Quantum Property Testing of Group Solvability

Yoshifumi Inui [⋆,†]   François Le Gall [†]

[⋆] *Department of Computer Science, The University of Tokyo*
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan

[†] *ERATO-SORST Quantum Computation and Information Project*
Japan Science and Technology Agency
5-28-3 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan

email: legall@qci.jst.go.jp

Abstract.   Testing efficiently whether a finite set Γ with a binary operation $\cdot$ over it, given as an oracle, is a group is a well-known open problem in the field of property testing. Recently, Friedl, Ivanyos and Santha have made a significant step in the direction of solving this problem by showing that it is possible to test efficiently whether the input $(\Gamma, \cdot)$ is an *abelian* group or is far, with respect to some distance, from any abelian group. In this paper, we make a step further and construct an efficient quantum algorithm that tests whether $(\Gamma, \cdot)$ is a *solvable* group, or is far from any solvable group. More precisely, the number of queries used by our algorithm is polylogarithmic in the size of the set Γ.

1   Introduction

In property testing, the problem considered is to decide whether an object given as an oracle has some expected property or is far from any object having that property. This is a very active research area and many properties including algebraic function properties, graph properties, computational geometry properties and regular languages were proved to be testable. We refer to, for example, [15, 19] for surveys on classical property testing. Quantum testers have also been studied [7, 11, 16], and they are known to be strictly more powerful than classical testers in some cases [7, 16].

In this paper, we focus on testing group-theoretical properties. A famous example is testing whether a function $f : G \to H$, where H and G are groups, is a homomorphism. It is well known that such a test can be done efficiently [5, 6, 21]. Another kind of problems deals with the case where the input is a finite set Γ and an oracle of a binary operation $\cdot : \Gamma \times \Gamma \to \Gamma$ over it. A classical algorithm testing associativity of the oracle $\cdot$ using $O(|\Gamma|^2)$ queries to the oracle has been constructed by Rajagopalan and Schulman [18], and Ergün et al. [8] have proposed an algorithm, using $\tilde{O}(|\Gamma|)$ queries, testing if $\cdot$ is close to the multiplication of a group. But notice that, since each element in Γ needs $\Theta(\log |\Gamma|)$ bits to be encoded, the query complexities of these algorithms can be considered as exponential in the input length when not Γ, but only $|\Gamma|$ is given (e.g., Γ is supposed to be the set of binary strings of length $\lceil \log_2 |\Gamma| \rceil$). Designing an algorithm deciding whether $(\Gamma, \cdot)$ is a group that uses a number of queries to $\cdot$ polynomial in $\log |\Gamma|$ is indeed a well-known open problem. Recently, Friedl et al. [10] have made a significant step in the direction of solving this problem by constructing a classical algorithm with query and time complexities polynomial in $\log |\Gamma|$ that tests whether $(\Gamma, \cdot)$ is an abelian group or is far from any abelian group.

In this work, we make a step further and construct an efficient quantum algorithm that tests whether $(\Gamma, \cdot)$ is a solvable group or the distance between $(\Gamma, \cdot)$ and any solvable group is at least $\epsilon |\Gamma|^2$. More precisely, our algorithm uses a number of queries polynomial in $\log |\Gamma|$ and ϵ^{-1}, and its time complexity is polynomial in $\exp((\log \log |\Gamma|)^2)$ and ϵ^{-1}, i.e., subexponential in $\log |\Gamma|$. Notice that the class of solvable groups is far much larger than the class of abelian groups and includes a vast class of non-abelian groups. To deal with those groups, we introduce new ideas relying on the

ability of quantum computation to solve fundamental group-theoretical problems, such as finding orders of elements or working with superpositions of all the elements of a subgroup.

Besides the theoretical interest of this result, our algorithm can be used when studying group-theoretical problems where the input is a black-box solvable group (i.e., given as a set a generators and an oracle performing group operations). Most known algorithms for such problems can have an unpredictable behavior when the input is not a solvable group. By applying our algorithm we can detect (in the quantum setting) if the input is far from any solvable group, and we thus obtain robust versions of the quantum algorithms already known for solvable black-box groups [9, 13, 14, 23]. We also hope that this will be useful to design new quantum property testers or group-theoretical quantum algorithms. In particular, our tester may be useful when considering quantum versions of classical algorithms solving problems over black-box solvable groups [1, 2, 3, 4] as well.

Finally, we believe that our quantum algorithm may also be a first step in the direction of designing efficient classical testers for solvable groups. Indeed, the efficient classical tester for abelian groups proposed by Friedl et al. [10] was inspired by a quantum algorithm solving the same problem. In this case, they were able to "dequantumize" the algorithm. A similar approach may be possible for our algorithm too.

2 Definitions

2.1 Distances between sets

Let Γ be a set and $\cdot : \Gamma \times \Gamma \to X$ a binary operation over it, where X is some set. We say that such couple $(\Gamma, \cdot)$ is a pseudo-magma. If $X \subseteq \Gamma$, we say that $(\Gamma, \cdot)$ is a magma. When there is no ambiguity we will denote a pseudo-magma or a magma $(\Gamma, \cdot)$ simply by Γ. We now define a distance between two pseudo-magmas. In this paper we adopt the so-called edit distance. This is the same distance as the one used by Friedl et al. [10].

Define a table of size k as a $k \times k$ matrix with entries in some arbitrary set. We consider three operations to transform a table to another. An exchange operation replaces elements in a table by arbitrary elements and its cost is the number of replaced elements. An insert operation at index i inserts a row and a column of index i. Its cost is $2k + 1$ if the original table is of size k. A delete operation at index i deletes both the row of index i and the column of index i, giving a table of size $(k-1) \times (k-1)$. Its cost is $(2k-1)$.

