Journal of Scientific and Practical Computing
Vol.3, No.1 (2009) 3–9

Testing Group Commutativity in Constant Time

Bin Fu
Department of Computer Science, University of Texas-Pan American
Edinburg, TX 78539, USA
binfu@cs.panam.edu

Abstract: Lipton and Zalcstein presented a constant time algorithm for testing if a group is abelian in [12]. However, the reference only contains a short abstract without proof. In this paper, we give a self contained proof for an $\frac{n^2}{3}$ lower bound for the number of pairs (a, b) of elements with $ab \neq ba$ in every non-commutative group of size n. It implies a constant time randomized algorithm that tests if a group of n elements is commutative. Our lower bound for the number of non-commutative pairs (a, b) $(ab \neq ba)$ in a non-commutative group of size n has a generalized format $\frac{(p-1)(q-1)n^2}{pq}$, where $p > 1$ is the least integer divisor of n, and $q > p$ is the second least integer divisor of n.

keywords: abelian group, randomization, algorithm.

1. Introduction

Groups are fundamental algebraic structures and have applications in many fields. The classification of arbitrary group structures under isomorphism has not been fully understood. On the other hand, the description of structures of some restricted groups have been completely obtained. One example is the structures of finite commutative groups, which are also called abelian groups. It is well known that a finite abelian group can be decomposed to a direct product of cyclic groups with prime-power order (called cyclic p-groups) [8]. There are some interesting algorithms to find the decomposition of finite abelian group [1−4]. Buchmann and Schmidt [2] showed an $O(m\sqrt{|G|})$ time algorithm to compute the decomposition of an abelian group G with a set of generator of size m. Recently, Chen and Fu [4] showed linear time deterministic algorithm and sublinear time randomized algorithm to compute the decomposition of an abelian group.

There is a series of works in group isomorphism testing. No polynomial-time algorithm has been found for determining if two general groups are isomorphic. Miller [14] showed an $O(n^{\log n+O(1)})$ time algorithm for the group isomorphism problem. Savage [16] claimed the isomorphism between two abelian groups can be checked in $O(n^2)$ steps. Vikas [18] improved it to $O(n)$ time for the abelian p-group and $O(n \log n)$ time for abelian group. Kavitha [10] showed that the abelian group isomorphism problem can be computed in $O(n \log \log n)$ time. Garzon and Zalcstein [6] also discussed polynomial time algorithms for the isomorphism problem of abelian groups. Recently, Kavitha [9], using a new method for computing the orders of all elements in $O(n)$

time, obtained an $O(n)$ time algorithm for the abelian group isomorphism problem. The methods for computing the order for one element in an abelian group was also reported in [1,17]. More recently, Gall [5] showed efficient isomorphism algorithms for a class of nonabelian groups.

Lipton and Zalcstein presented a constant time algorithm for testing if a group is abelian group in [12]. However the reference only contains a short abstract without proof. Hong and Tang [7] developed $O(n^2)$ algorithms to test finite groups, rings, and fields. Kumar and Rubinfeld [11] presented a series of super-linear time algorithms to test the properties of group operations. Since finite abelian groups have clear structure description, a natural problem is to test the commutative property of a group. Kavitha [10] showed an $O(n)$ time deterministic algorithm to test if a group is commutative. Testing the commutativity of a group that has a set of generators $g_1, g_2, \cdots, g_k$ has been studied by Pak [15] in randomized computing model and by Magniez and Nayak [13] in quantum computing model. Pak developed a randomized algorithm with $O(k)$ group operations and proved that the deterministic algorithms have quadratic time lower bound. Magniez and Nayak designed a quantum algorithm with complexity $\tilde{O}(k^{\frac{2}{3}})$ and also proved lower bound $\Omega(k^{\frac{2}{3}})$ for quantum query complexity, and lower bound $\Omega(k)$ for randomized complexity.

