LMS J. Comput. Math. 15 (2012) 38–43 © 2012 Author
doi:10.1112/S1461157012000046

Testing commutativity of a group and the power of randomization

Igor Pak

Abstract

Let G be a group generated by k elements, $G = \langle g_1, \ldots, g_k \rangle$, with group operations (multiplication, inversion and comparison with identity) performed by a black box. We prove that one can test whether the group G is abelian at a cost of $O(k)$ group operations. On the other hand, we show that a deterministic approach requires $\Omega(k^2)$ group operations.

Introduction

Let G be a finite *black box group* (see for example [3]), defined as follows. The elements are given as binary strings of a fixed length, say N, and a '*black box*', also called an *oracle*, performs group operations: multiplication, inversion and the recognition of the identity element. The group G is assumed to be given by a generating set $S = \{g_1, \ldots, g_k\}$, $\langle S \rangle = G$.

Examples of black box groups include groups defined as subgroups of certain large groups H. In this case G is generated by a set $S \subset H$. When H is isomorphic to S_n, these are called *permutation groups* and, when $H \simeq \mathrm{GL}(n, \mathbb{F}_q)$, these are called *matrix groups*. Permutation groups so far remain the best understood class of these, with the most efficient algorithms available. These algorithms were built on the fundamental algorithms of Sims [24] that made it possible to determine group membership and group order. Matrix groups are often difficult to work with, and many algorithms are defined in a generality of black box groups (see [3]). We refer to [12, 22, 27] for extensive overviews of various aspects of recognition and property testing in black box, permutation and matrix groups.

In this note we resolve the basic problem of testing whether a black box group is abelian. A simple deterministic algorithm, consisting of checking whether $[g_i, g_j] = \mathrm{id}$ for all $i \neq j$, is shown to be optimal up to a constant. On the other hand, in a probabilistic setting when a small probability of error is allowed, there exists a randomized algorithm which tests commutativity of a group at a cost of $O(k)$ group multiplications. This is another example of what we call the power of randomization.

As an application of our technique, we show that testing commutativity of matrix groups can be done in time linear in the size of the input. This is done by combining our main routine and the Freivalds approach [9]. Finally, we speculate that a solvability test can be performed in nearly linear time.

This is perhaps the first time when commutativity testing was asked in a generality of black box groups. The reason, probably, is the simple $O(k^2)$ deterministic algorithm described above, which is fast enough for most practical and theoretical needs (k tends to be small in practice).

We should mention that various other properties (whether G is nilpotent, solvable, etc.) have been studied in various settings (see [4, 6, 7, 14]). Our randomized algorithm, when used as a subroutine, easily improves complexity of some of these advanced algorithms.

Before we finish, let us clarify an important difference between our work and the work on testing properties of relations, on group isomorphism, etc (see for example [5, 13, 15, 20, 26, 28]). In that case the input is a multiplication table, of size $O(|G|^2)$, while in our case we usually assume that $k = O(\log |G|)$.

Received 19 July 2011; revised 11 September 2011.

2010 Mathematics subject classification 20F69, 20PXX, 68Q87 (primary).

https://doi.org/10.1112/S1461157012000046 Published online by Cambridge University Press

1. *Randomized algorithm*

Let $G = \langle g_1, \ldots, g_k \rangle$ be a black box group. Define *random subproducts*

$$h = g_1^{\epsilon_1} \cdot \ \ldots \ \cdot g_k^{\epsilon_k},$$

where $\epsilon_i \in \{0, 1\}$ are determined by independent flips of a fair coin. Now define the algorithm to consist of a constant number of checks whether two independently chosen random subproducts commute with each other. In black box notation, we will be testing whether $[h, h'] = \mathrm{id}$. We need the following result.

LEMMA 1.1. *Let G be non-abelian and let h, h' be two independently chosen random subproducts. Then*

$$\mathbf{P}([h, h'] \neq \mathrm{id}) \geqslant \tfrac{1}{4}.$$

We need the following terminology. A probabilistic algorithm is called *one-sided Monte Carlo* if for any $\varepsilon > 0$ in the input, when it outputs *Yes*, then the right answer is *Yes* with probability $>1 - \varepsilon$; but when it outputs *No*, then the right answer is indeed *No*, that is, with probability 1. We call ε the *probability of error*.

