On Testing Group Properties*

Oded Goldreich and Laliv Tauber
Department of Computer Science
Weizmann Institute of Science, Rehovot, Israel.

December 31, 2023

Abstract

Following Ergun *et al.* (JCSS 2000), we consider testing group properties and focus on the problem of testing whether a binary operation is a group operation. That is, given a finite set S and oracle access to a function $f : S \times S \to S$, we wish to distinguish the case that (S, f) constitutes a group from the case that f is far from any function g such that (S, g) is a group.

Our main result is a tester of running-time $\widetilde{O}(|S|)$, which improves over the tester of Ergun *et al.* that has running time $\widetilde{O}(|S|^{3/2})$.

Contents

*Partially supported by the Israel Science Foundation (grant No. 1041/18) and by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819702).

1 Introduction

In the last decades, the area of property testing has attracted much attention (see, e.g., the textbook [7]). Loosely speaking, property testing typically refers to sub-linear time probabilistic algorithms for deciding whether a given object has a predetermined property or is far from any object having this property. Such algorithms, called testers, obtain local views of the object by making adequate queries; that is, the object is modeled as a function and testers get oracle access to this function (and thus may be expected to work in time that is sub-linear in the size of the object).

The area of property testing first emerged in the context of testing algebraic properties such as group homomorphism (e.g., linearity) [3] and low-degree polynomials [14]. Hence, it is surprising that the problem of testing whether a matrix represents a group operation was first considered only a decade later, in [4], which is focused on a generalization (of property testing) called *spot checking*.[1]

Definition 1.1 (testing the property of being a group): *A* tester for being a group *is a probabilistic oracle machine that, on input parameters $n \in \mathbb{N}$ and $\epsilon > 0$, and oracle access to a function $f : [n] \times [n] \to [n]$ distinguishes between the following two cases.*

- *f* ***represents a group operation:*** *The set $[n] \stackrel{\text{def}}{=} \{1, 2, ..., n\}$ with the binary operation f is a group.*
- *f* ***is ϵ-far from representing a group operation:*** *For every $g : [n] \times [n] \to [n]$ such that $[n]$ with g constitutes a group, it holds that f is ϵ-far from g; that is, $|\{(x, y) \in [n]^2 : f(x, y) \neq g(x, y)\}| > \epsilon \cdot n^2$.*

The tester is said to have one-sided error *if it always accepts any function that represents a group operation.*

Ergun *et al.* [4] presented a tester for the group property that has running time $\widetilde{O}(n^{3/2}/\epsilon)$. Specifically, in [4, Sec. 4.1], they claimed a tester of time complexity $\widetilde{O}(n/\epsilon)$ under the guarantee that the function f is cancellative (i.e., $f(x, y) = f(x, y')$ implies $y = y'$ and $f(x, y) = f(x', y)$ implies $x = x'$), whereas in [4, Sec. 4.3] they waived this hypothesis/guarantee and claimed a tester of time complexity $\widetilde{O}(n^{2/3}/\epsilon)$.[2] Our main result is

Theorem 1.2 (efficiently testing the property of being a group (see Corollary 4.3)): *There exists a $\widetilde{O}(n/\epsilon)$-time tester for the property of being a group. Furthermore, the tester has one-sided error.*

The tester consists of natural tests of associativity and cancellativity. Actually, we use specific forms of these intuitive tests, whereas it is not clear whether different natural forms of the same intuitive tests will do.

We comment that focusing on the query complexity while ignoring the time complexity allows for a much simpler tester (for being a group). In fact, a generic test that applies to any *property of groups* is quite straightforward. Towards presenting this result, we generalized Definition 1.1 as follows.

[1] This generalization contains, as special cases, notions such as property testing, PCPs of proximity (see [2]), and interactive proofs of proximity (see [13]).

[2] We believe that there in a gap in the proof of Lemma 34 of [4, Sec. 4.1]: The second equality in the probabilistic claim seems to replace $\odot'$ by $\odot$ with no justification. This affect the correctness of both [4, Thm. 28] and [4, Thm. 49].