Let $(\Gamma, \cdot)$ be a pseudo-magma, with $\cdot : \Gamma \times \Gamma \to X$. A multiplication table for Γ is a table of size $|\Gamma|$ with entries in X for which both rows and columns are in one-to-one correspondence with elements in Γ, i.e., there exists a bijection $\sigma : \{1, \cdots, |\Gamma|\} \to \Gamma$ such that the element in the i-th row and the j-th column is $\sigma(i) \cdot \sigma(j)$. The distance between two pseudo-magmas is defined as follows.

Definition 1. *The edit distance between two tables T and T' is the minimum cost needed to transform T to T' by the above exchange, insert and delete operations. The edit distance between two pseudo-magmas Γ and Γ', denoted $d(\Gamma, \Gamma')$, is the minimum edit distance between T and T' where T (resp. T') runs over all tables corresponding to a multiplication table of Γ (resp. Γ'). For $\delta \geq 0$, we say that a pseudo-magma Γ is δ-close to another pseudo-magma Γ' if $d(\Gamma, \Gamma') \leq \delta$. Otherwise we say that Γ and Γ' are δ-far.*

Notice that if the sizes of Γ and Γ' are the same, then the edit distance becomes the minimal Hamming distance of the corresponding tables.

2.2 Property testing of group solvability

In this paper we assume that the reader is familiar with the standard notions of group theory. We refer to any standard textbook for details. For completeness, we only recall the definition of

solvable groups.

Definition 2. *A group G is solvable if there exists a collection of subgroups $G_0, \ldots, G_k$ of G such that:*

(i) for each $0 < j \leq k$, the subgroup G_{j-1} is normal in G_j and G_j/G_{j-1} is cyclic;

(ii) $\{e\} = G_0 \lhd \cdots \lhd G_k = G$.

We now give our definition of a quantum property tester of group solvability. We define such a tester as a quantum algorithm $\mathscr{A}$ receiving as input a magma $(\Gamma, \cdot)$. More precisely, the actual input of the algorithm is the value $|\Gamma|$, and two oracles are available: an oracle that generates random elements in Γ (the details of the implementation of this oracle are not essential because this oracle will only be used in a classical subprocedure), and a quantum oracle that performs the binary operation $\cdot$. Since the elements of Γ can be encoded by binary strings of length $k = \lceil \log_2 |\Gamma| \rceil$, we identify the elements with their encoding and suppose that this quantum oracle performs the map $|g\rangle|h\rangle|c\rangle \mapsto |g\rangle|h\rangle|c \oplus g \cdot h\rangle$, where g and h are elements in Γ and c is a string in $\{0, 1\}^k$. We denote by $\mathscr{A}(\Gamma)$ the behavior of the algorithm $\mathscr{A}$ on an input $(\Gamma, \cdot)$ given in this way. A more formal definition of a quantum property tester can be given but the following definition will be sufficient for our purpose.

Definition 3. *Let d be the distance defined in Subsection 2.1. A quantum ϵ-tester of group solvability is a quantum algorithm $\mathscr{A}$ such that, for any magma $(\Gamma, \cdot)$, the following holds:*

$$\begin{cases} \mathbf{Pr}[\mathscr{A}(\Gamma)\, accepts] > 2/3 & if\ d(\Gamma, \mathscr{S}) = 0 \\ \mathbf{Pr}[\mathscr{A}(\Gamma)\, rejects] > 2/3 & if\ d(\Gamma, \mathscr{S}) > \epsilon|\Gamma|^2. \end{cases}$$

Here we use $d(\Gamma, \mathscr{S})$ to represent $\inf_{G \in \mathscr{S}} d(\Gamma, G)$, where $\mathscr{S}$ denotes the set of finite solvable groups.

Notice that, a priori, requiring that the oracle is quantum may seem to give a problem different than in the classical setting, where the oracle is classical. But this is not really the case: if a classical procedure that computes the product $g \cdot h$ from g and h is available, such a quantum oracle can be effectively constructed using standard techniques of quantum computation [17].

The main result of this paper is the following theorem.

Theorem 4. *There exists a quantum ϵ-tester of group solvability that uses a number of queries polynomial in $\log |\Gamma|$ and ϵ^{-1}. The running time of this algorithm is polynomial in $\exp((\log \log |\Gamma|)^2)$ and ϵ^{-1}.*

2.3   Quantum algorithms for solvable groups

As stated in the following theorem, efficient quantum algorithms for studying the structure of solvable groups have been constructed by Watrous [23]. Our algorithm deeply relies on these algorithms.

Theorem 5. *([23]) Let G be a solvable group given as a black-box group. Then there exists a quantum algorithm running in time polynomial in $\log |G|$ that outputs, with probability at least $3/4$, $t = O(\log |G|)$ elements $h_1, \ldots, h_t$ of G and t integers $m_1, \ldots, m_t$ such that, if we denote $H_i = \langle h_1, \ldots, h_i \rangle$ for $1 \leq i \leq t$, the following holds.*

(a) $\{e\} = H_0 \lhd H_1 \lhd \cdots \lhd H_{t-1} \lhd H_t = G$; and

(b) H_i/H_{i-1} is cyclic, for $1 \leq i \leq t$, with $|H_i|/|H_{i-1}| = m_i$.

Moreover, given any $0 \leq i \leq t$, and any element g in H_i, there exists a quantum algorithm running in time polynomial in $\log |G|$ that outputs, with probability at least $3/4$, the (unique) factorization of g over H_i, i.e., integers $a_1, \ldots, a_i$ with each $a_k \in \mathbb{Z}_{m_k}$, such that $g = h_i^{a_i} h_{i-1}^{a_{i-1}} \cdots h_1^{a_1}$.

