in G such* *that* $(\langle M\rangle, \langle z\rangle)$ *is a standard decomposition of* G. *The time complexity of this algorithm is* $O(|G|^{1/2+o(1)})$.

Proof. The algorithm is as follows. Let G be a group in the class $\mathscr{S}$, input as a black box with generating set $\{g_1,\ldots,g_s\}$ where $s = O(\log|G|)$.

We first compute $|g_i|$ for each $i \in \{1,\ldots,s\}$ using Shanks' algorithm. Let $\bar{m}$ be the least common multiple of the s integers $|g_1|,\ldots,|g_s|$. We compute the set S of divisors of $\bar{m}$, and denote $m_1 < m_2 < \cdots < m_r$ the elements of S in increasing order.

For i from 1 to r we run the procedure Find-Decomposition(G, m_i) on the set $\{g_1,\ldots,g_s\}$ and m_i, and obtain an error message or an output $(\langle M_i\rangle, z_i)$. Let n be the maximum value of the quantity $m_i|\langle M_i\rangle|$ over all the i's such that the output is not an error message (we will show that for at least one value of i the output is not an error message so n is well defined). Notice that computing $|M_i|$ can be done using the deterministic algorithm by Buchmann and Schmidt [5] that computes the order of any abelian group K in time $\tilde{O}(|K|^{1/2})$. Finally the algorithm takes the smallest integer $i_0 \in \{1,\ldots,r\}$ such that $m_{i_0}|M_{i_0}| = n$, and then outputs z_{i_0} and M_{i_0}.

We now analyze this algorithm. First of all notice that for any m such that $\mathscr{D}_G^m$ is not empty, this integer m is in S since m divides $\bar{m}$. By Proposition 3.6, if $\mathscr{D}_G^{m_i}$ is not empty then the procedure Find-Decomposition(G, m_i) outputs an element $(\langle M_i\rangle, \langle z_i\rangle) \in \mathscr{D}_G^{m_i}$ and then $m_i|\langle M_i\rangle| = |G|$. Conversely, and again by Proposition 3.6, if the procedure Find-Decomposition(G, m_i) outputs (M_i, z_i), then $m_i|\langle M\rangle| = |\langle z_i, M_i\rangle| \leq |G|$. Thus n is well defined and is equal to the order of G. Finally, trying all the elements of S gives clearly the minimal m such that $\mathscr{D}_G^m$ is not empty. Then $(\langle M_{i_0}\rangle, z_{i_0})$ is a standard decomposition of G. The time complexity of the algorithm is shown to be $|G|^{1/2+o(1)}$ using Proposition 3.6 and the following two facts. First, computing the set S can be done in $\tilde{O}(|G|^{1/2})$ time. Second, the number of divisors of any integer k has for upper bound $O(k^{\varepsilon})$ for any positive constant ε (see for example [8]). Since $\bar{m} \leq |G|$ we conclude that $r = |G|^{o(1)}$. ■

4. Testing Conjugacy

In this section we study the automorphism group of any abelian group and describe how to decide whether two automorphisms are conjugate.

Let A be a finite abelian group. Then A is the direct product of all its Sylow subgroups. Since $Aut(A)$ is the direct product of the automorphism groups of the Sylow subgroups, we can assume without loss of generality that A is an abelian p-group for some prime p. In this section we suppose that A is isomorphic to the group $\mathbb{Z}_{p^{e_1}} \times \cdots \times \mathbb{Z}_{p^{e_s}}$, for some positive integers s and $e_1 \leq e_2 \leq \ldots \leq e_s$. Let $(g_1,\ldots,g_s)$ be a basis of A, i.e. s elements of A such that the order of each g_i is p^{e_i} and such that $A = \langle g_1\rangle \times \cdots \times \langle g_s\rangle$.

4.1. Automorphisms of an abelian group

We first introduce a matricial characterization of the group $Aut(A)$ and study its structure.