Everywhere below we denote by μ the cost of group operations, and by ν the cost of identity recognition.

THEOREM 1.2. *Let G be a black box group generated by k elements. There exists a one-sided Monte Carlo algorithm for testing whether the group is abelian, at a cost $O(k \log(1/\varepsilon)(\mu + \nu))$, where $\varepsilon > 0$ is the probability of error.*

Observe that Lemma 1.1 immediately implies Theorem 1.2. Before we prove the crucial Lemma 1.1, let us first prove the following preliminary result.

LEMMA 1.3. *Let $G = \langle g_1, \ldots, g_k \rangle$, $H \subsetneq G$ and let h be a random subproduct. Then* $\mathbf{P}(h \notin H) \geqslant \frac{1}{2}$.

Proof of Lemma 1.3. Let $i \in \{1, \ldots, k\}$ be the smallest number such that $g_i \notin H$. Write the subproduct h as follows:

$$h = \left(g_1^{\epsilon_1} \cdot \ \ldots \ \cdot g_{i-1}^{\epsilon_{i-1}}\right) \cdot g_i^{\epsilon_i} \cdot (g_{i+1}^{\epsilon_{i+1}} \cdot \ \ldots \ \cdot g_k^{\epsilon_k}) = u \cdot g_i^{\epsilon_i} \cdot v.$$

Note that $u \in H$. When $v \in H$, with probability $1/2$ we have $\epsilon_i = 1$, and $h = u \cdot g_i \cdot v \notin H$. Similarly, when $v \notin H$, with probability $1/2$ we have $\epsilon_i = 0$, and $h = u \cdot v \notin H$. This completes the proof. □

Proof of Lemma 1.1. Let G be non-abelian. Then the center $H = C(G)$ is a proper subgroup of G. By Lemma 1.3, $\mathbf{P}(h \notin H) \geqslant 1/2$.

Now assume $h \notin H = C(G)$. Let H' be the centralizer $C(h)$ of h. Then H' is also a proper subgroup of G and $\mathbf{P}(h' \notin H') \geqslant 1/2$. But this is equivalent to $[h, h'] \neq \mathrm{id}$. Therefore, the probability that $[h, h'] \neq \mathrm{id}$ is $\geqslant 1/2 \cdot 1/2 = 1/4$. □

EXAMPLE 1.4. Theorem 1.2 implies that for checking whether G is abelian, $O(k)$ group multiplications suffice. While multiplication of permutations is linear in the size of the input, the matrix multiplication is quite costly. Luckily, there is a way to avoid some of this cost.

THEOREM 1.5. *Let $G \subset \mathrm{GL}(n, \mathbb{F}_q)$ be a group of $n \times n$ matrices over the finite field $\mathbb{F}_q$, given by k generators. Denote by η the cost of addition and multiplication in $\mathbb{F}_q$. There exists a one-sided Monte Carlo algorithm for testing whether G is abelian, at a cost $O(n^2 k \log(1/\varepsilon)\, \eta)$, where $\varepsilon > 0$ is the probability of error.*

https://doi.org/10.1112/S1461157012000046 Published online by Cambridge University Press

Proof. Suppose we need to check whether a matrix $M = A \cdot B \cdot A^{-1} \cdot B^{-1}$ is Id. Rather than multiply (and invert) matrices, this can be done by checking whether $M\,v = v$ for random $v \in {\mathbb{F}_q}^n$. Indeed, if $M \neq \mathrm{Id}$, then the invariant subspace $W = \{v : M\,v = v\}$ has codimension $\geqslant 1$. Therefore, with probability $\geqslant 1/2$, we have $v \notin W$ and $M\,v \neq v$. Further, observe that for all matrices $R, T \in \mathrm{GL}(n, \mathbb{F}_q)$, the probability $\mathbf{P}(R\,v = T\,v) \leqslant \frac{1}{2}$, given $R \neq T$. Indeed, $R\,v = T\,v$ is equivalent to $(T^{-1}\,R)\,v = v$, and the claim follows from the previous observation.