In the algorithm of Theorem 5, the group is supposed to be input as a black-box group: the input is a set of strings representing a set of generators of the group and an oracle performing the group product is available. The oracle necessary for Watrous's algorithm [23] is the map $|g\rangle|h\rangle|c\rangle \mapsto |g\rangle|h\rangle|c \oplus g \cdot h\rangle$, for any elements $g, h \in G$ and any string c in $\{0, 1\}^k$. Notice that this is the same oracle as the one given to a quantum tester of group solvability as defined in Subsection 2.2.

3 Our Quantum Algorithm

In this section we describe our quantum algorithm. We first give an overview of the algorithm in Subsection 3.1. Then, in Subsection 3.2, we explain the details. Finally, we analyse its correctness and complexity in Subsection 3.3.

3.1 Outline of our algorithm

Our algorithm consists of four parts.

Decomposition of Γ
We first construct, using Theorem 5, $t = O(\log |\Gamma|)$ elements $h_1, \ldots, h_t$ of Γ that satisfy, if Γ is a solvable group, the relations $\{e\} = H_0 \lhd H_1 = \langle h_1 \rangle \lhd \cdots \lhd H_i = \langle h_1, \cdots, h_i \rangle \lhd \cdots \lhd H_t = \langle h_1, \cdots, h_t \rangle = \Gamma$, where each H_i is a subgroup of Γ, normal in H_{i+1}, such that H_{i+1}/H_i is cyclic. If Γ is a solvable group, this decomposition gives a so-called power-conjugate presentation of Γ. If Γ is not a solvable group, these elements $h_1, \ldots, h_t$ will still define some pseudo-magmas $H_0, \ldots, H_t$, although in general these sets satisfy no group-theoretic property (in particular, they are not necessarily magmas).

Test of embedding
Then, we take sufficiently many elements of Γ and check that they are all in H_t. Success of this test implies that $|\Gamma \backslash H_t|$ is small enough. Of course, if Γ is a solvable group, then $\Gamma = H_t$ with high probability and this test always succeeds. Assume that Γ is far from any solvable group $\tilde{H}_t$. If the test succeed, since the inequality $d(\Gamma, \tilde{H}_t) \leq d(\Gamma, H_t) + d(H_t, \tilde{H}_t)$ holds for any solvable group $\tilde{H}_t$, this will imply that H_t is far from any solvable group $\tilde{H}_t$ too (because the value of $d(\Gamma, H_t)$ is basically a function of $|\Gamma \backslash H_t|$, and thus small).

Construction of the group G_t
We construct, using the information about the structure of Γ obtained at the first part of the algorithm, t solvable groups $G_1, \ldots, G_t$ and a function $\psi : G_t \to H_t$ in a way such that, if Γ is a solvable group, then ψ is a group isomorphism from G_t to H_t.

Test of homomorphism
Finally, the algorithm will test whether ψ is "almost" an homomorphism. We will show that this test is robust: if ψ is close to an homomorphism, then H_t is close to the solvable group G_t. If H_t is far from any solvable group, then this cannot hold and the homomorphism test must fail with high probability.

Again, the similar idea of constructing a group G, a function $\psi : G \to \Gamma$ and use homomorphism tests was at the heart of the property tester for abelian groups proposed by Friedl et al. [10] and inspired this work (notice that the Friedl et al. first constructed a quantum property tester for abelian groups, and then were able to remove the quantum part in their algorithm). However there are new difficulties that arise when considering property testers for solvable groups. The first one is

that analyzing the decomposition the H_i's is more difficult and the power of quantum computation seems necessary to perform this task efficiently. The second complication is that, now, the groups G_i's we are considering are solvable, i.e., in general not commutative. In this case, we have to be very careful in the definition of G_i and additional tests have to be done to ensure that the G_i's we define are really groups.

3.2 Algorithm

Our algorithm appears in Figure 1 and each of the four parts are explained in details in Subsections 3.2.1 to 3.2.4. If all the tests performed succeed, we decide that Γ is a solvable group. Otherwise we decide that Γ is $(\epsilon|\Gamma|^2)$-far from any solvable group.

PART I: Decomposition of Γ
1. Take $O(\log|\Gamma|)$ random elements uniformly and independently in Γ.
2. Use the first algorithm of Theorem 5 on them and obtain the set $\{h_1, \ldots, h_t\}$ and integers $m_1, \ldots, m_t$.
3. For each $i \in \{1, \ldots, t\}$, use Shor's order finding algorithm on h_i and obtain some integer n_i.
4. Compute the decompositions of all $h_i^{m_i}$ and $h_i^{n_i-1} \cdot (h_k \cdot h_i)$ over H_{i-1}, for $i \in \{1, \ldots, t\}$
   and $k \in \{1, \ldots, i-1\}$, and check the obtained decompositions.

PART II: Test of embedding
5. Check that $|\Gamma| = m_1 \times \cdots \times m_t$ and $|\Gamma \backslash H_t|/|\Gamma| < \epsilon/4$.

PART III: Construction of the group G_t
6. For j from 2 to t check that Conditions (a), (b) and (c) of Proposition 7 hold.

PART IV: Test of homomorphism
7. Check that $\mathbf{Pr}_{x,y \in G_t}[\psi(x \circ y) = \psi(x) \cdot \psi(y)] > 1 - \eta$ with $\eta = \epsilon/422$.

Figure 1: Quantum ϵ-tester of group solvability

3.2.1 Decomposition of Γ

The first step in our algorithm finds a power-conjugate representation of Γ when Γ is a solvable group. We will prove that when Γ is far from any solvable group, then the output of this step cannot be a power-conjugate representation of a group close to Γ and that this can be detected by our algorithm at part II, III or IV.