Let ψ be an endomorphism of A and, for each $j \in \{1,\ldots,s\}$, denote $\psi(g_j) = g_1^{u_{1j}} \ldots g_s^{u_{sj}}$ where each u_{ij} is in the set $\{0,\ldots,p^{e_i}-1\}$. The values u_{ij}, which can be seen as an integer matrix (u_{ij}) of size $s \times s$, fully define the endomorphism ψ. However the converse is not true: an arbitrary integer matrix (u_{ij}) of size $s \times s$ with each value u_{ij} in $\{0,\ldots,p^{e_i}-1\}$ does not necessarily define an endomorphism of A, because ψ should be a homomorphism, and not only a linear map. It is easy to give necessary and sufficient conditions for these values u_{ij} to define an endomorphism of A: $p^{e_i - e_{min(i,j)}}$ should divide u_{ij} for any $i, j \in \{1, \ldots, s\}$.

More precisely, define $M(A)$ as the following set of integer matrices.

$$M(A) = \left\{(u_{ij}) \in \mathbb{Z}^{s\times s} \mid 0 \leq u_{ij} < p^{e_i} \text{ and } p^{e_i - e_{min(i,j)}} \text{ divides } u_{ij} \text{ for all } i, j \in \{1, \ldots, s\}\right\}.$$

Given U and U' in $M(A)$ we also define the multiplication $*$ as follows: $U * U'$ is the integer matrix W of size $s \times s$ such that $w_{ij} = (\sum_{k=1}^{s} u_{ik}u'_{kj} \bmod p^{e_i})$ for $i, j \in \{1, \ldots, s\}$, i.e. after computing the usual matrix multiplication UU', each entry is reduced modulo p^{e_i}, where i is the row of the entry. Let $R(A)$ be the set $R(A) = \{U \in M(A) \mid det(U) \not\equiv 0 \bmod p\}$. Ranum has shown [17] that the set $R(A)$ with the product operation $*$ is a group isomorphic to the group of automorphisms of A. Notice that a canonical isomorphism between $R(A)$ and $Aut(A)$ follows from the choice of a basis for A. An important example is the case $A = \mathbb{Z}_p^s$ for some integer s, for which $M(A)$ is the set of matrices of size $s \times s$ over the finite field $\mathbb{Z}_p$ and $R(A)$ is the general linear group $GL_s(p)$ of invertible matrices of size $s \times s$ over $\mathbb{Z}_p$.

Let us write $A \cong H_1 \times \cdots \times H_t$ with $H_i = \mathbb{Z}_{p^{f_i}}^{k_i}$ where $f_1 < f_2 < \cdots < f_t$ are positive strictly increasing integers and $k_1, \ldots, k_t$ are positive integers. Any matrix $M \in R(A)$ has t diagonal blocks $D_1, \ldots, D_t$ with $D_i \in GL_{k_i}(p)$ for $i \in \{1, \ldots, t\}$. Let Ψ be the map from $R(A)$ to $GL_s(p)$ such that any matrix $M \in R(A)$ is mapped as follows: the entries in the diagonal blocks are reduced modulo p; the other entries are set to zero. It is easy to show that Ψ is a group homomorphism from $Aut(A)$ to $GL_s(r)$. Let $N(A)$ denote its kernel and $V(A)$ denote its image. It is easy to see that N(A) is a subgroup of $R(A)$ of order p^r for some positive integer r, and that $V(A)$ is the subgroup of $GL_s(p)$ consisting of all the block diagonal matrices of the form $diag(D_1, \ldots, D_t)$ with $D_i \in GL_{k_i}(p)$ for $i \in \{1, \ldots, t\}$. We refer to [17] and to the full version of our paper for further details.

4.2. Testing conjugacy in $R(A)$

In this subsection we consider the following computational problem and present an efficient algorithm solving it.

