Symposium on Theoretical Aspects of Computer Science 2010 (Nancy, France), pp. 549-560
www.stacs-conf.org

AN EFFICIENT QUANTUM ALGORITHM FOR SOME INSTANCES OF THE GROUP ISOMORPHISM PROBLEM

FRANÇOIS LE GALL

Department of Computer Science, The University of Tokyo
E-mail address: legall@is.s.u-tokyo.ac.jp

Abstract. In this paper we consider the problem of testing whether two finite groups are isomorphic. 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. Le Gall has constructed an efficient classical algorithm for a class of groups corresponding to one of the most natural ways of constructing nonabelian groups from abelian groups: the groups that are extensions of an abelian group A by a cyclic group $\mathbb{Z}_m$ with the order of A coprime with m. More precisely, the running time of that algorithm is almost linear in the order of the input groups. In this paper we present a *quantum* algorithm solving the same problem in time polynomial in the *logarithm* of the order of the input groups. This algorithm works in the black-box setting and is the first quantum algorithm solving instances of the nonabelian group isomorphism problem exponentially faster than the best known classical algorithms.

1. Introduction

Testing group isomorphism (the problem asking to decide, for two given finite groups G and H, whether there exists an isomorphism between G and H) 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 [18], 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 [24] 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 [22] have given an algorithm working in $O(\log^2 n)$ space. However, no polynomial-time 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 the Smith normal form of integer matrices [8, 15]. More efficient methods have been given by Vikas [28] and Kavitha [16] for

1998 ACM Subject Classification: F.2.2 Nonnumerical Algorithms and Problems.
Key words and phrases: Quantum Algorithms, Group Isomorphism Problem, Black-box Groups.
This work was done while the author was a researcher at Kyoto University, affiliated with the ERATO-SORST Quantum Computation and Information Project, Japan Science and Technology Agency.

27th Symposium on Theoretical Aspects of Computer Science, Nancy, 2010
Editors: Jean-Yves Marion, Thomas Schwentick
Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany
Digital Object Identifier: 10.4230/LIPIcs.STACS.2010.2484

abelian groups given by their multiplication tables, and fast parallel algorithms have been constructed by McKenzie and Cook [23] for abelian permutation groups. The current fastest algorithm solving the abelian group isomorphism problem for groups given as black-boxes has been developed 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, until recently, the only polynomial-time algorithms testing isomorphism of nontrivial classes of nonabelian groups were a result by Garzon and Zalcstein [12], which holds for a very restricted class, and a body of works initiated by Cooperman et al. [9] on simple groups.

Very recently, Le Gall [19] proposed an efficient classical algorithm solving the group isomorphism problem over another class of nonabelian groups. Since for abelian groups the group isomorphism problem can be solved efficiently, that work focused on 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 considered in [19], denoted by $\mathscr{S}$, was the following.

Definition 1.1. Let G be a finite group. The group G is said to be in the class $\mathscr{S}$ if there exist 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, as discussed in details in [19]. For example, for $A = \mathbb{Z}_3^4$ and $m = 4$, there are exactly 9 isomorphism classes in $\mathscr{S}$ (1 class of abelian groups and 8 classes of nonabelian groups). Moreover, the class $\mathscr{S}$ includes several groups that have been the target of quantum algorithms, as discussed later. The main result in [19] was the following theorem.

Theorem 1.2 ([19])**.** *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 $n^{1+o(1)}$, where $n = min(|G|, |H|)$.*

In the present paper, we focus on *quantum algorithms* solving the group isomorphism problem in the black-box setting. Cheung and Mosca [7] have shown how to compute the decomposition of an abelian group into a direct product of cyclic subgroups in time polynomial in the logarithm of its order on a quantum computer, and thus how to solve the abelian group isomorphism problem in time polynomial in $\log n$ in the black-box model. This then gives an exponential speed-up with respect to the best known classical algorithms for the same task. One can naturally ask whether a similar speed-up can be obtained for classes of nonabelian groups. In this paper, we prove that this is the case. Our main result is the following theorem.

Theorem 1.3. *There exists a quantum algorithm checking with high probability 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 is polynomial in $\log n$, where $n = min(|G|, |H|)$.*

To our knowledge, this is the first quantum algorithm solving nonabelian instances of the group isomorphism problem exponentially faster than the best known classical algorithms.

Our algorithm relies on several new quantum reductions to instances of the so-called abelian Hidden Subgroup Problem, a problem that can be solved efficiently on a quantum computer. Our result can then be seen as an extension of the polynomial-time library of computational tasks which can be accomplished using Shor's factoring and discrete logarithm algorithms [27], and further quantum algorithms for abelian groups. We also mention that groups in the class $\mathscr{S}$ appear at several occasions in the quantum computation literature, mostly connected to the Hidden Subgroup Problem over semidirect product groups [4, 10, 13, 25]. Our techniques may have applications in the design of further quantum algorithms for this problem, or for other similar group-theoretic tasks.

Our quantum algorithm follows the same line as the classical algorithm in [19], but the two main technical parts are both significantly improved and modified.