We begin by picking $s = \Theta(\log|\Gamma|)$ random elements $\alpha_1, \cdots, \alpha_s$ uniformly and independently from the ground set Γ. For simplicity, we first suppose that Γ is a solvable group, and then discuss the general case.

Case where Γ is a solvable group. Denote $\Gamma' = \langle \alpha_1, \cdots, \alpha_s \rangle$. Then, with high probability, $\Gamma = \Gamma'$. Here we rely on the standard fact in computational group theory that, for any group K, $\Theta(\log|K|)$ random elements taken uniformly in K constitute, with high probability, a generating set of K. We now run the first algorithm of Theorem 5 with input Γ' presented as a black-box group as follows: $\alpha_1, \cdots, \alpha_s$ is the set of generators and the operation $\cdot$ is the oracle performing group multiplication. The output of the algorithm is then, with high probability, a set of t elements $h_1, \ldots, h_t$ of Γ and t integers $m_1, \ldots, m_t$ such that, if we denote $H_i = \langle h_1, \ldots, h_i \rangle$ for $1 \leq i \leq t$, the following holds:

(a) $\{e\} = H_0 \lhd H_1 \lhd \cdots \lhd H_{t-1} \lhd H_t = \Gamma'$; and

(b) H_i/H_{i-1} is cyclic for $1 \le i \le t$ and satisfies $|H_i|/|H_{i-1}| = m_i$.

We then use Shor's quantum algorithm [20] to compute the order n_i of each h_i in Γ. Moreover, we further analyze the structure of Γ' and use the second algorithm of Theorem 5 to decompose the elements $h_i^{m_i}$ and $h_i^{n_i-1} \cdot (h_k \cdot h_i)$ over H_{i-1}, for each $i \in \{2, \ldots, t\}$ and each $k \in \{1, \ldots, i-1\}$. Notice that, indeed, each $h_i^{m_i}$ and $h_i^{n_i-1} \cdot (h_k \cdot h_i) = h_i^{-1} \cdot h_k \cdot h_i$ are in H_{i-1} when Γ is a solvable group. We denote the decompositions obtained by

$$h_i^{m_i} = h_{i-1}^{r_{i-1}^{(i)}} \cdot \left(\cdots \left(h_3^{r_3^{(i)}} \cdot \left(h_2^{r_2^{(i)}} \cdot h_1^{r_1^{(i)}} \right) \right) \right) \quad \text{for } 2 \le i \le t, \tag{1}$$

$$h_i^{n_i-1} \cdot (h_k \cdot h_i) = h_{i-1}^{s_{k,i-1}^{(i)}} \cdot \left(\cdots \left(h_3^{s_{k,3}^{(i)}} \cdot \left(h_2^{s_{k,2}^{(i)}} \cdot h_1^{s_{k,1}^{(i)}} \right) \right) \right) \quad \text{for } 1 \le k < i \le t, \tag{2}$$

where each $r_\ell^{(i)}$ and each $s_{k,\ell}^{(i)}$ are in $\mathbb{Z}_{m_\ell}$. (The parentheses are superfluous when $\cdot$ is associative, but not in the general case we discuss below.)

General Case. In general, we do not know whether Γ is a solvable group or not but we do exactly the same as above: we first run the first algorithm of Theorem 5 on the set $\{\alpha_1, \cdots, \alpha_s\}$ with the oracle $\cdot$. If this algorithm errs, we conclude that Γ is not a solvable group (this decision is correct with high probability because, if Γ is a solvable group, then the algorithm of Theorem 5 succeeds with high probability). Now suppose that we have obtained elements $h_1, \ldots, h_t$ and a set of integers $m_1, \ldots, m_t$. We define the following sets by recurrence: $H_1 = \{h_1^a | a \in \mathbb{Z}_{m_1}\}$, and, for $2 \le j \le t$, $H_j = \{h_j^a \cdot h | a \in \mathbb{Z}_{m_j}, h \in H_{j-1}\}$. Here, and in many other places in this paper, we use the notation h^r, for $h \in \Gamma$ and $r \ge 1$, to denote the product $h \cdot (\cdots (h \cdot (h \cdot h)))$, since $\cdot$ is not in general associative. Moreover we use the convention $h^0 = h_1^{m_1}$ for any $h \in \Gamma$. Notice that the value of h^r can be computed using $O(\log r)$ queries to the oracle $\cdot$ using repeated squaring methods.

Notice that, in general, the pseudo-magmas H_i's have no group-theoretical structure at all (in particular they may not be magmas). We then use Shor's order finding algorithm [20] on each h_i and obtain some integer n_i. Then we run the second algorithm of Theorem 5 to decompose the elements $h_i^{m_i}$ and $h_i^{n_i-1} \cdot (h_k \cdot h_i)$ over H_{i-1}, for each $i \in \{2, \ldots, t\}$ and each $k \in \{1, \ldots, i-1\}$. If the algorithm errs or outputs something irrelevant, we conclude that Γ is not a solvable group. Suppose that the algorithm succeeds and outputs decompositions. We use the notations of Equations (1) and (2) to denote the decompositions obtained. We check whether these decompositions are correct, i.e., we compute the right sides of Equations (1) and (2) and check that they match the left sides. If they are correct, we move to the next step (Subsection 3.2.2). Otherwise, we conclude that Γ is not a solvable group.