Definition 1.3 (testing properties of groups): *Let $\mathcal{G}$ be a set of finite groups that is closed under isomorphism; that is, for every group G in $\mathcal{G}$, all groups that are isomorphic to G are in $\mathcal{G}$.*[3] *A* tester for being in $\mathcal{G}$ *is a probabilistic oracle machine that, on input parameters $n \in \mathbb{N}$ and $\epsilon > 0$, and oracle access to a function $f : [n] \times [n] \to [n]$ distinguishes between the following two cases.*

- *f* ***represents the operation of a group in $\mathcal{G}$:*** *The set $[n] \stackrel{\text{def}}{=} \{1, 2, ..., n\}$ with the binary operation f is a group in $\mathcal{G}$.*
- *f* ***is ϵ-far from representing the operation of a group in $\mathcal{G}$:*** *For every $g : [n] \times [n] \to [n]$ such that $[n]$ with g constitutes a group in $\mathcal{G}$, it holds that f is ϵ-far from g; that is, $|\{(x, y) \in [n]^2 : f(x, y) \neq g(x, y)\}| > \epsilon \cdot n^2$.*

We may call $\mathcal{G}$ a property of groups. Using the fact that the number of groups over $[n]$ is $\exp(\widetilde{O}(n))$, we have

Theorem 1.4 (testing any property of groups (see Theorem 2.2)): *For any set $\mathcal{G}$ of finite groups that is closed under isomorphism, there exists a $\widetilde{O}(n/\epsilon)$-query tester for $\mathcal{G}$. Furthermore, the tester has one-sided error.*

Note that, unlike Theorem 1.2, which provides a computationally efficient tester, Theorem 1.4 does not bound the running time of the tester. Our proof of Theorem 1.4 uses a tester that runs in time that is exponential in its query complexity. Recall that Theorem 1.2 (as well as Theorem 1.5) does bound the time-complexity by $\widetilde{O}(n/\epsilon)$, but it only refers to the set of all groups (resp., all Abelian groups).

Yet another tester for the property of being a group is presented in Theorem 3.2. This tester uses $\widetilde{O}(n/\epsilon)$ queries but runs in time $\widetilde{O}(n^2)$ and has two-sided error. However, its proof is much simpler than the proof of Theorem 1.2. We mention that all three aforementioned testers can be augmented to test Abelian groups. In particular, we prove

Theorem 1.5 (efficiently testing the property of being an Abelian group (see Theorem 4.4)): *There exists a $\widetilde{O}(n/\epsilon)$-time tester for the property of being an Abelian group. Furthermore, the tester has one-sided error.*

(Note that $\widetilde{O}(n/\epsilon) = \widetilde{O}(n)/\epsilon$, because we may assume (w.l.o.g.) that $\epsilon \geq 1/n^2$.)

1.1 Related work

Following the work of Ergun *et al.* [4], studies of testing properties of groups appeared in two works [6, 8]. The first work (i.e., [6]) studies a related model in which the distance measure allows for an n-element group to be close to an $(n + 1)$-element group, whereas the second work studies both the model of [6] and the model that we use (i.e., Definition 1.3). Both works focus on testing properties of groups such as being Abelian and being cyclic. The results of [8] that refer to Definition 1.3 are (1) for every fixed $k \geq 2$, a lower bound on the query complexity of testing whether an Abelian group is generated by k elements, and (2) a polylogarithmic-time tester for cyclic groups.

[3] Two groups are called isomorphic if there exists a bijective group homomorphism between them.

We highlight [8, Lem. 3], although we do not use it in this work. It asserts (as a special case) that if two equal-sized groups are not isomorphic, then the tables of their operations disagree on at least 2/9 fraction of the entries (i.e., are 0.222-far apart).

We stress that testers are oracle machines that are given access to the main input, which is a function from $[n] \times [n]$ to $[n]$. Hence, bounds on their running time postulates that each query to the oracle is answered in unit time. Note that this model is essentially equivalent to the standard RAM model, except that the testers may afford not reading the entire input. In contrast, exact decision procedures as presented in [12, 5] do read the entire input, and have running-time $O(n^2)$, which is optimal for exact decision.

1.2 Organization

Our three testers of the group property are presented in three sections that are independent of one another. Our main result (i.e., Theorem 1.2) is presented in Section 4 (which contains also a proof of Theorem 1.5).