We derive an $\frac{n^2}{3}$ lower bound for the number of pairs (a, b) with $ab \neq ba$ of elements in a non-commutative group of size n. It implies a constant time randomized algorithm to test if a group of n elements is commutative. The upper bound in our randomized algorithm has no conflict with the lower bound of Magniez and Nayak [13] since the input of our algorithm is the entire set of group instead of a set of generators used in the algorithms in [13,15]. This is in contrast to the previous $O(n)$ time deterministic algorithm of Kavitha [9], who also pointed out that all deterministic algorithms for group commutativity have an $\Omega(n)$-time lower bound. The input group with n elements to our algorithm is stored in an array of n elements and the multiplication table of the group can be accessed as a black box during the computation.

2. Notations

For a set A, $|A|$ denotes the number of elements in A. For two sets A and B, $A \cap B$ is the intersection set of A and B, $A - B$ is the set of all elements that are in A but not in B; and $A \times B$ is the set $\{(a, b)|a \in A \text{ and } b \in B\}$. For elements a, b in a set A, (a, b) represents a pair of elements of A. We note that all pairs considered in this paper are ordered. In other words, $(a, b) \neq (b, a)$ if $a \neq b$. For two integers x and y, $x|y$ means that $y = xc$ for some integer c (in other words, x is a divisor of y).

A *group* is a nonempty set G with a binary operation "$\cdot$", which is closed in set G and satisfies the following properties (for simplicity, "ab" represents "$a \cdot b$"): 1)for every three elements a, b and c in G, $a(bc) = (ab)c$ (associativity); 2)there exists an identity element $e \in G$ such that $ae = ea = a$ for every $a \in G$; 3)for every element $a \in G$, there exists an inverse $a^{-1} \in G$ with $aa^{-1} = a^{-1}a = e$. A group G is *finite* if G has only finite elements. Let e be the identity element of G, i.e. $ae = a$ for each $a \in G$. For a subgroup H of the group G and an element $a \in G$, define the *left coset* $aH = \{ah | h \in H\}$. A group G is an *abelian* group if $ab = ba$ for every pair of elements $a, b \in G$. If H is a subset of a group G with the binary operation $\cdot$ and is also a group under $\cdot$, then H is called a *subgroup* of G.

3. Lower Bound and Algorithm

Our constant time algorithm for testing group commutativity follows from a lower bound that every non-commutative group of size n has at least $\frac{n^2}{4} + \frac{n}{2}$ pairs of elements (a, b) with $ab \neq ba$. The lower bound will be improved later. In order to let the people without any group theory background understand the algorithm, we prove the following Lemma 3.1 and Lemma 3.2, which are well known in the classical group theory. The entire proof of our algorithm is self-contained.

Lemma 3.1. *Let G be a group and H be a finite subset of G. If $ab \in H$ for any $a, b \in H$, then H is a subgroup of G.*

Proof. We show that H has the identity element e of G, and for every element a of H, its inverse a^{-1} is also in H. Assume that $a_1, a_2, \cdots, a_m$ is the list of all elements in set H. Let a be an arbitrary element in H. Consider the list $aa_1, aa_2, \cdots, aa_m$. If $aa_i = aa_j$ for some a_i and a_j, then $a^{-1}aa_i = a^{-1}aa_j$, which implies $a_i = a_j$. Since H is closed under the multiplication, the elements in the list $aa_1, aa_2, \cdots, aa_m$ belong to H. Thus, the list $aa_1, aa_2, \cdots, aa_m$ is a permutation of the list $a_1, a_2, \cdots, a_m$ in H. Since $a \in H$, we have $aa_i = a$ for some $a_i \in H$. Therefore, $a_i = e$. Thus, $e \in H$. Furthermore, $aa_j = e$ for some $a_j \in H$. Therefore, element a has its inverse element a_j in H. □

Lemma 3.2. *Assume that G is a group and H is a subgroup of G. For any a, b in G, either $aH \cap bH = \emptyset$ or $aH = bH$.*

Proof. Assume $aH \cap bH \neq \emptyset$. There are elements $h \in H$ and $h' \in H$ with $ah = bh'$. We have $(ah)h^{-1} = (bh')h^{-1} = b(h'h^{-1})$. Since H is a subgroup of G, $h'h^{-1} \in H$. Therefore, $a \in bH$. We have $aH \subseteq bH$. Similarly, we also have $bH \subseteq aH$. Thus, $aH = bH$. □