Now let $G = \langle g_1, \ldots, g_k \rangle$, $g_i \in \mathrm{GL}(n, \mathbb{F}_q)$. Let A, B be independent random subproducts of g_i. Assume that G is non-abelian. Then, for random $v \in {\mathbb{F}_q}^n$, we have

$$\mathbf{P}(A\,B\,v \neq B\,A\,v) = \mathbf{P}(A\,B \neq B\,A) \cdot \mathbf{P}(A\,B\,v \neq B\,A\,v \mid A\,B \neq B\,A) \geqslant \tfrac{1}{8}.$$

On the other hand, computing $A\,B\,v$ takes $O(k \cdot n^2)$ additions and multiplications in $\mathbb{F}_q$. Therefore, at a cost $O(k\,n^2 \eta)$, we can discover that G is non-abelian with probability of error $\leq 7/8$. This implies the result. □

Remark 1.6. Lemma 1.1 is motivated by the following result. Let G be non-abelian group. Then

$$\mathbf{P}([h_1, h_2] = \mathrm{id}) = \frac{\mathrm{r}(G)}{|G|} \leqslant \frac{5}{8},$$

where h_1, h_2 are uniform and independent in G and $\mathrm{r}(G)$ is the number of conjugacy classes. The identity goes back to Frobenius [**10**], while the inequality is due to MacHale [**16**].

Let us note that the proof of the latter follows along the lines of the proof of Lemma 1.1. Indeed, as the quotient $G/C(G)$ of a non-abelian group cannot be cyclic, we have $\mathbf{P}(h_1 \notin C(G)) \geqslant \frac{3}{4}$. Assuming $h_1 \notin C(G)$, we have $\mathbf{P}(h_2 \notin C(h_1)) \geqslant \frac{1}{2}$, and therefore $\mathbf{P}([h_1, h_2] \neq \mathrm{id}) \geqslant \frac{3}{8}$.

Remark 1.7. Following [**4, 6**], testing whether a black box group G is solvable or nilpotent can be done in time $O(\log^3 |G| \log\log |G| (\mu + \nu))$, given $k = O(\log |G|)$. One can ask whether this timing can be improved. We believe it can, assuming the following conjecture holds.

Conjecture 1.8. Let G be non-solvable and let h, h' be two independently chosen random subproducts. Then $\mathbf{P}(\langle h, h' \rangle$ is non-solvable $) \geqslant \varepsilon$ for some universal constant $\varepsilon > 0$.

The conjecture is motivated by Lemma 1.1, the previous remark and a recent result of Guralnick and Wilson [**11**] that

$$\mathbf{P}(\langle h, h' \rangle \text{ is non-solvable}) \geqslant \tfrac{11}{30}, \qquad (*)$$

where h, h' are uniform and independent in G. The inequality is tight for $G = A_5$.

Let us remark that the conjecture is open even for $G = A_n$. On the other hand, in $(*)$, the probability $\mathbf{P} \to 1$ as $n \to \infty$. Further, for two random permutations $\sigma, \sigma' \in A_n$, we have $\mathbf{P}(\langle \sigma, \sigma' \rangle = A_n) \sim 1 - 1/n$ (see for example [**23**] for references and estimates for other simple groups).

Theorem 1.9. *Let G be a black box group generated by k elements. Assuming Conjecture 1.8, there exists a Monte Carlo algorithm for testing whether the group is solvable at a cost*

$$O((k + L) \log^4(L) \log(1/\varepsilon)(\mu + \nu)),$$

where $L \leqslant \log |G|$ is an a priori upper bound on the length of the longest subgroup chain of G and $\varepsilon > 0$ is the probability of error.

For example, if $k = \theta(\log |G|)$, this gives a nearly linear algorithm for testing solvability. Note also that if $G \subset S_n$, then $L \leqslant 2n$ (see references in [4]).

https://doi.org/10.1112/S1461157012000046 Published online by Cambridge University Press

Proof. First, choose C independent random subproducts $\langle h_1, \ldots, h_C\rangle = H_1$. Their $O(C^2)$ commutators generate a subgroup whose normal closure is $H' = [H_1, H_1]$. Compute all $O(C^3)$ conjugates $[h_i, h_j]^{h_r}$ of the commutators by the generators of H_1. Now compute C independent random subproducts of these which generate H_2. Repeat the procedure. Stop after L iterations. If the obtained group H_L is non-trivial, then G is non-solvable and, if H_L is trivial, then G is solvable, both claims with probability of error α. Below, we show that the error α defined here is constant.