Conjugacy
input: an abelian p-group A and two matrices U_1 and U_2 in $R(A)$ such that
$$\textit{the orders of } U_1 \textit{ and } U_2 \textit{ are coprime with } p \tag{4.1}$$
output: an element $U \in R(A)$ such that $U * U_1 = U_2 * U$ if such an element exists

Trying all the possibilities for U requires $|R(A)|$ trials. Since for example in the case $A = \mathbb{Z}_{p^k}^s$ with p and k constant the bound $|R(A)| = \Theta(|A|^{\log|A|})$ holds, such a naive approach is not efficient. However, notice that in the case $A = \mathbb{Z}_p^s$ the group A has more than the structure of an abelian group: A is a vector space over the field $\mathbb{Z}_p$ and then $R(A) = GL_s(p)$ as explained above. A mathematical criterion for the conjugacy of matrices in $GL_s(p)$ (even without the condition (4.1) on their orders) is known: two matrices are conjugate if and only if their canonical rational forms are equal. Since the canonical rational form of a matrix can be computed efficiently [21], this gives an algorithm solving the problem Conjugacy in time polynomial in $\log|A|$. However, when A has no vector space structure, there is no known simple mathematical criterion for the conjugacy of matrices and, to our knowledge, no algorithm faster than the above naive approach is known, even for the case where $A = \mathbb{Z}_{p^2}^s$. We now show that with the additional condition (4.1) on the order of U_1 and U_2 there exists an algorithm solving the problem CONJUGACY in time polynomial in $\log |A|$ for any abelian p-group A.

Our algorithm is based on the following proposition, which is a generalization of an argument by Pomfret [16].

Proposition 4.1. *Let A be an abelian p-group and U_1, U_2 be two matrices in $R(A)$ of order coprime with p. Then U_1 and U_2 are conjugate in $R(A)$ if and only if $\Psi(U_1)$ and $\Psi(U_2)$ are conjugate in $V(A)$. Moreover if U_1 and U_2 are conjugate in $R(A)$ then for any $X \in R(A)$ such that $\Psi(U_1) = \Psi(X)^{-1}\Psi(U_2)\Psi(X)$ there exists a matrix $Y \in N(A)$ such that $X * Y * U_1 = U_2 * X * Y$.*

Proof. For brevity we omit the symbol $*$ when denoting multiplications in $R(A)$. Since Ψ is an homomorphism, if U_1 and U_2 are conjugate in $R(A)$ then $\Psi(U_1)$ and $\Psi(U_2)$ are conjugate in $V(A)$. Now suppose that $\Psi(U_1)$ and $\Psi(U_2)$ are conjugate in $V(A)$. Since the image of Ψ is $V(A)$, there exists some $X \in R(A)$ such that $\Psi(U_1) = \Psi(X)^{-1}\Psi(U_2)\Psi(X)$ and thus $U_1 = X^{-1}U_2XM$ for some $M \in N(A)$. Then $\langle U_1\rangle N(A) = \langle X^{-1}U_2X\rangle N(A)$ (since $N(A)$ is a normal subgroup of $R(A)$) and the two subgroups $\langle U_1\rangle$ and $\langle X^{-1}U_2X\rangle$ are Hall subgroups of the group $\langle U_1\rangle N(A)$. Moreover since $\langle U_1\rangle N(A)$ is a cyclic extension of the p-group $N(A)$, this is a solvable group. Then, from Theorem 2.1, this implies that the two subgroups $\langle U_1\rangle$ and $\langle X^{-1}U_2X\rangle$ are conjugate in $\langle U_1\rangle N(A)$ and thus there exists an element $Y \in \langle U_1\rangle N(A)$ and some $r > 0$ such that $Y^{-1}X^{-1}U_2XY = U_1^r$. Without loss of generality Y can be taken in $N(A)$. Thus $\Psi(U_1) = \Psi(X)^{-1}\Psi(U_2)\Psi(X) = \Psi(U_1)^r$. Since the order of the kernel of Ψ is coprime with the order of U_1, the matrices U_1 and $\Psi(U_1)$ have the same order, and thus $U_1 = U_1^r$. We conclude that $Y^{-1}X^{-1}U_2XY = U_1$. The matrices U_1 and U_2 are thus conjugate in $R(A)$. The second part of the theorem follows from the observation that X can be chosen in an arbitrary way. ■

We now present our algorithm.