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 use, as in [19], 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). A method for computing efficiently standard decompositions in the black-box model was one of the main contributions of [19], where the time complexity of this step was $O(n^{1+o(1)})$ due to the fact that the procedure proposed had to try, in the worst case, for each generator g of G, all the divisors of $|g|$. Instead, in the present work we propose a different procedure for this task (Section 3), which can be implemented in time polynomial in $\log n$ on a quantum computer, based on careful reductions to group-theoretic problems for which known efficient quantum algorithms are known: order finding, decomposing abelian groups and constructive membership in abelian groups.

Knowing standard decompositions of G and H 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 $\mathbb{Z}_m$ (Proposition 6.1). Two matrices M_1 and M_2 in the group $GL(r, \mathbb{F})$ of invertible matrices of size $r \times r$ over some well-chosen finite field $\mathbb{F}$ can then be associated to the action of $\mathbb{Z}_m$ on A in the groups G and H respectively. The second main technical contribution of [19] showed that, loosely speaking, testing isomorphism of G and H then reduces (when the order of A is coprime with m) to checking whether there exists an integer $k \in \{1, \ldots, m\}$ such that M_1 and M_2^k are conjugate in $GL(r, \mathbb{F})$. The strategy adopted in [19] to solve this problem had time complexity close to n in the worst case (basically, all the integers k in $\{1, \ldots, m\}$ were checked). In the present paper, we give a poly$(\log n)$ time quantum algorithm for this problem. More generally, we show in Section 5 that the problem of testing, for any two matrices M_1 and M_2 in $GL(r, \mathbb{F})$ where r is any positive integer and $\mathbb{F}$ is any finite field, whether there exists a positive integer k such that M_1 and M_2^k are conjugate in the group $GL(r, \mathbb{F})$ reduces to solving an instance of a problem we call Set Discrete Logarithm. This quantum reduction is efficient in that it can be implemented in time polynomial in both r and $\log |\mathbb{F}|$, and works by considering field extensions of $\mathbb{F}$ and matrix invariants of M_1 and M_2.

Loosely speaking, the problem Set Discrete Logarithm asks, given two sets $\{x_1, \ldots, x_v\}$ and $\{y_1, \ldots, y_v\}$ of elements in $\mathbb{F}$, to compute an integer k such that $\{y_1^k, \ldots, y_v^k\} = \{x_1, \ldots, x_v\}$, if such an integer exists. This computational problem is a generalization of the standard discrete logarithm problem (which is basically the case $v = 1$) but appears to be much more challenging. The quantum algorithm we propose (in Section 4) works in time polynomial in v and $\log |\mathbb{F}|$, and relies on a reduction to several instances of the abelian

Hidden Subgroup Problem. Our solution to the problem Set Discrete Logarithm is then an extension of the computational tasks which can be solved efficiently using known quantum algorithms for abelian groups.

2. Preliminaries

2.1. Group theory and standard decompositions

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

For any positive integer m, we denote by $\mathbb{Z}_m$ the additive cyclic group of integers $\{0,\ldots,m-1\}$, and by $\mathbb{Z}_m^*$ the multiplicative group of integers in $\{1,\ldots,m-1\}$ coprime with m.

Let G be a finite group. For any subgroup H and any normal subgroup K of G we denote by HK the subgroup $\{hk \mid h \in H, k \in K\} = \{kh \mid h \in H, k \in K\}$. Given a set S of elements of G, the subgroup generated by the elements of S is written $\langle S \rangle$. We say that two elements g_1 and g_2 of G are conjugate in G if there exists an element $y \in G$ such that $g_2 = y g_1 y^{-1}$. For any two elements $g, h \in G$ we denote by $[g,h]$ the commutator of g and h, i.e., $[g,h] = ghg^{-1}h^{-1}$. More generally, given two subsets S_1 and S_2 of G, we define $[S_1,S_2] = \langle [s_1,s_2] \mid s_1 \in S_1, s_2 \in S_2 \rangle$. The commutator subgroup of G is defined as $G' = [G,G]$. 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 two groups G_1 and G_2, a map $\phi : G_1 \to G_2$ is a homomorphism from G_1 to G_2 if, for any two elements g and g' in G_1, the relation $\phi(gg') = \phi(g)\phi(g')$ holds. We say that G_1 and G_2 are isomorphic if there exists a one-one homomorphism from G_1 to G_2, and we write $G_1 \cong G_2$.

Given any finite group G, we denote by $|G|$ its order and, given any element g in G, we denote by $|g|$ the order of g in G. For any prime p, we say that a group is a p-group if its order is a power of p. If $|G| = p_1^{e_i} \ldots p_r^{e_r}$ for distinct prime numbers p_i, then for each $i \in \{1,\ldots,r\}$ the group G has a subgroup of order $p_i^{e_i}$. Such a subgroup is 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 is isomorphic to a direct product of cyclic p-groups $\mathbb{Z}_{p^{f_1}} \times \cdots \times \mathbb{Z}_{p^{f_s}}$ for some positive integer s and positive integers $f_1 \leq \ldots \leq f_s$, and this decomposition is unique. We say that a set $\{g_1,\ldots,g_t\}$ of elements of an abelian group G is a basis of G if $G = \langle g_1 \rangle \times \cdots \times \langle g_t \rangle$ and the order of each g_i is a prime power.