Definition 3.1. Assume that G is a group. We have the following definitions:

- A pair (a, b) of elements in G is *non-commutative* if $ab \neq ba$.
- For each element $a \in G$, define $C_G(a) = \{b \in G | ab = ba\}$ and $\neg C_G(a) = G - C_G(a)$.
- Define $D_G = \{a \in G | ab \neq ba$ for some $b \in G\}$. In other words, $D_G = \{a \in G | \neg C_G(a) \neq \emptyset\}$.

Lemma 3.3. *Every non-commutative group G with n elements has at least $\frac{n^2}{4}+\frac{n}{2}$ non-commutative pairs.*

Proof. For two arbitary elements $b_1, b_2 \in C(a)$, we have the following equations:

$$\begin{aligned} a(b_1b_2) &= (ab_1)b_2 && (1)\\ &= (b_1a)b_2 && (2)\\ &= b_1(ab_2) && (3)\\ &= b_1(b_2a) && (4)\\ &= (b_1b_2)a. && (5) \end{aligned}$$

The transitions from (1) to (2), and (3) to (4) are due to the assumption that b_1 and b_2 are in $C_G(a)$. The other transitions follow from the associativity of the group operation. By the definition of $C_G(a)$, we have $b_1b_2 \in C_G(a)$. Since G is finite, so is $C_G(a)$. By Lemma 3.1, $C(a)$ is a subgroup of G.

Since G is not commutative, there exist elements $a \in D_G$ and $a' \in G$ such that $aa' \neq a'a$. By the definition of $C_G(a)$, we get $a' \notin C_G(a)$. By Lemma 3.2, G can be partitioned into equal size subsets $C_G(a), b_1C_G(a), \cdots, b_kC_G(a)$ such that there is no intersection between any two of them. Therefore, $|\neg C_G(a)| \geq \frac{n}{2}$. Since $aa = aa$, $a \in C_G(a)$ for every $a \in G$. Therefore, $|D_G| \geq |\neg C_G(a) \cup \{a\}| \geq \frac{n}{2}+1$. For every element $a \in D_G$, we have proved that $|\neg C_G(a)| \geq \frac{n}{2}$. Therefore, the number of non-commutative pairs is at least $\sum_{a\in D_G} |\neg C_G(a)| \geq |D_G|(\frac{n}{2}) \geq (\frac{n}{2}+1)\frac{n}{2} = \frac{n^2}{4}+\frac{n}{2}$. □

We now give a description of our algorithm.

```
Algorithm A
Input: a group $G$, its size $n = |G|$, and an integer parameter $m$;
Output: “Commutative” if $G$ is commutative, or “Non-commutative” otherwise;
begin
    Randomly select $m$ pairs of elements $(a_1, b_1)$,$(a_2, b_2)$, $\cdots$, and $(a_m, b_m)$
    in $G$;
    If $(a_ib_i \neq b_ia_i$ for some $i \in \{1, 2, \cdots, m\})$
    then return “Non-commutative”
    else return “Commutative”;
end
End of Algorithm A
```

Theorem 3.1. For every integer parameter $m > 0$ (which can be considered as a constant), there exists a randomized $O(m)$ time algorithm such that if group G is commutative, it always returns “Commutative”, and if group G is not commutative, it returns “Non-commutative” with probability at least $1 - (\frac{3}{4})^m$.