Theorem 4.2. *There exists a deterministic algorithm that solves the problem* CONJUGACY *in time polynomial in* $\log |A|$.

Proof. The algorithm is as follows.

Given U_1 and U_2 in $R(A)$ satisfying Condition (4.1), we first compute the two matrices $V_1 = \Psi(U_1)$ and $V_2 = \Psi(U_2)$ in $V(A)$. Then we check the conjugacy of V_1 and V_2 in $V(A)$ using the following approach. Let $D_i(V_1)$ (resp. $D_i(V_2)$) be the i-th diagonal block of V_1 (resp. V_2) for $i \in \{1, \ldots, t\}$, i.e. a matrix in $GL_{k_i}(p)$. The matrices V_1 and V_2 are conjugate in $V(A)$ if and only if the blocks $D_i(V_1)$ and $D_i(V_2)$ are conjugate in $GL_{k_i}(p)$ for each $i \in \{1, \ldots, t\}$, that is, if $D_i(V_1)$ and $D_i(V_2)$ have the same rational normal form. The rational normal form of matrices of size $n \times n$ (and transformation matrices) over any finite field can be computed using $O(n^4)$ field operations (see for example [21]). Thus we can decide in time polynomial in $\log |A|$ whether $D_i(V_1)$ and $D_i(V_2)$ are conjugate for all $i \in \{1, \ldots, t\}$. If this is not the case then we conclude that U_1 and U_2 are not conjugate in $R(A)$ from Proposition 4.1. Otherwise U_1 and U_2 are conjugate in $R(A)$ and the remaining of the proof shows how to compute a matrix $U \in R(A)$ such that $U * U_1 = U_2 * U$.

We compute transformation matrices $T_i \in GL_{k_i}(p)$, for $i \in \{1, \ldots, t\}$, such that $T_iD_i(V_1) = D_i(V_2)T_i$ using, for example, again the algorithm [21]. Then we take any matrix X in $R(A)$ such that $\Psi(X) = diag(T_1, \ldots, T_t)$, e.g. the matrix X in $R(A)$ with diagonal blocks equal to $T_1, \ldots, T_t$ and zero everywhere else. We finally determine a solution Y in

$N(A)$ of the matrix equation $X*Y*U_1 = U_2*X*Y$. Such solution exists by Proposition 4.1. To do this, we write the general form of an element Y of $N(A)$ using s^2 variables y_{ij}: the entry corresponding to the i-th row and the j-th column of Y, for $i,j \in \{1,\ldots,s\}$, is of the form $(1+py_{ij})$ if $i=j$ and is of the form $p^{d_{ij}}y_{ij}$ for some appropriate nonnegative integer d_{ij} otherwise. Then the equation $X*Y*U_1 = U_2*X*Y$ can be rewritten as the following system of s^2 linear modular equations of s^2 variables y_{ij}: $\sum_{i,j=1}^{s} \alpha_{ij}^{(k,\ell)} y_{ij} \equiv \beta^{(k,\ell)} \bmod p^{e_k}$ for $1 \leq k,\ell \leq s$, where $\alpha_{ij}^{(k,\ell)}$ and $\beta^{(k,\ell)}$ are known. Now we add on each modular equation a new variable $z_{k\ell}$ with coefficient p^{e_k}. This transforms the above system into the following system of s^2 linear Diophantine solutions of $2s^2$ variables: $\sum_{i,j=1}^{s} \alpha_{ij}^{(k,\ell)} y_{ij} + p^{e_k} z_{k\ell} = \beta^{(k,\ell)}$ for $1 \leq k,\ell \leq s$. It is known that any system of linear Diophantine equations with n_1 equations and n_2 variables can be solved in time polynomial in n_1, n_2 and $\log N$, where N is the largest coefficient appearing in the system [6]. Then a solution $Y \in N(A)$ of the equation $X*Y*U_1 = U_2*X*Y$ can be computed in time polynomial in $\log|A|$. The output of the algorithm is the matrix $X*Y$. ∎

5. Our Algorithm

In this section we give a proof of Theorem 1.2. We first present the following rather simple result that shows necessary and sufficient conditions for the isomorphism of two groups in $\mathscr{S}$.