For a given group G in the class $\mathscr{S}$ in general many different decompositions as an extension 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, as it was done in [19].

Definition 2.1. Let G be a finite group in the class $\mathscr{S}$. 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) $G = AB$. Let $\gamma(G)$ be the smallest positive integer such that $\mathscr{D}_G^{\gamma(G)} \neq \varnothing$. A standard decomposition of G is an element of $\mathscr{D}_G^{\gamma(G)}$.

2.2. Black-box groups

In this paper we work in the black-box model. A black-box group is a representation of a group G where elements are represented by strings, and an oracle is available to perform group operations. To be able to take advantage of the power of quantum computation when dealing with black-box groups, the oracles performing group operations have to be able to deal with quantum superpositions. These quantum black-box groups have been first studied by Ivanyos et al. [14] and Watrous [29, 30], and have become the standard model for studying group-theoretic problems in the quantum setting.

More precisely, a quantum black-box group is a representation of a group where elements are represented by strings (of the same length, supposed to be logarithmic in the order of the group). We assume the usual unique encoding hypothesis, i.e., each element of the group is encoded by a unique string, which is crucial for technical reasons (without it, most quantum algorithms do not work). A quantum oracle V_G is available, such that $V_G(|g\rangle|h\rangle) = |g\rangle|gh\rangle$ for any g and h in G (using strings to represent the group elements), and behaving in an arbitrary way on other inputs. 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 oracle 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.

Any efficient quantum black-box algorithm gives rise to an efficient concrete quantum algorithm whenever the oracle operations can be replaced by efficient procedures. Especially, when a mathematical expression of the generators input to the algorithm is known, performing group operations can be done directly on the elements in polynomial time (in $\log |G|$) for many natural groups, including permutation groups and matrix groups.

Quantum algorithms are very efficient for solving computational problems over abelian groups. In the following theorem, we describe the main results we will need in this paper.

Theorem 2.2 ([7, 14, 27])**.** *There exists quantum algorithms solving, in time polynomial in* $\log |G|$*, the following computational tasks with probability at least* $1 - 1/poly(|G|)$*:*

(i) *Given a group G given as a black-box (with unique encoding) and any element $g \in G$, compute the order of g in G.*
(ii) *Given an abelian group G given as a black-box (with unique encoding), compute a basis $(g_1, \ldots, g_s)$ of G.*
(iii) *Given an abelian group G given as a black-box (with unique encoding), a basis $(g_1, \ldots, g_s)$ of G, and any $g \in G$, compute a decomposition of g over $(g_1, \ldots, g_s)$, i.e., integers $u_1, \ldots, u_s$ such that $g = g_1^{u_1} \cdots g_s^{u_s}$.*

Actually, all the tasks in Theorem 2.2 can be seen as black-boxes versions of instances of the so-called Hidden Subgroup Problem (HSP) over abelian groups. It is known that the abelian HSP can be solved in time polynomial in $\log |G|$ [17], even if G is given as a black-box group with unique encoding [14, 26].

3. Computing a Standard Decomposition

In this section we present a quantum algorithm computing a standard decomposition of any group in the class $\mathscr{S}$ in time polynomial in the logarithm of the order of the group.

The precise description of the algorithm, which we denote Procedure DECOMPOSE, is given in metacode in Figure 1. Further descriptions on how each step is implemented follow.

Procedure DECOMPOSE

INPUT: a set of generators $\{g_1, \ldots, g_s\}$ of a group G in $\mathscr{S}$ with $s = O(\log |G|)$.
OUTPUT: a pair (U, v) where U is a subset of G and $v \in G$.

1 compute generators $\{g'_1, \ldots, g'_t\}$ of the derived subgroup G' with $t = O(\log |G|)$;
2 compute $\kappa = lcm(|g_1|, \ldots, |g_s|)$;
3 factorize κ and write $\kappa = p_1^{e_1} \cdots p_r^{e_r}$ where the prime numbers p_i are distinct;
4 $U \leftarrow \{g'_1, \ldots, g'_t\}$; $V \leftarrow \varnothing$; $\Sigma \leftarrow \varnothing$;
5 **for** $i = 1$ **to** r
6 **do**
7 $\Gamma_i \leftarrow \varnothing$;
8 **for** $j = 1$ **to** s **do** $\Gamma_i \leftarrow \Gamma_i \cup \{g_j^{\kappa/p_i^{e_i}}\}$;
9 **if** $[\Gamma_i, G'] = e$ **and** $gcd(p_i, |G'|) \neq 1$ **then** $U \leftarrow U \cup \Gamma_i$;
10 **if** $[\Gamma_i, G'] = e$ **and** $gcd(p_i, |G'|) = 1$
11 **then**
12 search for an element $\gamma_i \in \Gamma_i$ such that $\langle \Gamma_i \rangle G' = \langle \gamma_i, G' \rangle$;
13 **if** no such element exists
14 **then** $U \leftarrow U \cup \Gamma_i$
15 **else** $\Sigma \leftarrow \Sigma \cup \{\gamma_i\}$;
16 **endthen**
17 **if** $[\Gamma_i, G'] \neq e$ **then** { take an element $\gamma_i \in \Gamma_i$ such that $|\gamma_i| = \max_{\gamma \in \Gamma_i} |\gamma|$;
18 $V \leftarrow V \cup \{\gamma_i\}$; }
19 **enddo**
20 **for** all w in Σ
21 **do**
22 **if** there exists an element z in Σ such that $[w, z] \neq e$
23 **then** { **if** $zwz^{-1} \in \langle w \rangle$ **then** $U \leftarrow U \cup \{w\}$ **else** $V \leftarrow V \cup \{w\}$; }
24 **enddo**
25 **for** all $w \in \Sigma \backslash (U \cup V)$
26 **do**
27 **if** $[w, u] = \{e\}$ for all $u \in U$ **then** $U \leftarrow U \cup \{w\}$ **else** $V \leftarrow V \cup \{w\}$;
28 **enddo**
29 $b \leftarrow \Pi_{g \in V} |g|$; $z \leftarrow \Pi_{g \in V} g$; $v \leftarrow z^{|z|/b}$;
30 output (U, v);