3.2.2 Test of embedding

In the second part of our algorithm, we first check that $|\Gamma| = m_1 \times \cdots \times m_t$. Then, we want to check whether $|\Gamma \backslash H_t|$ is small enough. Otherwise we conclude that Γ is not a solvable group. Indeed, if Γ is a group, then with high probability (on the choice of $\alpha_1, \ldots, \alpha_s$ and on the randomness of the algorithm of Theorem 5) $\Gamma = H_t$.

More precisely we check whether $|\Gamma \backslash H_t|/|\Gamma| < \epsilon/4$ holds. In order to perform this test, we simply take c_1 elements of Γ and check whether they are all in H_t (by using the second algorithm of Theorem 5 and checking the obtained decompositions). It is easy to show that, when taking $c_1 = \Theta(\epsilon^{-1})$, we can detect whether $|\Gamma \backslash H_t|/|\Gamma| > \epsilon/4$ with constant probability.

6

3.2.3   Construction of the group G_t

We now show how to construct an abstract group G_t defined by the power-conjugate presentation found in Part I of our algorithm (Equations (1) and (2)) when such a group exists, i.e., when the presentation is consistent with the definition of a group.

We first define by recurrence the family of magmas $\{G_j\}_{1 \leq j \leq t}$, where each G_j is equal (as a set) to $\mathbb{Z}_{m_j} \times \cdots \times \mathbb{Z}_{m_1}$. G_1 is defined as the cyclic group $(\mathbb{Z}_{m_1}, +)$, where $+$ is the addition modulo m_1. For any $i \in \{2, \ldots, t\}$, denote by u_i the element $(r_{i-1}^{(i)}, \ldots, r_1^{(i)})$ of G_{i-1} and, for any $i \in \{2, \ldots, t\}$ and $k \in \{1, \ldots, i-1\}$, denote by $v_{i,k}$ the element $(s_{k,i-1}^{(i)}, \ldots, s_{k,1}^{(i)})$ of G_{i-1}.

Definition 6. *Define $G_1 = (\mathbb{Z}_{m_1}, +)$ and, for $2 \leq j \leq t$, let G_j be the magma $(\mathbb{Z}_{m_j} \times G_{j-1}, \circ_j)$ with*

$$
(a, x) \circ_j (b, y) = \begin{cases} \left(a + b, \phi_j^{(b)}(x) \circ_{j-1} y \right) & \text{if } a + b < m_j \\ \left(a + b - m_j, u_j \circ_{j-1} \phi_j^{(b)}(x) \circ_{j-1} y \right) & \text{if } a + b \geq m_j \end{cases}
$$

where $\phi_j : G_{j-1} \to G_{j-1}$ maps any element $(a_{j-1}, \cdots, a_1)$ of G_{j-1} to the element $\phi_j((a_{j-1}, \cdots, a_1)) = v_{j,j-1}^{a_{j-1}} \circ_{j-1} \left(\cdots \circ_{j-1} \left(v_{j,2}^{a_2} \circ_{j-1} v_{j,1}^{a_1} \right) \right)$ of G_{j-1}, and $\phi_j^{(b)}$ means ϕ_j composed by itself b times.

We will usually denote $\circ_j$ or $\circ_{j-1}$ simply by $\circ$ when there is no ambiguity.

In order to illustrate this definition, let us consider the case where all the H_j's are solvable groups. In this case, each $H_j = \{h_j^{a_j} \cdot \cdots \cdot h_1^{a_1} \mid a_j \in \mathbb{Z}_{m_j}\}$ is in bijection with $\mathbb{Z}_{m_j} \times \cdots \times \mathbb{Z}_{m_1}$ (as a set). Fix a j and consider H_j. Each element $h_j^{a_j} \cdots h_1^{a_1}$ is associated with the element $(a_j, \ldots, a_1)$ of G_j. Now the element $\phi_j((a_{j-1}, \cdots, a_1))$ corresponds to the element

$$
h_j^{-1} \cdot (h_{j-1}^{a_{j-1}} \cdots h_1^{a_1}) \cdot h_j = \left(h_{j-1}^{s_{j-1,j-1}^{(j)}} \cdots h_1^{s_{j-1,1}^{(j)}} \right)^{a_{j-1}} \cdots \left(h_{j-1}^{s_{1,j-1}^{(j)}} \cdots h_1^{s_{1,1}^{(j)}} \right)^{a_1}.
$$

In other words, the map ϕ_j in G_{j-1} corresponds to the automorphism $h \mapsto h_j^{-1} h h_j$ of H_j. For any two elements g and g' in H_{j-1}, since $h_j^a \cdot g \cdot h_j^b \cdot g' = h_j^{a+b} \cdot (h_j^{-b} \cdot g \cdot h_j^b) \cdot g'$ we see that the G_j's are defined to be isomorphic to the H_j's in the case where the H_j's are solvable groups.

If the H_j's are not groups, then the G_j's constructed in Definition 6 are not necessarily groups. But we now show that when some additional conditions are satisfied, the G_j's become groups. In technical words these are necessary and sufficient conditions to make the presentation of G_j a consistent presentation of successive cyclic extensions. In the next proposition, we denote by $x_{j,k}$, for $1 \leq k \leq j \leq t$, the element of G_j with one 1 at the index k (from the right) and zeros at all the other indexes.

Proposition 7. *Let $1 < j < t$. Suppose that G_{j-1} is a solvable group and, if $j \geq 3$, suppose additionally that G_{j-2} is a solvable group and ϕ_{j-1} is a group automorphism of G_{j-2}. Assume that the following three conditions hold.*

(a) $x_{j-1,k} \circ v_{j-1,j-1} = v_{j-1,j-1} \circ v_{j-1,k}$ for all $1 \leq k < j-1$; and

(b) $\phi_j(u_j) = u_j$; and

(c) $\phi_j^{(m_j)}(x_{j-1,i}) = u_j^{-1} \circ x_{j-1,i} \circ u_j$ for all $1 \leq i \leq j-1$.