Proposition 5.1. *Let G and H be two groups in $\mathscr{S}$. Let $(A_1, \langle y_1\rangle)$ and $(A_2, \langle y_2\rangle)$ be standard decompositions of G and H respectively and let φ_1 (resp. φ_2) be the action by conjugation of y_1 on A_1 (resp. of y_2 on A_2). The groups G and H are isomorphic if and only if the following three conditions hold: (i) $A_1 \cong A_2$; and (ii) $|y_1| = |y_2|$; and (iii) there exists an integer $k \in \{1,\ldots,|y_1|\}$ coprime with $|y_1|$ and an isomorphism $\psi : A_1 \to A_2$ such that $\varphi_1 = \psi^{-1}\varphi_2^k\psi$.*

Proof. First notice that for a group G in $\mathscr{S}$, the integer $\gamma(G)$ is a group invariant. Now suppose that G and H are two isomorphic groups in $\mathscr{S}$ with standard decomposition respectively $(A_1, \langle y_1\rangle)$ and $(A_2, \langle y_2\rangle)$. Then $|y_1| = |y_2| = \gamma(G) = \gamma(H)$. Denote by ψ an isomorphism from G to H and notice that $(\psi(A_1), \psi(y_1)) \in \mathscr{D}_H^{\gamma(H)}$. From Lemma 3.4 this implies that $\psi(A_1) = A_2$ and, in particular, $A_1 \cong A_2$. The element $\psi(y_1)$ can be written as zy_2^k for some $z \in A_2$ and some integer $k \in \{1,\ldots,\gamma(H)\}$ coprime with $\gamma(H)$. By definition of φ_1, for any $x \in A_1$ the relation $y_1x = (y_1xy_1^{-1})y_1 = \varphi_1(x)y_1$ holds. Applying ψ to each term gives $zy_2^k\psi(x) = \psi(\varphi_1(x))zy_2^k$ and then $\varphi_2^k(\psi(x))zy_2^k = \psi(\varphi_1(x))zy_2^k$ for any $x \in A_1$. Thus $\varphi_2^k = \psi\varphi_1\psi^{-1}$.

Now consider two groups G and H in $\mathscr{S}$ satisfying the conditions (i), (ii) and (iii) of the statement of the theorem. Denote $m = |y_1| = |y_2|$. Let μ be the map from G to H such that $\mu(xy_1^j) = \psi(x_1)y_2^{kj}$ for any x in A_1 and any $j \in \{0,\ldots,m-1\}$. The map μ is clearly a bijection from G to H. We now show that μ is a homomorphism, and thus an isomorphism from G to H. Let x and x' be two elements of A_1 and let j and j' be two elements of $\in \{0,\ldots,m-1\}$. Then $\mu(xy_1^jx'y_1^{j'}) = \mu(x\varphi_1^j(x')y_1^{j+j'}) = \psi(x\varphi_1^j(x'))y_2^{k(j+j')} = \psi(x)\psi(\varphi_1^j(x'))y_2^{k(j+j')}$. Now the relation $\mu(xy_1^j)\mu(x'y_1^{j'}) = \psi(x)y_2^{kj}\psi(x')y_2^{kj'} = \psi(x)\varphi_2^{kj}(\psi(x'))y_2^{k(j+j')}$ holds. Condition (iii) of the statement of the theorem implies that $\psi(\varphi_1^j(x')) = \varphi_2^{kj}(\psi(x'))$ and thus $\mu(xy_1^jx'y_1^{j'}) = \mu(xy_1^j)\mu(x'y_1^{j'})$. ∎

We now present our proof of Theorem 1.2.