The other sections are much easier, but reading them is not a useful warm-up towards reading Section 4. Section 2 presents a generic tester (which establishes Theorem 1.4), whereas Section 3 presents an alternative tester that makes $\widetilde{O}(n/\epsilon)$ queries but runs in time $\widetilde{O}(n^2)$ and has two-sided error.

2 A generic result

A generic tester for any property (equiv., set) of groups $\mathcal{G}$ is derived from a generic tester for any property (equiv., set) of functions. Indeed, let $\mathcal{F}$ be an arbitrary set of functions from D to R. Then, $\mathcal{F}$ can be tested within query complexity $O(\epsilon^{-1} \log |\mathcal{F}|)$.

Theorem 2.1 (a totally generic tester): *Testing whether $f : D \to R$ is in the set $\mathcal{F}$ can be done using $O(\epsilon^{-1} \log |\mathcal{F}|)$ queries, and $\widetilde{O}(|\mathcal{F}|/\epsilon)$ time. Furthermore, the tester has one-sided error.*

Proof: The tester just picks $s = O(\epsilon^{-1} \log |\mathcal{F}|)$ random points in $[n]$ and checks whether the value of f on them is consistent with some function in $\mathcal{F}$. (Indeed, the time complexity of this tester is $O(s \cdot |\mathcal{F}|)$.)

Clearly, this tester always accepts any $f \in \mathcal{F}$. On the other hand, if f is ϵ-far from $\mathcal{F}$, then for every $f' \in \mathcal{F}$ it holds that

$$\Pr_{a_1,...,a_s \in D}[(\forall i \in [s])\, f(a_i) = f'(a_i)] < (1-\epsilon)^s.$$

Using a union bound (and noticing that $(1-\epsilon)^s = o(1/|\mathcal{F}|)$), it follows that the tester rejects f with probability at least 2/3. ■

Theorem 2.2 (generic tester of group properties (Theorem 1.4, restated)): *Let $\mathcal{G}_n$ be a set of n-element groups. Then, testing whether $f : [n] \times [n] \to [n]$ describes a group that is isomorphic to a group in $\mathcal{G}_n$ can be done using $\widetilde{O}(n)/\epsilon$ queries, and $\exp(\widetilde{O}(n))/\epsilon$ time. Furthermore, the tester has one-sided error.*

Proof: The key observation is that the number of non-isomorphic groups over $[n]$ is $\exp(O(\log n)^3)$; see [10, 9, 11]. Now, letting $\mathcal{F}$ be the set of all functions obtained by relabelling all $g \in \mathcal{G}_n$, the task of testing the group property $\mathcal{G}_n$ reduces to testing membership in $\mathcal{F}$, whereas $|\mathcal{F}| \leq n! \cdot |\mathcal{G}_n|$. Using Theorem 2.1, the claim follows. ■

3 Testing whether f represents a group, take 2

The down side of Theorem 2.2 is its prohibitively high computational complexity (i.e., $\exp(\widetilde{O}(n))$-time). Here, we significantly improve the computational complexity at the cost of allowing two-sided error probability, but we still do not reach sub-linear (in n^2) time.

The basic idea is any group can be generated by a small set of expanding generators; specifically, for any group of n elements, a random set of $O(\log n)$ elements can serve as a set of expanding generators [1], where expanding means that composing a random $O(\log n)$-long sequence of generators (and inverses) yields an almost uniformly distributed element. Hence, the group operation (i.e., $[n] \times [n] \to [n]$) is fully specified by the values assigned to pairs of the form $[n] \times S$, where S is a generating set (which includes the inverses of elements in S). Specifically, we construct the following succinct data structure that allows to perform group operations in logarithmic time.

Construction 3.1 (producing a succinct representation of an alleged group): *Given oracle access to $f : [n] \times [n] \to [n]$, we proceed as follow.*

1. Finding a neutral element. *This is done by picking an arbitrary $a \in [n]$, and finding the unique $e \in [n]$ that satisfies $f(a, e) = f(e, a) = a$.*

   If this or any of the next steps fails, then we halt while indicating failure.