Then G_j is a solvable group and ϕ_j is a group automorphism of G_{j-1}.

7

Proof. If ϕ_j is an automorphism of G_{j-1}, then Conditions (b) and (c) imply that G_j, as defined in Definition 6, is a so-called cyclic extension of G_{j-1} and thus a solvable group (see for example [22, Section 9.8]). We will show below that Condition (a) implies that ϕ_j is an endomorphism of G_{j-1}. Since $\phi_j^{(m_j)}$ is an automorphism of G_{j-1} from Condition (c), ϕ_j is thus an automorphism too.

We now prove that ϕ_j is an endomorphism of G_{j-1}. If $j = 2$, then this is obviously the case: ϕ_2 is the endomorphism of $G_1 = (\mathbb{Z}_{m_1}, +)$ mapping a to $av_{11}^{(2)}$. In the following we suppose that $j \geq 3$. We first start with a few useful observations. First notice that, for any a and b in $\mathbb{Z}_{m_{j-1}}$, the equality $\phi_j((a + b, e)) = \phi_j((a, e)) \circ \phi_j((b, e))$, where e denotes the unity element of G_{j-2}, holds from the definition of ϕ_j. Also notice that, for any a in $\mathbb{Z}_{m_{j-1}}$ and any x in G_{j-2}, the equality $\phi_j((a, x)) = \phi_j((a, e)) \circ \phi_{j-1}(x)$ holds.

Any element $z \in G_{j-2}$ can be written in the form $z = x_{j-1,j-2}^{\alpha_{j-2}} \cdots x_{j-1,1}^{\alpha_1}$ for some integers $\alpha_1, \ldots, \alpha_{j-2}$. Condition (a) then implies that the equality

$$z \circ v_{j-1,j-1} = v_{j-1,j-1} \circ v_{j-1,j-2}^{\alpha_{j-2}} \circ \cdots \circ v_{j-1,1}^{\alpha_1} = v_{j-1,j-1} \circ \phi_{j-1}(z)$$

holds (since ϕ_{j-1} is an endomorphism of G_{j-2} and $\phi_{j-1}(x_{j-1,k}) = v_{j-1,k}$ for any $1 \leq k < j - 1$). More generally, for any $b \in \mathbb{Z}_{m_{j-1}}$ and any $z \in G_{j-2}$, we have

$$z \circ \phi_j((b, e)) = z \circ v_{j-1,j-1}^{b} = v_{j-1,j-1}^{b} \circ \phi_{j-1}^{(b)}(z) = \phi_j((b, e)) \circ \phi_{j-1}^{(b)}(z).$$

Let a, b be two elements of $\mathbb{Z}_{m_{j-1}}$ and x, y be two elements of G_{j-2}. Putting together the above observations we can write

$$
\begin{aligned}
\phi_j((a, x)) \circ \phi_j((b, y)) &= \phi_j((a, e)) \circ \phi_{j-1}(x) \circ \phi_j((b, e)) \circ \phi_{j-1}(y) \\
&= \phi_j((a, e)) \circ \phi_j((b, e)) \circ \phi_{j-1}^{(b+1)}(x) \circ \phi_{j-1}(y) \\
&= \phi_j((a, e)) \circ \phi_j((b, e)) \circ \phi_{j-1}(\phi_{j-1}^{(b)}(x) \circ y) \\
&= \phi_j((a, e)) \circ \phi_j((b, \phi_{j-1}^{(b)}(x) \circ y)) \\
&= \phi_j((a + b, v \circ \phi_{j-1}^{(b)}(x) \circ y)),
\end{aligned}
$$

where $v = u_j$ if $a + b \geq m_j$ and $v = e$ otherwise. We conclude that

$$\phi_j((a, x)) \circ \phi_j((b, y)) = \phi_j((a, x) \circ (b, y)),$$

and thus ϕ_j is an endomorphism of G_{j-1}. $\square$

To illustrate the three conditions of Proposition 7, let us again consider the case where $(\Gamma, \cdot)$ is a group. Then conditions (b) and (c) hold due to the facts that u_j in G_{j-1} corresponds to the element $h_j^{m_j}$ and that ϕ_j corresponds to the automorphism $h \mapsto h_j^{-1} h h_j$ of H_{j-1}. Condition (a) follows from Equation (2).

For each $j \in \{2, \ldots, t\}$, testing that Conditions (a) and (b) hold can be done using a number of multiplications in the group G_{j-1} polynomial in $\log |\Gamma|$. The best known classical algorithm for computing products in a solvable group given as a power-conjugate presentation is an algorithm by Höfling [12] with time complexity $O(\exp((\log \log |G_{j-1}|)^2)) = O(\exp((\log \log |\Gamma|)^2))$. Notice that if Condition (a) holds then ϕ_j is a homomorphism. Then each term $\phi_j^{(m_j)}(x_{j-1,i})$ in Condition (c) can be computed using a number of group products polynomial in $\log |\Gamma|$ by computing, step by step by increasing ℓ from 0 to $\lfloor \log m_j \rfloor$, the values $\phi_j^{(2^\ell)}(x_{j-1,k})$ for all $1 \leq k \leq j - 1$. The total time complexity of checking that all the G_i's are solvable groups is thus $O(\exp((\log \log |\Gamma|)^2))$. No query to the oracle $\cdot$ is needed.