Figure 1: Procedure DECOMPOSE.

- At Step 1 a set of generators $\{g'_1, \ldots, g'_t\}$ of the derived subgroup G' with $t = O(\log |G|)$ is computed in time polynomial in $\log |G|$ with success probability $1 - 1/\text{poly}(|G|)$ using the classical algorithm by Babai et al. [3].
- The order of G' at Steps 9 and 10, and the orders of elements at Steps 2, 17 and 29 are computed using the quantum algorithms for Tasks (i) and (ii) in Theorem 2.2.
- The least common multiple at Step 2 is computed using standard algorithms, and is factorized at Step 3 using Shor's factoring algorithm [27].

- At Step 12, notice that $[\Gamma_i, G'] = e$ implies that $\langle \Gamma_i \rangle G'$ is an abelian group. For each element γ_i in Γ_i (there are $O((\log |G|)^2)$ such elements), the quantum algorithms for Tasks (i) and (ii) in Theorem 2.2 are used to check whether $|\langle \Gamma_i \rangle G'| = |\langle \gamma_i, G' \rangle|$. Since necessarily $\langle \gamma_i, G' \rangle \leq \langle \Gamma_i \rangle G'$, this test is sufficient to check whether $\langle \Gamma_i \rangle G' = \langle \gamma_i, G' \rangle$.
- The tests at Steps 9, 10 to 17 are done by noticing that $[\Gamma_i, G'] = \{e\}$ if and only if $[\gamma, g'_j] = e$ for each $\gamma \in \Gamma_i$ and each $j \in \{1, \ldots, t\}$.
- Testing whether zwz^{-1} is in $\langle w \rangle$ at Step 23 is done by trying to decompose zwz^{-1} over $\langle w \rangle$ using the quantum algorithm for Task (iii) in Theorem 2.2, and then checking if the decomposition indeed represents zwz^{-1} (since, a priori, this algorithm can have an arbitrary behavior when $zwz^{-1} \notin \langle w \rangle$).

This description, along with Theorem 2.2 and with the observation that the sets U, V and Σ have size $O((\log |G|)^2)$, show that all the steps of Procedure DECOMPOSE can be implemented in time polynomial in $\log |G|$. The following theorem states its correctness.

Theorem 3.1. *Let G be a group in the class $\mathscr{S}$, given as a black-box group (with unique encoding). The procedure* DECOMPOSE *on input G outputs, with high probability, a pair (U, v) such that $(\langle U \rangle, \langle v \rangle)$ is a standard decomposition of G. It can be implemented in time polynomial in $\log |G|$ on a quantum computer.*

A complete proof of Theorem 3.1 can be found in the full version of this paper [20].

4. Set Discrete Logarithm

We first introduce the following useful notation. Let $\mathbb{F}$ be a finite field, and $\Sigma = \{x_1, \ldots, x_t\}$ be any subset of $\mathbb{F}$ with possible repetitions, i.e., all the x_i's are elements of $\mathbb{F}$, but may not be distinct. For any integer k, we denote by Σ^k the subset of $\mathbb{F}$ with possible repetitions $\{x_1^k, \ldots, x_t^k\}$.

In this section we consider the following problem. Here u is a positive integer which is a parameter of the problem (taking $u \geq 2$ does not make the problem significantly harder, but this enables us to give a more convenient presentation of our results).

SET DISCRETE LOGARITHM
INPUT: two lists $(S_1, \ldots, S_u)$ and $(T_1, \ldots, T_u)$ where, for each integer $h \in \{1, \ldots, u\}$, S_h and T_h are subsets with possible repetitions of some finite field $\mathbb{F}_h$.
OUTPUT: a positive integer k such that $T_h^k = S_h$ for all $h \in \{1, \ldots, u\}$, if such k exists.