2. Finding a set of expanding generators (along with their inverses). *We select uniformly at random a set S of $O(\log n)$ elements in $[n]$, and for each $s \in S$ find the unique $s' \in [n]$ such that $f(s, s') = f(s', s) = e$.*

   We redefine S to contain S as well as the foregoing inverses. We also record $I : S \to S$ such that $I(s)$ is the inverse of s.

3. Constructing the main table. *Querying f on $[n] \times S$, we construct $T : [n] \times S \to [n]$ such that $T(a, s) = f(a, s)$ for every $a \in [n]$ and $s \in S$.*

   (Actually, this information is already available to us (from the search for inverses of S performed in the previous step).)

4. Expressing all elements in terms of the generating set. *We generate $O(n \log n)$ random sequences over S, each of length $\ell = O(\log n)$, in hope of reaching all n elements of the group. The element generated by a sequence $(s_1, ..., s_\ell) \in S^\ell$ is obtained by repeated composition. Specifically, for each sampled sequence $(s_1, ..., s_\ell) \in S^\ell$, we compute the value $f(\cdots(f(f(s_1, s_2), s_3), ...), s_\ell)$ by using $T(\cdots(T(T(s_1, s_2), s_3), ...), s_\ell)$.*

   For each $x \in [n]$, we let $(s_1^x, ..., s_\ell^x)$ denote a sequence that is used to generate x. We define $E : [n] \times [\ell] \to S$ such that $E(x, i) = s_i^x$.

   Note that no queries are made at this step.

We output e, S and the tables I, T and E.

Overall, we made $O(n) + O(n \log n) + O(n \log n)$ queries and spent $\widetilde{O}(n)$ time. We stress that if f represents a group, then, with high probability, S is a set of expanding generators, and the sample of ℓ-sequences reaches all elements of the group. In this case, the construction returns the desired tables. In any case, if Construction 3.1 returns tables as stipulated, then we define the following function $f' : [n] \times [n] \to [n]$

$$f'(x, y) \stackrel{\text{def}}{=} T(\cdots(T(T(x, E(y, 1)), E(y, 2)), ...), E(y, \ell)). \tag{1}$$

That is, starting with $v \leftarrow x$, for $i = 1, ..., \ell$, we let $v \leftarrow T(v, E(y, i))$, and finally let $f'(x, y) \leftarrow v$. Note that evaluating f' amounts to $O(\log n)$ accesses to the tables E and T. Hence, testing whether f represents a group (resp., a group with certain properties) reduces to invoking Construction 3.1, testing whether f' equals f, and checking that f' satisfies the group property (resp., properties).

Theorem 3.2 (testing the group property): *Testing whether $f : [n] \times [n] \to [n]$ represents a group can be done using $\widetilde{O}(n) + O(1/\epsilon)$ queries, and $\widetilde{O}(n^2) + O(\epsilon^{-1} \cdot \log n)$ time.*

Note that this tester has two-sided error probability, which is due to the (small) error probability of Construction 3.1.

Proof: As stated upfront, we first invoke Constuction 3.1 and reject if it failed. Otherwise, we obtain tables that allow us to compute f' in logarithmic time, while making no additional queries, where f' is as defined in Eq. (1). In this case, we check that f' represents a group, and make $O(1/\epsilon)$ additional queries (to f) in order to test that f equals f' (while rejecting (w.h.p.) when f is ϵ-far from f'). Specifically, checking that f' represents a group is done by explicitly constructing f' (according to Eq. (1)) and checking all group axioms (see details below). Testing equality between f and f' amounts to querying f at $O(1/\epsilon)$ random entries, and comparing these value to the corresponding values obtained from f'.

As for checking that f' satisfies all group axioms, a straightforward implementation allows checking the unique existence of a neutral element and the inverses in $O(n^2)$ time, whereas associativity is checked in $O(n^3)$ time. We improve upon the latter by observing that there is no need to check all triplets in $[n]$; it suffices to check triplets in which one of the elements is confined to S. Actually, this observation was made by Light in 1949 (see [12]), and we prove it next for sake of self-containment.