Proof of Theorem 1.2. Suppose that G and H are two groups in the class $\mathscr{S}$. Denote $n = min(|G|,|H|)$ and $\gamma = min(\gamma(G),\gamma(H))$. Without loss of generality let us suppose that $|G| = |H|$. In order to test whether these two groups are isomorphic, we first run the algorithm of Theorem 3.7 on the inputs G and H and obtain outputs (S_1,y_1) and (S_2,y_2) such that $(\langle S_1\rangle,\langle y_1\rangle)$ and $(\langle S_2\rangle,\langle y_2\rangle)$ are standard decompositions of G and H respectively. The running time of this algorithm is $O(n^{1/2+o(1)})$ by Theorem 3.7. Denote $A_1 = \langle S_1\rangle$ and $A_2 = \langle S_2\rangle$.

We then check whether $|y_1| = |y_2|$. If $|y_1| \neq |y_2|$ we conclude that G and H are not isomorphic by Proposition 5.1. Otherwise notice that $|y_1| = |y_2| = \gamma$. Then we compute a basis $(g_1,\ldots,g_s)$ of A_1 and a basis $(h_1,\ldots,h_t)$ of A_2 using the algorithm by Buchmann and Schmidt [5]. The running time of this step is $\tilde{O}(n^{1/2})$. Given these bases it is easy to check the isomorphism of A_1 and A_2: the groups A_1 and A_2 are isomorphic if and only if $s = t$ and $|g_i| = |h_i|$ for each $i \in \{1,\ldots,s\}$. If $A_1 \ncong A_2$ we conclude that G and H are not isomorphic by Proposition 5.1.

Now suppose that $A_1 \cong A_2$ (and then $s = t$) and denote $R = R(A_1) = R(A_2)$. We want to decide whether the action by conjugation φ_1 of y_1 on A_1 and the action by conjugation φ_2 of y_2 on A_2 satisfy Condition (iii) in Proposition 5.1. Let $p_1^{d_1}\cdots p_r^{d_r}$ be the prime power decomposition of $|A_1| = |A_2|$, with $p_1 < \cdots < p_d$ and denote P_i the Sylow p_i-subgroup of A_1 for each $i \in \{1,\ldots,r\}$. We compute the matrix M_1 in R corresponding to the automorphism φ_1 of A_1 with respect to the basis $(g_1,\ldots,g_s)$. More precisely let us denote $\varphi_1(g_i) = y_1g_iy_1^{-1} = g_1^{u_{i1}}\cdots g_j^{u_{is}}$ for each $i \in \{1,\ldots,s\}$. The values u_{ij} for each i can be found by using the algorithm of Proposition 3.3 on the input $y_1g_iy_1^{-1}$. Then the matrix $M_1 = (u_{ij})$ can be computed in time $\tilde{O}(n^{1/2})$. Similarly we compute the matrix $M_2 \in R$ corresponding to the automorphism φ_2 of A_2 with respect to the basis $(h_1,\ldots,h_s)$. A key observation is that M_1 and M_2 are block diagonal, consisting in r blocks. More precisely the i-th block is a matrix in $R(P_i)$.

Finally for each integer $k \in \{1,\ldots,\gamma\}$ coprime with γ, we test whether M_1 and M_2^k are conjugate in R. This is done by using the algorithm of Theorem 4.2 to check whether, for each $i \in \{1,\ldots,r\}$, the i-th block of M_1 is conjugate to the i-th block of M_2 in $R(P_i)$. If there is no k such that M_1 and M_2^k are conjugate in R we conclude that G and H are not isomorphic. Otherwise we take one value k such that M_1 and M_2^k are conjugate and compute an explicit block diagonal matrix X in R such that $M_1 = X^{-1}M_2^kX$. This can be done in time polynomial in $\log n$ by Theorem 4.2. The matrix X is naturally associated to an isomorphism ψ from A_1 to A_2 through the bases $(g_1,\ldots,g_s)$ and $(h_1,\ldots,h_s)$. The map $\mu : G \to H$ defined as $\mu(xy_1^j) = \psi(x)y_2^{kj}$ for any $x \in A_1$ and any $j \in \{0,\ldots,\gamma-1\}$ is then an isomorphism from G to H (see the proof of Proposition 5.1 for details). The total complexity of this final step is $O(\gamma\log^c n)$ for some constant c.