Notice that the case $u = 1$ with $|S_1| = |T_1| = 1$ is the usual discrete logarithm problem over the multiplicative group of the field $\mathbb{F}_1$. Actually, our algorithm solving the problem SET DISCRETE LOGARITHM will only need the multiplicative structure of the fields, and then also works if we replace in the definition each field $\mathbb{F}_h$ by any multiplicative finite group G_h. However, since the main applications of our algorithm deal with field structures, we describe our results in the present slightly less general form.

Given an instance of SET DISCRETE LOGARITHM, let m_S denote the smallest positive integer such that $x^{m_S} = 1$ for all $x \in S_1 \cup \cdots \cup S_u$, and let m_T denote the smallest positive integer such that $y^{m_T} = 1$ for all $y \in T_1 \cup \cdots \cup T_u$. The main result of this section is the following theorem.

Theorem 4.1. *There exists a quantum algorithm that solves with high probability the problem* Set Discrete Logarithm*, and runs in time polynomial in* u, $\log(m_S + m_T)$, *and* $\max_{1\leq h\leq u}(|S_h| + |T_h| + \log|\mathbb{F}_h|)$.

Proof. For the sake of brevity, let us denote $\Sigma = S_1 \cup \cdots \cup S_u \cup T_1 \cup \cdots \cup T_u$. We first compute the orders of all the elements in Σ using Shor's algorithm [27]. The value m_S is the least common multiple of the orders of all the elements in $S_1 \cup \cdots \cup S_u$, and the value m_T is the least common multiple of the orders of all the elements in $T_1 \cup \cdots \cup T_u$. The values m_S and m_T can then be computed in time polynomial in $\log(m_S + m_T)$, $|\Sigma|$, and $\max_{1\leq h\leq u} \log|\mathbb{F}_h|$. Notice that, for any positive integer k, the least common multiple of the orders of all the elements in $T_1^k \cup \cdots \cup T_u^k$ is $m_T/gcd(k, m_T)$. Then, if m_S does not divide m_T, there is no solution to the problem Set Discrete Logarithm. If m_S divides m_T but $m_S \neq m_T$, then a solution (if it exists) can be found by replacing the list $(T_1, \ldots, T_u)$ by the list $(T_1^{m_T/m_S}, \ldots, T_u^{m_T/m_S})$. Thus, without loss of generality, we suppose hereafter that $m_S = m_T$ and denote by m this value. Then a solution k can be searched for in the set $\mathbb{Z}_m^*$.

Let $\{m_1, \ldots, m_\ell\} = \cup_{z\in\Sigma}\{|z|\}$ denote the set of orders of the elements in Σ. For each $h \in \{1, \ldots, u\}$ and each $i \in \{1, \ldots, \ell\}$, we define the subsets

$$S_{h,i} = \{x \in S_h \mid |x| = m_i\} \text{ and } T_{h,i} = \{y \in T_h \mid |y| = m_i\}.$$

Let us also define the sets

$$K_{h,i} = \{k \in \mathbb{Z}_m^* \mid T_{h,i}^k = S_{h,i}\} \text{ and } \overline{K}_{h,i} = \{k \in \mathbb{Z}_m^* \mid T_{h,i}^k = T_{h,i}\}.$$

It is straightforward to check that the set $\overline{K}_{h,i}$ is a subgroup of $\mathbb{Z}_m^*$, and that the set $K_{h,i}$ is either empty, or is a coset of $\overline{K}_{h,i}$ in $\mathbb{Z}_m^*$.

Let $K \subseteq \mathbb{Z}_m^*$ denote the set of solutions of the instance of Set Discrete Logarithm we are considering. Then

$$K = \bigcap_{1\leq h\leq u} \Big(\bigcap_{1\leq i\leq \ell} K_{h,i} \Big).$$

The set K can be computed efficiently using a quantum computer if, for each $h \in \{1, \ldots, u\}$ and each $i \in \{1, \ldots, \ell\}$, the set $K_{h,i}$ is known. More precisely, this is done by using a quantum algorithm for computing the intersections of two cosets of an abelian group — more details can be found in the full version of this paper [20].

The final part of the proof shows how to compute these sets $K_{h,i}$. Let us fix an integer $h \in \{1, \ldots, u\}$ and an integer $i \in \{1, \ldots, \ell\}$. We suppose that $S_{h,i}$ and $T_{h,i}$ have the same size (otherwise $K_{h,i} = \varnothing$ and thus $K = \varnothing$). Denote $S_{h,i} = \{x_1, \ldots, x_v\}$ and $T_{h,i} = \{y_1, \ldots, y_v\}$, where $v = |S_{h,i}|$ depends on h and i. We present a quantum procedure computing a set of generators of $\overline{K}_{h,i}$, and an element $k_{h,i}$ in $K_{h,i}$ when this set is not empty, in time polynomial in v, $\log m$, and $\log|\mathbb{F}_h|$.