Observe that if the number of subproducts $C = C(\alpha)$ is chosen appropriately, then, at each step, C random subproducts generate a non-solvable group with probability $>1 - \alpha/L$. After L iterations the group H_L is then trivial or perfect, depending on whether G is solvable or non-solvable with probability of error $<\alpha$. Now check that it suffices to take $C = \lceil \log(L/\alpha)\rceil$. This follows from Conjecture 1.8 and the independence of successive random subproducts pairs that we sample.

Compute the cost of the algorithm. It costs $O(C\,k\,\mu)$ to generate the first C subproducts. It costs $O(C^3\,\mu)$ to compute $O(C^3)$ commutator conjugates, $O(C^4\,\mu)$ to compute C random subproducts of these, etc. The total number of iterations is L and, at each but the first one, the cost is $O(C^4\,\mu)$. Thus, the total cost is as in the theorem[†].

Finally, let $\alpha = 1/4$. Perform $O(\log(1/\varepsilon))$ of the tests as above and choose the most frequent answer. Now use the Chernoff bound to prove that it is correct with high probability $>1 - \varepsilon$. □

Remark 1.10. The notion of random subproducts goes back to Erdős and Rényi [**8**]. Versions of Lemma 1.3 had previously appeared in connection to various permutation group algorithms (see [**4, 6, 7**]). The problem of testing whether the group is abelian was motivated by the randomized algorithms for generating random elements in finite groups (see for example [**19**]). Note also that the algorithm applies to infinite black box groups as well, such as subgroups of $\mathrm{SL}(n, \mathbb{Z})$. Finally, the conjectured algorithm for testing whether a group is solvable can be transformed to test whether a group is nilpotent with little difficulty. Conjecture 1.8 seems to be also somewhat related to additive combinatorics on non-commutative groups [**25**].

2. *Deterministic algorithm*

We first need to define our model of computation. As before, we assume that the group G is given as a black box group with k generators $g_1, \ldots, g_k$. We assume that the multiplication and inversion (both at cost μ) and recognition of identity (called *identity test*, performed at cost ν) are done by an oracle. The '*commutativity checker*' is allowed to start with the generators and obtain new group elements out of these. Namely, at any time the checker can multiply any of the two elements found, invert any of them and compare any of them with the identity. The decision of which operation to perform can also depend on the previous observations of the checker. The goal of the checker is to determine whether G is abelian.

Theorem 2.1. *Every deterministic algorithm as above for testing whether a black box group G generated by k elements is abelian works at a cost $\Omega\big(k^2(\mu+\nu)\big)$. Moreover, every such algorithm requires at least $\binom{k}{2}$ identity tests.*

Proof. First we show that the checker needs at least $N = \binom{k}{2}$ calls for the identity test oracle. We will design a strategy (of which answers to give) for an adversary of the checker, so that after fewer than N such calls, there exist an abelian group A and a non-abelian group B, which satisfy all the answers.

The adversary starts with $A = \mathbb{Z}_2^k$ and $B = B_0$, where the latter is defined as

$$B_0 = \langle x_1, \ldots, x_k\rangle / (x_i^2 = [x_i, x_j]^2 = [x_i, [x_j, x_r]] = \mathrm{id}).$$

[†] In fact, the term involving k is much smaller, but for $k = o(L)$ small this term is irrelevant anyway.

https://doi.org/10.1112/S1461157012000046 Published online by Cambridge University Press

By definition, B_0 is a central extension of A, and the center $C(B_0) \simeq \mathbb{Z}_2^N$ is a group generated by all commutators $[x_i, x_j]$. We think of elements $g \in B_0$ as words

$$g = x_1^{\epsilon_1} \dots x_k^{\epsilon_k} \cdot [x_1, x_2]^{\alpha_{1,2}} \cdot [x_1, x_3]^{\alpha_{1,3}} \dots [x_{k-1}, x_k]^{\alpha_{k-1,k}},$$

where $\epsilon_i, \alpha_{j,r} \in \{0, 1\}$, $1 \leqslant i \leqslant k$, $1 \leqslant j < r \le k$.