The time complexity of this algorithm is $O(\gamma\log^c n) + O(n^{1/2+o(1)}) \leq (\sqrt{n}+\gamma)^{1+o(1)}$. ∎

Acknowledgments

The author is grateful to Yoshifumi Inui for many discussions on similar topics. He also thanks Igor Shparlinski and Erich Kaltofen for helpful comments.

References

[1] V. Arvind and J. Torán. Solvable group isomorphism. In *Proceedings of the 19th IEEE Conference on Computational Complexity*, pages 91–103, 2004.

[2] L. Babai. Trading group theory for randomness. In *Proceedings of the 17th annual ACM symposium on Theory of computing*, pages 421–429, 1985.

[3] L. Babai. Local expansion of vertex-transitive graphs and random generation in finite groups. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, pages 164–174, 1991.

[4] L. Babai and E. Szemerédi. On the complexity of matrix group problems I. In *Proceedings of the 25th Annual Symposium on Foundations of Computer Science*, pages 229–240, 1984.

[5] J. Buchmann and A. Schmidt. Computing the structure of a finite abelian group. *Mathematics of Computation*, 74(252):2017–2026, 2005.

[6] T. J. Chou and G. E. Collins. Algorithms for the solution of systems of linear diophantine equations. *SIAM Journal on Computing*, 11(4):687–708, 1982.

[7] M. H. Garzon and Y. Zalcstein. On isomorphism testing of a class of 2-nilpotent groups. *Journal of Computer and System Sciences*, 42(2):237–248, 1991.

[8] G. H. Hardy and E. M. Wright. *An introduction to the theory of numbers.* Oxford Science Publications, 1979.

[9] D. F. Holt, B. Eick, and E. A. O'Brien. *Handbook of computational group theory.* Chapman & Hall / CRC, 2005.

[10] R. Kannan and A. Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM Journal on Computing*, 8(4):499–507, 1979.

[11] T. Kavitha. Linear time algorithms for abelian group isomorphism and related problems. *Journal of Computer and System Sciences*, 73(6):986–996, 2007.

[12] J. Köbler, J. Torán, and U. Schöning. *The graph isomorphism problem: its structural complexity.* Birkhäuser, 1993.

[13] R. J. Lipton, L. Snyder, and Y. Zalcstein. The complexity of word and isomorphism problems for finite groups. Technical report, John Hopkins, 1976.

[14] G. Miller. On the $n^{\log n}$ isomorphism technique. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing*, pages 51–58, 1978.

[15] E. A. O'Brien. Isomorphism testing for p-groups. *Journal of Symbolic Computation*, 17:133–147, 1994.

[16] J. Pomfret. Similarity of matrices over finite rings. *Proceedings of the American Mathematical Society*, 37(2):421–422, 1973.

[17] A. Ranum. The group of classes of congruent matrices with application to the group of isomorphisms. *Transactions of the American Mathematical Society*, 8(1):71–91, 1907.

[18] S. Shanks. Class number, a theory of factorization and generata. *Proceedings of Symposia in Pure Mathematics*, 20(419-440), 1969.

[19] V. Shoup. *A computational introduction to number theory and algebra.* Cambridge University Press, 2005.

[20] M. Smith. *Computing automorphisms of finite soluble groups.* PhD thesis, Australian National University, 1994.

[21] A. K. Steel. Algorithm for the computation of canonical forms of matrices over fields. *Journal of Symbolic Computation*, 24(3/4):409–432, 1997.

[22] N. Vikas. An $O(n)$ algorithm for abelian p-group isomorphism and an $O(n \log n)$ algorithm for abelian group isomorphism. *Journal of Computer and System Sciences*, 53(1):1–9, 1996.