We first show how to compute the subgroup $\overline{K}_{h,i}$. Let $\prec$ be an arbitrary strict total ordering of the elements of $\mathbb{F}_h$. Without loss of generality we can suppose that $x_1 \preceq x_2 \preceq \cdots \preceq x_v$. Let μ be the function from $\mathbb{Z}_m^* \times \{1, \ldots, v\}$ to $\mathbb{F}_h$ defined as follows: for any $k \in \mathbb{Z}_m^*$ and any $j \in \{1, \ldots, v\}$, $\mu(k, j)$ is the j-th element (with respect to the order $\prec$) of the set $T_{h,i}^k$. Let f be the function from $\mathbb{Z}_m^*$ to $(\mathbb{F}_h)^v$ such that, for any $k \in \mathbb{Z}_m^*$:

$$f(k) = (\mu(k, 1)y_1^{-1}, \ldots, \mu(k, v)y_v^{-1}).$$

Notice that the set $\{k \in \mathbb{Z}_m^* \mid f(k) = (1, \ldots, 1)\}$ is precisely the subgroup $\overline{K}_{h,i}$ of $\mathbb{Z}_m^*$. Moreover, the function f is constant on cosets of $\overline{K}_{h,i}$ in $\mathbb{Z}_m^*$, with distinct values on distinct cosets (since $f(k_1) = f(k_2)$ implies that $T_{h,i}^{k_1} = T_{h,i}^{k_2}$ and thus $k_1 \in k_2\overline{K}_{h,i}$). This is thus an instance of the abelian HSP, and a set of generators of $\overline{K}_{h,i}$ can be found in time polynomial in v, $\log m$ and $\log |\mathbb{F}_h|$ using the algorithm described in Subsection 2.2 (notice that the underlying group is $\mathbb{Z}_m^*$, and that the value of the function f can be computed in time v, $\log m$ and $\log |\mathbb{F}_h|$).

We now show how to compute an element $k_{h,i}$ in $K_{h,i}$ if this set is not empty. We first try to find an element $\alpha \in \mathbb{Z}_{m_i}^*$ such that $T_{h,i}^{\alpha} = S_{h,i}$. This is done by, for each $j \in \{1, \ldots, v\}$, trying to find an integer $\alpha_j \in \mathbb{Z}_{m_i}^*$ such that $x_1^{\alpha_j} = y_j$, if such an integer exists (notice that, for each j, there is at most one element α_j in $\mathbb{Z}_{m_i}^*$ satisfying this condition, which can be computed in time polynomial in $\log m_i$ and $\log |\mathbb{F}_h|$ using the quantum algorithm for the standard discrete logarithm problem [27]) and checking whether $T_{h,i}^{\alpha_j} = S_{h,i}$. If no such value α can be found, we conclude that $K_{h,i}$ is empty. Otherwise we take any such value α and compute $k_{h,i}$ as follows. Let us write the prime power decomposition of m as $m = p_1^{\epsilon_1} \cdots p_r^{\epsilon_r} p'^{\eta_1}_1 \cdots p'^{\eta_s}_s q_1^{\delta_1} \cdots q_t^{\delta_t}$, where each prime p_l divides m_i for $l \in \{1, \ldots, r\}$, each prime p'_l divides α but not m_i for $l \in \{1, \ldots, s\}$, and each prime q_l divides neither m_i nor α for $l \in \{1, \ldots, t\}$. Then the integer

$$k_{h,i} = \alpha + m_i q_1^{\delta_1} \cdots q_t^{\delta_t} \bmod m$$

is coprime with m (since α is coprime with m_i and then each prime p_l, p'_l or q_l does not divide $k_{h,i}$), and hence is in $\mathbb{Z}_m^*$. From the choice of α and since any element in $T_{h,i}$ has order m_i, we conclude that $k_{h,i}$ is in the set $K_{h,i}$. ∎

5. Discrete Logarithm up to Conjugacy

Given a positive integer r and a finite field $\mathbb{F}$, remember that $GL(r, \mathbb{F})$ denotes the multiplicative group of invertible matrices of size $r \times r$ with entries in $\mathbb{F}$. In this section we consider the following problem. Here u is again a positive integer which is a parameter of the problem.

Discrete Log up to Conjugacy

INPUT: two lists of matrices $(M_1^{(1)}, \ldots, M_1^{(u)})$ and $(M_2^{(1)}, \ldots, M_2^{(u)})$ where, for each integer $h \in \{1, \ldots, u\}$, $M_1^{(h)}$ and $M_2^{(h)}$ are in $GL(r_h, \mathbb{F}_h)$ for some positive integer r_h and some finite field $\mathbb{F}_h$.

OUTPUT: a positive integer k and u matrices $M^{(h)} \in GL(r_h, \mathbb{F}_h)$ such that $M^{(h)} \cdot M_1^{(h)} = [M_2^{(h)}]^k \cdot M^{(h)}$ for each $h \in \{1, \ldots, u\}$, if such elements exist.

In the statement of the above problem, the notation $[M_2^{(h)}]^k$ simply means $M_2^{(h)}$ raised to the k-th power. Notice that the case $u = 1$ and $r_1 = 1$ is basically the usual discrete logarithm problem over the multiplicative group of the finite field $\mathbb{F}_1$.

Let m_1 and m_2 denote the smallest positive integers such that $[M_1^{(h)}]^{m_1} = I$ and $[M_2^{(h)}]^{m_2} = I$ for all $h \in \{1, \ldots, u\}$. The main result of this section is the following theorem.