3.2.4 Test of homomorphism

We now suppose that the G_i's have passed all the tests of Proposition 7 and thus G_t is a solvable group. Let ψ be the surjective map from G_t to H_t defined as

$$\psi(a_t, a_{t-1}, \cdots, a_1) = h_t^{a_t} \cdot (h_{t-1}^{a_{t-1}} \cdot (\cdots \cdot (h_2^{a_2} \cdot h_1^{a_1}))).$$

We will test whether ψ is a homomorphism from G_t to H_t. If $(\Gamma, \cdot)$ is a solvable group, then ψ is an homomorphism by construction. We now show that this test is robust.

Proposition 8. *Let η be a constant such that $0 < \eta < 1/120$. Assume that $|H_t| > 3|G_t|/4$. Suppose that*

$$\mathbf{Pr}_{x,y \in G_t}[\psi(x \circ y) = \psi(x) \cdot \psi(y)] > 1 - \eta. \tag{3}$$

Then there exists a solvable group $\tilde{H}_t$ that is $(211\eta|\Gamma|^2)$-close to H_t.

Proof. From Condition (3), Theorem 2 of [10] implies that there exists a group $(\tilde{H}_t, *)$ with $|\tilde{H}_t| \leq |G_t|$, and a homomorphism $\tilde{\psi} : G_t \to \tilde{H}_t$ such that:

(a) $|\tilde{H}_t \backslash H_t| \leq 30\eta|\tilde{H}_t|$;

(b) $\mathbf{Pr}_{h,h' \in \tilde{H}_t}[h * h' \neq h \cdot h'] \leq 91\eta$; and

(c) $\mathbf{Pr}_{x \in G_t}[\tilde{\psi}(x) \neq \psi(x)] \leq 30\eta$.

Notice that, strictly speaking, Theorem 2 of [10] is stated only in the case where H_t is a magma, i.e., closed under $\cdot$. This is not the case here because H_t may not be a magma, but only a pseudo-magma. However, careful inspection of the proof of Theorem 2 of [10] shows that exactly the same result holds when H_t is a pseudo-magma too. The distance between $\tilde{H}_t$ and H_t is determined by the number of elements being a member of either set and the number of pairs of two elements for which the result of the multiplication differ. In particular, this distance has for upper bound the cost of the following transform: starting from the table of $\tilde{H}_j$, we first delete rows and columns corresponding to elements in $\tilde{H}_t \backslash H_t$, insert rows and columns corresponding to elements in $H_t \backslash \tilde{H}_t$, and then exchange multiplication entries which differ between two tables. It follows from (a) and (b) that the number of elements in $\tilde{H}_t \backslash H_t$ is less than $30\eta|\tilde{H}_t|$ and the number of pairs $(h, h') \in \tilde{H}_t \times \tilde{H}_t$ such that $h * h' \neq h \cdot h'$ is less than $91\eta|\tilde{H}_t|^2$. It remains to show that $H_t \backslash \tilde{H}_t$ is small enough too and that $\tilde{H}_t$ is a solvable group.

Suppose towards a contradiction that $|\tilde{\psi}(G_t)| < |G_t|$. Then $|\tilde{\psi}(G_t)| \leq |G_t|/2$. From Condition (c), we obtain $|H_t| = |\psi(G_t)| \leq |G_t|/2 + 30\eta|G_t| \leq 3|G_t|/4$. This gives a contradiction. Thus $|\tilde{\psi}(G_t)| = |\tilde{H}_t| = |G_t|$ and $\tilde{\psi}$ is an isomorphism from G_t to $\tilde{H}_t$. Since G_t is a solvable group, $\tilde{H}_t$ is solvable too. Since $|H_t| \leq |G_t|$, it also follows that $|H_t| \leq |\tilde{H}_t|$ and thus $|H_t \backslash \tilde{H}_t| \leq |\tilde{H}_t \backslash H_t| \leq 30\eta|\tilde{H}_t|$.

Deleting $|\tilde{H}_t \backslash H_t|$ rows and column from the table of $\tilde{H}_t$ costs

$$2|\tilde{H}_t||\tilde{H}_t \backslash H_t| - |\tilde{H}_t \backslash H_t|^2 \leq 60\eta|\tilde{H}_t|^2.$$

Then inserting $|H_t \backslash \tilde{H}_t|$ rows and columns similarly costs at most $60\eta|\tilde{H}_t|^2$ too. Thus the distance between H_t and the solvable group $\tilde{H}_t$ is at most $[(60 + 60 + 91)\eta|\tilde{H}_t|^2] \leq 211\eta|\Gamma|^2$. $\qquad\square$

More precisely, we perform the following test. We want to test which of $\mathbf{Pr}_{x,y \in G}[\psi(x \circ y) = \psi(x) \cdot \psi(y)] = 1$ and $\mathbf{Pr}_{x,y \in G_t}[\psi(x \circ y) = \psi(x) \cdot \psi(y)] \leq 1 - \eta$ with $\eta = \epsilon/422$ holds. We take c_2 pairs (x, y) of elements of G_t and test whether they all satisfy $\psi(x \circ y) = \psi(x) \cdot \psi(y)$. It is easy to show that, when taking $c_2 = \Theta(\eta^{-1}) = \Theta(\epsilon^{-1})$, we can decide which case holds with constant probability.

3.3 Correctness and complexity

We now evaluate the performance of our algorithm. This gives the result of Theorem 4.

First, suppose that the magma $(\Gamma, \cdot)$ is a solvable group. With high probability the set of elements taken at step 1 of the algorithm of Figure 1 is a generating set of Γ and the first algorithm of Theorem 5 succeeds on this set. In this case, each of the tests realized at steps 3 to 5 succeeds with high probability (since the success probability of Shor's algorithm and of the second algorithm of Theorem 5 can be amplified), and then all the tests at steps 6 and 7 succeed with probability 1. Thus the global error probability is constant.