Proof. We show that Algorithm A has the property of the theorem. If G is a commutative group, $a_ib_i = b_ia_i$ for each pair (a_i, b_i) selected in the algorithm. Thus, the algorithm always returns "Commutative". Assume that G is not a commutative group. For a randomly selected pair (a_i, b_i) of elements in G, with probability at least $(\frac{\frac{n^2}{4}+\frac{n}{2}}{n^2}) > \frac{1}{4}$, we have $a_ib_i \neq b_ia_i$ (by Lemma 3.3). In other words, with probability at most $1 - \frac{1}{4} = \frac{3}{4}$, $a_ib_i = b_ia_i$ for a randomly selected pair (a_i, b_i) of elements from G. For m pairs (a_1, b_1),(a_2, b_2), $\cdots$, and (a_m, b_m) selected randomly, with probability at most $(\frac{3}{4})^m$, we have $a_ib_i = b_ia_i$ for $i = 1, 2, \cdots, m$. With probability at most $(\frac{3}{4})^m$, the algorithm returns "Commutative". Thus, with probability at least $1 - (\frac{3}{4})^m$, the algorithm returns "Non-commutative". □

An interesting problem is to identify the largest constant c_0 such that every non-commutative group has at least c_0n^2 non-commutative pairs. We have proved $c_0 > \frac{1}{4}$. The quaternion group Q_8 has 8 elements $\{1, -1, i, j, k, -i, -j, -k\}$ with the following fundamental equation $i^2 = j^2 = k^2 = ijk = -1$, which implies the following multiplication table:

$$ij = k, ji = -k,$$
$$jk = i, kj = -i,$$
$$ki = j, ik = -j.$$

The identity of Q_8 is 1. The elements 1 and -1 have the property $1a = a$, and $(-1)a = -a$ for every element $a \in Q_8$.

Let A be an abelian group of k elements. The group $G = A \times Q_8$ has $n = 8k$ elements, where the operation "$\cdot$" is defined by $(a, b) \cdot (c, d) = (ac, bd)$ for $(a, b), (c, d) \in G$. The subgroup $G' = A \times \{1, -1\}$ of G contains $2k = \frac{n}{4}$ elements. For every element $a \in G'$ and every element $b \in G$, we have $ab = ba$. $G' \times G$ contains $\frac{n^2}{4}$ elements. Therefore, the group G contains at most $n^2 - \frac{n^2}{4} = \frac{3n^2}{4}$ non-commutative pairs. Therefore, $c_0 \leq \frac{3}{4}$. A similar example was also used by Kavitha showing the lower bound $\Omega(n)$ for deterministic algorithms testing group commutativity [10]. We present the following theorem, which is a generalization of Lemma 3.3.

Theorem 3.2. Every non-commutative group G with n elements has at least $\frac{(p-1)(q-1)n^2}{pq}$ non-commutative pairs, where $p > 1$ is the least divisor of n, and $q > p$ is the second least divisor of n.

Proof. The proof is similar to that of Lemma 3.3. Let $C_G = \{a|ab = ba$ for every b in $G\}$ be the center of group G. We have $C_G = G - D_G$. Assume $a \in D_G$. $C_G(a)$ is a proper subgroup of G. Thus, $\frac{|G|}{|C_G(a)|} > 1$. On the other hand, $\frac{|G|}{|C_G(a)|}$ is a divisor of $|G| = n$. Since p is the least prime number with $p|n$, we have $\frac{|G|}{|C_G(a)|} \geq p$. Thus, $|C_G(a)| \leq \frac{n}{p}$ and $|\neg C_G(a)| \geq \frac{(p-1)n}{p}$. Since $\neg C_G(a) \subseteq D_G$, $a \in D_G$ and $a \notin \neg C_G(a)$, set D_G contains at least $\frac{(p-1)n}{p} + 1$ elements. Therefore, C_G contains at most $\frac{n}{p} - 1$ elements.

On the other hand, C_G is a subgroup of G. We have $|C_G| = \frac{|G|}{k}$ for some integer k with $k||G|$. Since $|C_G| \leq \frac{|G|}{p} - 1$, we have $k > p$, which implies $k \geq q$. Therefore, $|D_G| = |G| - |C_G| \geq n - \frac{n}{q} = \frac{(q-1)n}{q}$. Thus, group G has at least $\sum_{a \in D_G} |\neg C_G(a)| \geq |D_G|(\frac{(p-1)n}{p}) \geq \frac{(q-1)n}{q}(\frac{(p-1)n}{p}) = \frac{(p-1)(q-1)n^2}{pq}$ non-commutative pairs. □

It is easy to see that $\frac{(p-1)(q-1)}{pq} = \frac{1}{3}$ in Theorem 3.2 is minimal when $p = 2$ and $q = 3$. We have the following corollary, which slightly improves Lemma 3.3.