Theorem 5.1. *There exists a quantum algorithm that solves with high probability the problem* Discrete Log up to Conjugacy*, and runs in time polynomial in* u, $\log(m_1+m_2)$, *and* $\max_{1\leq h\leq u}(r_h+\log|\mathbb{F}_h|)$

The quantum algorithm solving the problem Discrete Log up to Conjugacy follows from an efficient reduction to the problem Set Discrete Logarithm, using the concepts of elementary divisors, Jordan normal forms and similarity of matrices. A complete proof of Theorem 5.1 is given in the full version of this paper [20].

6. Proof of Theorem 1.3

We will need the following result from [19] that shows necessary and sufficient conditions for the isomorphism of two groups in the class $\mathscr{S}$.

Proposition 6.1 (Proposition 5.1 in [19])**.** *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 $\varphi_1\in Aut(A_1)$ (resp. $\varphi_2\in Aut(A_2)$) 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 exist a positive integer k and an isomorphism $\chi\colon A_1\to A_2$ such that $\varphi_1=\chi^{-1}\varphi_2^k\chi$, where φ_2^k means φ_2 composed by itself k times.*

We now present our proof of Theorem 1.3.

Proof of Theorem 1.3. Suppose that G and H are two groups in the class $\mathscr{S}$. In order to test whether these two groups are isomorphic, we first run Procedure Decompose on G and H and obtain outputs (U_1,y_1) and (U_2,y_2) such that $(\langle U_1\rangle,\langle y_1\rangle)$ and $(\langle U_2\rangle,\langle y_2\rangle)$ are standard decompositions of G and H respectively with high probability (from Theorem 3.1). The running time of this step is polynomial in the logarithms of $|G|$ and $|H|$, from Theorem 3.1. Denote $A_1=\langle U_1\rangle$ and $A_2=\langle U_2\rangle$. The orders of A_1, A_2, y_1 and y_2 are then computed using the quantum algorithms for Tasks (i) and (ii) in Theorem 2.2. Notice that $|G|=|A_1|\cdot|y_1|$ and $|H|=|A_2|\cdot|y_2|$. If $|G|\neq|H|$, we conclude that G and H are not isomorphic. In the following, we suppose that $|G|=|H|$ and denote by n this order.

If $|y_1|\neq|y_2|$ we conclude that G and H are not isomorphic, from Proposition 6.1. Otherwise denote $|y_1|=|y_2|=m$. Then we compute a basis $(g_1,\ldots,g_s)$ of A_1 and a basis $(h_1,\ldots,h_{s'})$ of A_2 using the quantum algorithm for Task (ii) in Theorem 2.2. 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 $s=s'$ and there exists a permutation σ of $\{1,\ldots,s\}$ such that $|g_i|=|h_{\sigma(i)}|$ for each $i\in\{1,\ldots,s\}$. If $A_1\ncong A_2$ we conclude that G and H are not isomorphic, from Proposition 6.1.

Now suppose that $A_1\cong A_2\cong(\mathbb{Z}_{p_1^{f_1}})^{r_1}\times\cdots\times(\mathbb{Z}_{p_t^{f_t}})^{r_t}$, where each p_i is a prime, but $p_i^{f_i}\neq p_j^{f_j}$ for $i\neq j$. We want to decide whether the action by conjugation $\varphi_1\in Aut(A_1)$ of y_1 on A_1 and the action by conjugation $\varphi_2\in Aut(A_2)$ of y_2 on A_2 satisfy Condition (iii) in Proposition 6.1. Notice that, for each $j\in\{1,\ldots,s\}$, we can compute (in time polynomial in $\log n$) integers u_{ij} and v_{ij} such that $\varphi_1(g_j)=y_1g_jy_1^{-1}=g_1^{u_{1j}}\cdots g_s^{u_{sj}}$ and $\varphi_2(h_j)=y_2h_jy_2^{-1}=h_1^{v_{1j}}\cdots h_s^{v_{sj}}$ using the quantum algorithm for Task (iii) in Theorem 2.2.

Denote $\mathsf{V}=GL(r_1,\mathbb{Z}_{p_1})\times\cdots\times GL(r_t,\mathbb{Z}_{p_t})$. The theory developed in [19] shows that we can compute efficiently two elements M_1 and M_2 in V satisfying the following two conditions:

(a) $M_1^m=M_2^m=I$; and

(b) for each integer k, M_1 and M_2^k are conjugate in the group V if and only if there exists an isomorphism $\chi\colon A_1 \to A_2$ such that $\varphi_1 = \chi^{-1}\varphi_2^k\chi$.

If we denote $M_1 = (M_1^{(1)}, \ldots, M_1^{(t)})$ and $M_2 = (M_2^{(1)}, \ldots, M_2^{(t)})$, where each $M_1^{(\ell)}$ and each $M_2^{(\ell)}$ are matrices in $GL(r_\ell, \mathbb{Z}_{p_\ell})$, then checking if the later condition holds becomes an instance of the problem Discrete Log up to Conjugacy, and can be solved using the algorithm of Theorem 5.1 in time polynomial in t, $\log m$, and $\max_{1\leq\ell\leq t}(r_\ell + \log p_\ell)$, i.e., in time polynomial in $\log n$.