After the ith call, $1 \leqslant i < N$, the adversary keeps an abelian group A and produces a new non-abelian group B_i as a quotient of B_{i-1}. By abuse of speech, denote by $S = \{x_1, \dots, x_k\}$ the natural set generators in both A and B_i.

Now, suppose at call i the checker asks whether the word w_i in generators $x_1, \dots, x_k$ is id or not. The adversary always answers depending on what is the truth in A. If indeed $w_i = \mathrm{id}$ in A, then the adversary sets $B_i = B_{i-1}/\langle w_i \rangle$. Observe that $w_i = \mathrm{id}$ in A only if w_i is equal in A to a product of commutators $[x_j, x_r]$. Now use induction to conclude that B_i is a nilpotent group of class 2 with center of rank $\geqslant N - i$.

We conclude that after $i < N$ calls of the identity recognition oracle, the checker is unable to distinguish between A and B_i, where the former group is abelian and the latter is not. This implies that the cost is $\Omega(k^2 \nu)$. Note also that for the algorithm to work one needs words w_i to generate the group $\mathbb{Z}_2^N$ generated by the commutators. But this requires at least N different words and therefore at least N group operations. Thus, the total cost is also $\Omega(k^2 \mu)$, which completes the proof. □

Remark 2.2. Recall the simple deterministic algorithm which consists of checking commutativity of all $\binom{k}{2}$ commutators. Observe that the lower bound in Theorem 2.1 is tight up to a small constant then. We believe that this algorithm is actually the fastest in our model of computation.

Remark 2.3. By analogy with the lower bound in Theorem 2.1, one can obtain a lower bound $\Omega(k^l/l)$ for testing whether a group is nilpotent of class $<l$. This is particularly striking in comparison with polynomial (in both k and $l < \log |G|$) randomized algorithms (see Remark 1.7).

To prove the above lower bound, apply the Witt formula (see [18]) to obtain a bound on the dimension for the free Lie algebra on k generators of degree l. This gives a lower bound $\Omega(k^l/l)$ on the number of words w to be used to ensure that all the commutators of length l in g_i are trivial. The proof follows if one proceeds as in the proof of Theorem 2.2.

Remark 2.4. In the study of black box groups it is often convenient to assume the knowledge of the upper bound M on the size of the group (in [2], Babai proved an exponential lower bound for approximating the size of a black box group). For example, for matrix groups or permutation groups this is the size of the group it is imbedded into. A straightforward modification of the proof gives the same lower bound under the assumption that $k < \lfloor \log_2 M \rfloor$. It would be interesting to extend this bound. Of course, the lower bound is false for large k: there exists an easy algorithm which lists group elements and compares and commutes new ones with them, at a cost $O(kM(\mu + \nu))$.

Remark 2.5. Since this paper appeared in a preprint form, both the lower and upper bounds found applications to quantum algorithms and quantum complexity [1, 17, 21].

Acknowledgements. This paper was written over eight years ago, but was never submitted. I am grateful to Benny Sudakov for urging me to publish it. Over the years, I was supported in part by the NSF, the NSA and the BSF; I am thankful to these agencies for their generous support. I am grateful to Sergey Bratus and Gene Cooperman for introducing me to the field and the random subproducts technique. I thank Laci Babai, Bob Guralnick, Greg

https://doi.org/10.1112/S1461157012000046 Published online by Cambridge University Press

Kuperberg, Leonard Schulman and Bill Thurston for helpful remarks, and Mike Newman and Alex Lubotzky for help with the references.