Corollary 3.1. *Every non-commutative group G with n elements has at least $\frac{n^2}{3}$ non-commutative pairs.*

Theorem 3.2 indicates an interesting property about non-commutative groups. For a non-commutative group G of size n with least prime p divisor, at least half of the pairs (a, b) in $G \times G$ are not commutative if $p \geq 3$.

4. Conclusion

In this paper, we obtained an $O(1)$-time randomized algorithm to check if a group is commutative. We do not know if $\frac{n^2}{3}$ is the optimal lower bound shared by all non-commutative groups of size n. We hope the gap between $\frac{n^2}{3}$ and $\frac{3n^2}{4}$ will be closed.

5. Acknoledgements

I would like to thank Li Chen for introducing me to the area of computational group theory, and also Feng Li for providing helpful comments. I would like to thank Francois Le Gall for his helpful comments that improve the presentation of this paper.

References

1. J. Buchman, M. J. Jacobson Jr., and E. Teske. On some computational problems in finite abelian groups. *Mathematics of Computation*, 66:1663–1687, 1997.
2. J. Buchmann and A. Schmidt. Computing the structure of a finite abelian group. *Mathematics of Computation*, 74:2017–2026, 2005.
3. L. Chen. Algorithms and their complexity analysis for some problems in finite group. *Journal of Sandong Normal University, in Chinese*, 2:27–33, 1984.
4. L. Chen and B. Fu. Linear and sublinear time algorithms for the basis of abelian groups. *Electronic Colloquium on Computational Complexity, TR07-052*, 2007.
5. F. L. Gall. Efficient Isomorphism Testing for a Class of Group *In Proceedings of the 26th International Symposium on Theoretical Aspects of Computer Science (STACS 2009)*, 625–636.
6. M. Garzon and Y. Zalcstein. On isomorphism testing of a class of 2-nilpotent groups. *Journal of Computer and System Sciences*, 42:237–248, 1991.
7. J. Hong and S. Tang. Some testing algorithms about finite groups, rings, and fields. *Advances in Mathematics*, 14:235–238, 1985.

8. M. I. Kargapolov and J. I. Merzljako. *Fundamentals of the Theory of Groups.* Springer-Verlag, 1979.
9. T. Kavitha. Linear time algorithms for abelian group isomorphism and related problem. *Journal of Computer and System Sciences,* Volume 73:986-996, September 2007.
10. T. Kavitha. Efficient algorithms for abelian group isomorphism and related problems. In *Proceedings of Foundations of Software Technology and Theoretical Computer Science, Lecture notes in computer science, 2914*, pages 277–288, 2003.
11. S. R. Kumar and R. Rubinfeld. Property testing of abelian group operations. Manuscript, 1998.
12. R. Lipton and Y. Zalcstein. Probabilistic algorithms for group-theoretic problems. Abstract appeared in ACM SIGSAM Bulletin, 1978.
13. F. Magniez and A. Nayak. Quantum complexity of testing group commutativity. *Algorithmica*, 48:221–232, 2007.
14. G. L. Miller. On the $n^{\log n}$ isomorphism technique. In *Proceedings of the tenth annual ACM symposium on theory of computing*, pages 128–142, 1978.
15. I. Pak. testing commutativity of a group and the power or randomization. Electronic version at http://www-math.mit.edu/ pak/research.html, 2000.
16. C. Savage. An $O(n^2)$ algorithm for abelian group isomorphism. Technical report, North Carolina State University, January 1980.
17. D. Shanks. Class number, a theory of factorization and genera. *Proc. Symp. Pure Math.*, 20:414–440, 1971.
18. N. Vikas. An $O(n)$ algorithm for abelian p-group isomorphism and an $O(n \log n)$ algorithm for abelian group isomorphism. *Journal of Computer and System Sciences*, 53:1–9, 1996.