Claim: If f' satisfies $f'(x, f'(y, z)) = f'(f'(x, y), z)$ for all $x, y, z \in [n]$ such that $\{x, y, z\} \cap S \neq \emptyset$, then f' satisfies $f'(x, f'(y, z)) = f'(f'(x, y), z)$ for all $x, y, z \in [n]$.

Proof: Let $\overline{f}(s) \stackrel{\text{def}}{=} s$ for every $s \in S$. Using induction on $t \geq 1$, we define $\overline{f} : S^t \to [n]$ such that $\overline{f}(s_1, s_2, ..., s_t) \stackrel{\text{def}}{=} f'(s_1, \overline{f}(s_2, ..., s_t))$. We first prove that

$$f'(\overline{f}(s_1, ..., s_{t'}), \overline{f}(s_{t'+1}, ..., s_t)) = f'(\overline{f}(s_1, ..., s_{t''}), \overline{f}(s_{t''+1}, ..., s_t)) \tag{2}$$

for every $t', t'' \in [t-1]$.
The special case of $t' = 1$ implies that $\overline{f}(s_1, s_2, ..., s_t) = f'(\overline{f}(s_1, ..., s_{t''}), \overline{f}(s_{t''+1}, ..., s_t))$ for every $t'' \in [t-1]$.

We prove Eq. (2) by induction on t. Assuming, without loss of generality that $t'' < t'$, we first establish Eq. (2) for the case of $t'' = t' - 1 \in [t-1]$. In this case, we have

$$\begin{aligned} f'(\overline{f}(s_1, ..., s_{t'}), \overline{f}(s_{t'+1}..., s_t))) &= f'(f'(\overline{f}(s_1, ..., s_{t'-1}), s_{t'}), \overline{f}(s_{t'+1}..., s_t))) \\ &= f'(\overline{f}(s_1, ..., s_{t'-1}), f'(s_{t'}, \overline{f}(s_{t'+1}..., s_t))) \\ &= f'(\overline{f}(s_1, ..., s_{t'-1}), \overline{f}(s_{t'}, s_{t'+1}..., s_t)), \end{aligned}$$

where the second equality uses the claim's hypothesis (with $x = \overline{f}(s_1, ..., s_{t'-1})$, $y = s_{t'}$ and $z = \overline{f}(s_{t'+1}..., s_t)$) and the other equalities use the induction hypothesis. The general case (of $t'' < t'$) is handled by $t' - t''$ iterations of the foregoing special case (i.e., going from t' to $t' - 1$, then to $t' - 2$, and similarly all the way till $t' - (t' - t'')$).

Having established Eq. (2), we turn to the actual claim. Recalling that any $w \in [n]$ can be written as $\overline{f}(E(w,1), ..., E(w,\ell)) = \overline{f}(s_1^w, ..., s_\ell^w)$, we establish $f'(x, f'(y,z)) = f'(f'(x,y),z)$ by using

$$\begin{aligned} f'(\overline{f}(s_1^x, ..., s_\ell^x), f'(\overline{f}(s_1^y..., s_\ell^y), \overline{f}(s_1^z..., s_\ell^z))) &= f'(\overline{f}(s_1^x, ..., s_\ell^x), \overline{f}(s_1^y..., s_\ell^y, s_1^z..., s_\ell^z)) \\ &= f'(\overline{f}(s_1^x, ..., s_\ell^x, s_1^y..., s_\ell^y), \overline{f}(s_1^z..., s_\ell^z)) \\ &= f'(f'(\overline{f}(s_1^x, ..., s_\ell^x), \overline{f}(s_1^y..., s_\ell^y)), \overline{f}(s_1^z..., s_\ell^z)). \end{aligned}$$

This completes the proof of the claim. ■

Conclusion: As stated upfront, if f represents a group, then, with high probability, Constuction 3.1 returns $f' = f$, and in this case our tester accepts. On the other hand, if f is ϵ-far from representing a group, then either Constuction 3.1 rejects or it returns f' that is either ϵ-far from f or does not represent a group. In each of these cases, our tester rejects (with high probability in case f' is ϵ-far from f). ■

Digest. The proof of Theorem 3.2 reduces testing a function $f : [n] \times [n] \to [n]$, which supposedly represents a group, to checking a data structure of size $\widetilde{O}(n)$. Recall that the latter data structure is created by Constuction 3.1, which uses $\widetilde{O}(n)$ queries to f (and $\widetilde{O}(n)$ times). When f represents a group, the data structure allows to compute f in logarithmic time (without access to f itself). In general, this data structure yields a function f', and we can test whether $f' = f$ and reduce testing properties of f to checking properties of f'. In particular, we tested whether f is a group by checking f' represents a group (and $f' = f$). Likewise, we can test whether f represents an Abelian group by checking whether f' represents an Abelian group.