References

1. V. Arvind and P. Mukhopadhyay, 'Quantum query complexity of multilinear identity testing', *Proc. STACS 2009*, Dagstuhl Seminar Proceedings (Schloss Dagstuhl, 2009) 87–98.
2. L. Babai, 'Local expansion of vertex-transitive graphs and random generation in finite groups', *Proc. STOC 1991* (ACM, New York, 1991) 164–174.
3. L. Babai, 'Randomization in group algorithms: conceptual questions', *Groups and computation II* (American Mathematical Society, Providence, RI, 1997).
4. L. Babai, G. Cooperman, L. Finkelstein, E. M. Luks and Á. Seress, 'Fast Monte Carlo algorithms for permutation groups', *J. Comput. Sys. Sci.* 50 (1995) 296–308.
5. M. Blum, M. Luby and R. Rubinfeld, 'Self-testing/correcting with applications to numerical problems', *J. Comput. Syst. Sci.* 47 (1993) 549–595.
6. G. Cooperman and L. Finkelstein, 'Random algorithms for permutation groups', *CWI Quart.* 5 (1992) 107–125.
7. G. Cooperman and L. Finkelstein, 'Combinatorial tools for computational group theory', *Groups and computation I* (American Mathematical Society, Providence, RI, 1993).
8. P. Erdős and A. Rényi, 'Probabilistic methods in group theory', *J. Anal. Math.* 14 (1965) 127–138.
9. R. Freivalds, 'Fast probabilistic algorithms', *Mathematical foundations of computer science (Olomouc, 1979)* (Springer, Berlin, 1979) 57–69.
10. F. G. Frobenius, 'Über Gruppencharaktere (in German)', *Sitz. Berl. Akad.* (1896) 985–1021.
11. R. Guralnick and J. Wilson, 'The probability of generating a finite soluble group', *Proc. Lond. Math. Soc.* 81 (2000) 405–427.
12. D. F. Holt, B. Eick and E. A. O'Brien, *Handbook of computational group theory* (Chapman & Hall/CRC, Boca Raton, FL, 2005).
13. T. Kavitha, 'Linear time algorithms for abelian group isomorphism and related problems', *J. Comput. System Sci.* 73 (2007) 986–996.
14. E. M. Luks, 'Computing in solvable matrix groups', *Proc. FOCS 1992* (IEEE, 1992) 111–120.
15. E. M. Luks, 'Permutation groups and polynomial-time computation', *Groups and computation I* (American Mathematical Society, Providence, RI, 1993).
16. D. MacHale, 'How commutative can a non-commutative group be?', *Math. Gaz.* 58 (1974) 199–202.
17. F. Magniez and A. Nayak, 'Quantum complexity of testing group commutativity', *Algorithmica* 48 (2007) 221–232.
18. W. Magnus, A. Karrass and D. Solitar, *Combinatorial group theory*, 2nd edition (Dover, New York, 1976).
19. I. Pak, 'What do we know about the product replacement algorithm?', *Groups and computation, III (Columbus, OH, 1999)* (de Gruyter, Berlin, 2001) 301–347.
20. S. Rajagopalan and L. J. Schulman, 'Verifying identities', *Proc. FOCS 1996* (IEEE, 1996) 612–616.
21. M. Santha, *Quantum walk based search algorithms*, Lecture Notes in Computer Science 4978 (Springer, Berlin) 31–46.
22. A. Seress, *Permutation group algorithms* (Cambridge University Press, Cambridge, UK, 2003).
23. A. Shalev, 'Probabilistic group theory', *Groups St. Andrews 1997 in Bath*, London Mathematical Society Lecture Note Series 261 (Cambridge University Press, Cambridge, UK, 1999).
24. C. C. Sims, 'Computation with permutation groups', *Proceedings of the Second Symposium on Symbolic and Algebraic Manipulation* (ACM Press, New York, 1971) 23–28.
25. T. Tao and V. Vu, *Additive combinatorics* (Cambridge University Press, Cambridge, UK, 2006).
26. N. Vikas, 'An $O(n)$ algorithm for abelian p-group isomorphism and an $O(n \log n)$ algorithm for abelian group isomorphism', *J. Comput. System Sci.* 53 (1996) 1–9.
27. Ş. Yalçinkaya, 'Black box groups', *Turkish J. Math.* 31 (2007) 171–210.
28. J. Zumbrägel, G. Maze and J. Rosenthal, 'Efficient recovering of operation tables of black box groups and rings', *Proceedings of ISIT 2008,* Toronto, Canada (IEEE, 2008) 639–643.

Igor Pak
Department of Mathematics
University of California, Los Angeles
530 Portola Plazza
6363 Math Sciences Building
Los Angeles, CA 90095-1555
USA

pak@math.ucla.edu

https://doi.org/10.1112/S1461157012000046 Published online by Cambridge University Press