If the above instance of Discrete Log up to Conjugacy has no solution, 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, along with an element $X \in \mathsf{V}$ such that $XM_1 = M_2^k X$ (such an element is obtained from the output of the algorithm of Theorem 5.1), and compute an isomorphism χ from A_1 to A_2 such that $\varphi_1 = \chi^{-1}\varphi_2^k\chi$. The map $\mu : G \to H$ defined as $\mu(xy_1^j) = \chi(x)y_2^{kj}$ for any $x \in A_1$ and any $j \in \{0, \ldots, m-1\}$ is then an isomorphism from G to H — more details on this construction can be found in the full version of this paper [20]. ∎

Acknowledgments

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

References

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

[2] Babai, L. Trading group theory for randomness. In *Proceedings of the 17th annual ACM Symposium on Theory of Computing* (1985), pp. 421–429.

[3] Babai, L., Cooperman, G., Finkelstein, L., Luks, E. M., and Seress, Á. Fast Monte Carlo algorithms for permutation groups. *Journal of Computer and System Sciences 50*, 2 (1995), 296–308.

[4] Bacon, D., Childs, A. M., and van Dam, W. From optimal measurement to efficient quantum algorithms for the hidden subgroup problem over semidirect product groups. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science* (2005), pp. 469–478.

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

[6] Cantor, D., and Zassenhaus, H. A new algorithm for factoring polynomials over finite fields. *Mathematics of Computation 36* (1981), 587–592.

[7] Cheung, K., and Mosca, M. Decomposing finite abelian groups. *Quantum Information and Computation 1*, 3 (2001), 26–32.

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

[9] Cooperman, G., Finkelstein, L., and Linton, S. Recognizing $GL_n(2)$ in non-standard representation. In *Groups and Computation II, Proceedings of a SIMACS Workshop* (1997), pp. 85–100.

[10] Ettinger, M., and Høyer, P. On quantum algorithms for noncommutative hidden subgroups. *Advances in Applied Mathematics 25*, 3 (2000), 239–251.

[11] Friedl, K., Ivanyos, G., Magniez, F., Santha, M., and Sen, P. Hidden translation and orbit coset in quantum computing. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing* (2003), pp. 1–9.

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

[13] Inui, Y., and Le Gall, F. Efficient quantum algorithms for the hidden subgroup problem over a class of semi-direct product groups. *Quantum Information and Computation 7*, 5&6 (2007), 559–570.

[14] Ivanyos, G., Magniez, F., and Santha, M. Efficient quantum algorithms for some instances of the non-abelian hidden subgroup problem. *International Journal of Foundations of Computer Science 14*, 5 (2003), 723–740.
[15] Kannan, R., and Bachem, A. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM Journal on Computing 8*, 4 (1979), 499–507.
[16] Kavitha, T. Linear time algorithms for abelian group isomorphism and related problems. *Journal of Computer and System Sciences 73*, 6 (2007), 986–996.
[17] Kitaev, A. Y. Quantum measurements and the abelian stabilizer problem. arXiv.org e-Print archive, arXiv:quant-ph/9511026, 1995.
[18] Köbler, J., Torán, J., and Schöning, U. *The graph isomorphism problem: its structural complexity*. Birkhäuser, 1993.
[19] Le Gall, F. Efficient isomorphism testing for a class of group extensions. In *Proceedings of the 26th International Symposium on Theoretical Aspects of Computer Science* (2009), pp. 625–636. Full version available at http://arxiv.org/abs/0812.2298.
[20] Le Gall, F. An efficient quantum algorithm for some instances of the group isomorphism problem. Full version of the present paper. Available at http://arxiv.org/abs/1001.0608.
[21] Lidl, R., and Niederreiter, H. *Finite fields*. Cambridge University Press, 2008.
[22] Lipton, R. J., Snyder, L., and Zalcstein, Y. The complexity of word and isomorphism problems for finite groups. Tech. rep., John Hopkins, 1976.
[23] McKenzie, P., and Cook, S. A. The parallel complexity of abelian permutation group problems. *SIAM Journal on Computing 16*, 5 (1987), 880–909.
[24] Miller, G. On the $n^{\log n}$ isomorphism technique. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing* (1978), pp. 51–58.
[25] Moore, C., Rockmore, D. N., Russell, A., and Schulman, L. J. The power of basis selection in fourier sampling: hidden subgroup problems in affine groups. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (2004), pp. 1113–1122.
[26] Mosca, M. *Quantum Computer Algorithms*. PhD thesis, Oxford university, 1999.
[27] Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing 26*, 5 (1997), 1484–1509.
[28] Vikas, N. 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 (1996), 1–9.
[29] Watrous, J. Succinct quantum proofs for properties of finite groups. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science* (2000), pp. 537–546.
[30] Watrous, J. Quantum algorithms for solvable groups. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing* (2001), pp. 60–67.