4 Main result: Testing whether f represents a group, take 3

In this section, we present a tester of sub-linear (in n^2) time complexity; specifically, the tester will have time complexity $\widetilde{O}(n/\epsilon)$. Here it is instructive to view $f : [n] \times [n] \to [n]$ as an n-by-n matrix (or table). Intuitively, Steps 1 and 2 in Construction 4.1 constitute a test of cancellativity, whereas Step 3 is a test of associativity. Recall that f represents a group if and only if it is both cancellative (i.e., $f(x,y) = f(x,y')$ implies $y = y'$ and $f(x,y) = f(x',y)$ implies $x = x'$) and associative (i.e., $f(x, f(y,z)) = f(f(x,y),z)$).

We comment that Steps 1 and 2 in Construction 4.1 replace a less efficient test that appears in [4, Sec. 4.3], which checks for *collisions* in *each* row of f. In contrast, Step 3 in Construction 4.1 is identical to the tester that appears in [4, Sec. 4.1.1]. (Our analysis utilizes all three steps, but it may be that Step 2 is actually unnecessary.)

Construction 4.1 (a direct tester for the group property): *On input $n \in \mathbb{N}$ and $\epsilon > 0$, and oracle access to $f : [n] \times [n] \to [n]$, viewed as an n-by-n matrix over $[n]$, the tester proceeds as follows.*

1. Check that random rows and columns contains permutations: *Select uniform at random $O(1/\epsilon)$ elements $r \in [n]$, and verify that for each of these r's the mappings $x \mapsto f(r,x)$ and $x \mapsto f(x,r)$ are permutations. Otherwise, reject.*

2. Check that random rectangles consist of columns that contain no repeated values:[4] *For each $i \in [\log_2 n]$, select uniformly at random a set of $O(n/2^i)$ rows and a set of $O((2^i \log n)/\epsilon)$ columns and check that each value appears at most once in each of the residual columns. Otherwise, reject.* (That is, if for one of the selected columns c and two selected rows $r_1 \neq r_2$ it holds that $f(r_1,c) = f(r_2,c)$, then reject.)

3. Checking associativity. *Specifically, for random values of two of the three variables, we check the associativity condition for all values of the remaining variable. This is done for each of the possible three choices of the identity of the remaining variable.*

   Repeat each of the following three checks $O(1/\epsilon)$ times.

   (a) Checking all values of the first variable: *Select uniformly at random $r,s \in [n]$, and verify that $f(a,f(r,s)) = f(f(a,r),s)$ holds for every $a \in [n]$.*

   (b) Checking all values of the second variable: *Select uniformly at random $r,s \in [n]$, and verify that $f(r,f(a,s)) = f(f(r,a),s)$ holds for every $a \in [n]$.*

   (c) Checking all values of the third variable: *Select uniformly at random $r,s \in [n]$, and verify that $f(r,f(s,a)) = f(f(r,s),a)$ holds for every $a \in [n]$.*

   Reject if any of these checks fails.

If none of the steps rejected, then accept.

The time complexity of the foregoing tester is $O(n/\epsilon)+O(\epsilon^{-1} \cdot n \log^2 n)+O(n/\epsilon) = O(\epsilon^{-1} \cdot n \log^2 n)$, and it always accepts if f represents a group operation. We shall show that, if the tester accepts with probability at least $1/3$, then f is ϵ-close to representing a group operation.

Theorem 4.2 (Construction 4.1 rejects (whp) functions that are ϵ-far from respresenting a group): *If Construction 4.1 accepts f with probability at least $1/3$, then f is ϵ-close to representing a group operation.*

Proof: Using a sufficiently small $\epsilon' = \Omega(\epsilon)$, we show that if Construction 4.1 accepts f with probability at least $1/3$, then f is $O(\epsilon')$-close to a self-corrected version (defined in Eq. (3) below), which in turn represents a group operation. Specifically, after establishing a few preliminary claims, we shall show that the latter function is cancellative and associative, which implies that it describes a group.