Now, we would like to show that any magma Γ that is $(\epsilon|\Gamma|^2)$-far from any solvable group is rejected with high probability. Take such a magma Γ. Then H_t is $(\frac{\epsilon}{2}|\Gamma|^2)$-far from any solvable group $\tilde{H}_t$ or $|\Gamma \backslash H_t|/|\Gamma| > \epsilon/4$. This assertion holds because for any solvable group $\tilde{H}_t$, the inequalities $\epsilon|\Gamma|^2 < d(\Gamma, \tilde{H}_t) \leq d(\Gamma, H_t) + d(H_t, \tilde{H}_t)$ hold and $d(\Gamma, H_t) = 2|\Gamma \backslash H_t||\Gamma| - |\Gamma \backslash H_t|^2 \leq 2|\Gamma \backslash H_t||\Gamma|$ since $H_t \subseteq \Gamma$ and the operation is the same. If the latter holds, it should be rejected with high probability at test 5. Now suppose that the former holds and that all the steps 1–6 succeed. Then with high probability $|H_t| \geq (1 - \epsilon/4)|\Gamma| \geq 3|\Gamma|/4 = 3|G_t|/4$. From Proposition 8 this implies that $\mathbf{Pr}_{x,y \in G_t}[\psi(x \circ y) = \psi(x) \cdot \psi(y)] \leq 1 - \epsilon/422$. This is detected with high probability at step 7.

The algorithm queries the oracle Γ a number of times polynomial in $\log|\Gamma|$ at each of the steps 1 to 4, and a number of times polynomial in $\log|\Gamma|$ and ϵ^{-1} at steps 5 and 7. Additional computational work is needed at steps 6 and 7 to compute a polynomial number of products in the groups G_i's. Since each product can be done (without queries) using $O(\exp((\log\log|G_i|)^2)) = O(\exp((\log\log|\Gamma|)^2))$ time using the algorithm by Höfling [12], the total time complexity of the algorithm is polynomial in $\exp((\log\log|\Gamma|)^2)$ and ϵ^{-1}.

Acknowledgments

References

[1] V. Arvind and N. V. Vinodchandran, *Solvable black-box group problems are low for PP*, Theoretical Computer Science, 180(1-2), pp. 17–45, 1997.

[2] L. Babai and R. Beals, *Las Vegas algorithms for matrix groups*, Proceedings of the 34th Annual Symposium on Foundations of Computer Science, pp. 427–436, 1993.

[3] L. Babai, G. Cooperman, L. Finkelstein, E. Luks and Á. Seress, *Fast Monte Carlo algorithms for permutation groups*, Journal of Computer and System Sciences, 50(2), pp. 296–307, 1995.

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

[5] M. Ben-Or, D. Coppersmith, M. Luby and R. Rubinfeld, *Non-Abelian homomorphism testing, and distributions close to their self-convolutions*, Proceedings of the 8th International Workshop on Randomization and Computation, pp. 273–285, 2004.

[6] M. Blum, M. Luby and R. Rubinfeld, *Self-testing/correcting with applications to numerical problems*, Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, pp. 73–83, 1990.

[7] H. Buhrman, L. Fortnow, I. Newman and H. Röhrig, *Quantum property testing*, Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 873–882, 2001.

[8] F. Ergün, S. Kannan, R. Kumar, R. Rubinfeld and M. Viswanathan, *Spot-checkers*, Journal of Computer and System Sciences, 60(3), pp. 717–751, 2000.

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

[10] K. Friedl, G. Ivanyos and M. Santha, *Efficient testing of groups*, Proceedings of the 37th Annual ACM Symposium on Theory of Computing, pp. 157–166, 2005.

[11] K. Friedl, F. Magniez, M. Santha and P. Sen, *Quantum testers for hidden group properties*, Proceedings of the 28th International Symposium on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science, 2747, pp. 419–428, 2003.

[12] B. Höfling, *Efficient multiplication algorithms for finite polycyclic groups*, preprint, available at http://www-public.tu-bs.de:8080/~bhoeflin/, 2004.

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

[14] G. Ivanyos, F. Magniez and M. Santha, *Efficient quantum algorithms for some instances of the non-Abelian hidden subgroup problem*, International Journal of Foundations of Computer Science, 14(5), pp. 723–740, 2003.

[15] M. Kiwi, F.Magniez and M. Santha, *Exact and approximate testing/correcting of algebraic functions: a survey*, Proceedings of the 1st Summer School on Theoretical Computer Science, Lecture Notes in Computer Science, 2292, pp. 30–83, 2000.

[16] F. Magniez and A. Nayak, *Quantum complexity of testing group commutativity*, Proceedings of the 32nd International Colloquium on Automata, Languages and Programming, Lecture Notes in Computer Science, 3580, pp.1312–1324, 2005.

[17] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*, Cambridge University Press, 2000.

[18] S. Rajagopalan and L. Schulman, *Verification of identities*, Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science, pp. 612–616, 1996.

[19] D. Ron, *Property testing*, In Handbook of Randomized Computing, Kluwer Academic Publishers, pp. 597–649, 2001.

[20] P. W. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM Journal on Computing, 26(5), pp. 1484–1509, 1997.

[21] A. Shpilka and A. Wigderson, *Derandomizing homomorphism testing in general groups*, Proceedings of the 36th Annual ACM Symposium on Theory of Computing, pp. 427–435, 2004.

[22] C. Sims, *Computation with Finitely Presented Groups*, Cambridge University Press, 1994.

[23] J. Watrous, *Quantum algorithms for solvable groups*, Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, pp. 60–67, 2001.