Let R (resp., C) denote the set of r's such that $x \mapsto f(r,x)$ (resp., $x \mapsto f(x,r)$) is a permutation, and note that $|R| \geq (1-\epsilon') \cdot n$ (resp., $|C| \geq (1-\epsilon') \cdot n$), since otherwise Step 1 rejects with high probability. The lower bounds on $|R|$ and $|C|$ implies that each value occurs at most $\min(|R|,|C|)$ times in the rectangle $R \times C$, but it does not imply that each value occurs at least $\min(|R|,|C|)$ times. Nevertheless, a slightly weaker statement does hold.

[4] Indeed, it is natural to perform the same check for rows, but this is unnecessary for our analysis. Performing both checks covers Step 1 as a special case for $i = \log_2 n$ and $i = 1$, respectively.

Claim 4.2.1 ($R \times C$ contains at least $|R| - 2\epsilon' n$ occurrences of each value): *For every $v \in [n]$, it holds that $|\{(r,c) \in R \times C : f(r,c) = v\}| \geq |R| - 2\epsilon' n$.*

Note that if f is guaranteed to be cancellative, as in [4, Sec. 4.1], then $R = C = [n]$, and Claims 4.2.1 and 4.2.2 hold vacuously. In this case, Steps 1 and 2 of Construction 4.1 are unnecessary, and the time complexity of the tester is reduced to $O(n/\epsilon)$.

Proof: Fixing v and letting $\overline{C} \stackrel{\text{def}}{=} [n] \setminus C$, we shall prove that $|\{(r,c) \in [n] \times \overline{C} : f(r,c) = v\}| \leq 2\epsilon' n$, and the claim will follow (since each $r \in R$ contains the value v). For each $c \in \overline{C}$, we let n_c denote the number of v's in column c. Assuming, towards the contradiction that $\sum_{c \in \overline{C}} n_c > 2\epsilon' n$, we shall show that in such a case Step 2 rejects with high probability.

Letting $\ell = \log_2 n$, for each $i \in [\ell]$, we define $B_i = \{c \in \overline{C} : 2^i \leq n_c < 2^{i+1}\}$, and observe that $B_0 \stackrel{\text{def}}{=} \overline{C} \setminus \bigcup_{i \in [\ell]} B_i = \{c \in \overline{C} : n_c \leq 1\}$, whereas $\sum_{c \in B_0} n_c \leq |\overline{C}|$. Hence,

$$\sum_{i \in [\ell]} \sum_{c \in B_i} n_c \;>\; 2\epsilon' n - |\overline{C}| \;\geq\; \epsilon' n.$$

It follows that there exists $i \in [\ell]$ such that $\sum_{c \in B_i} n_c > \epsilon' n/\ell$. Fixing this i, observe that $|B_i| > \frac{\epsilon' n/\ell}{2^{i+1}} = \frac{\epsilon'}{2^{i+1}\ell} \cdot n$. Hence, with high probability, a random set of $O(2^i \ell/\epsilon)$ columns contains a column in B_i, denoted c. Recalling that $n_c \geq 2^i \geq 2$, with high probability, a random set of $O(n/2^i)$ rows contains two distinct rows r_1, r_2 such that $f(r_1, c) = v = f(r_2, c)$. But in such a case, Step 2 rejects. The claim follows. ■

A self-corrector for f. For any $v \in [n]$ and $r \in R$, let $c_v(r)$ be the unique value c that satisfies $f(r,c) = v$, where unique existence is guaranteed by $r \in R$ (which implies that v occurs exactly once in row r). Letting $a \odot b = f(a,b)$, we define a candidate self-correcting function $g : [n] \times [n] \to [n]$ as follows[5]

$$g(x,y) \stackrel{\text{def}}{=} \text{Plurality}_{r \in R}\{(x \odot r) \odot c_y(r)\}. \tag{3}$$

Indeed, for every $x, y \in [n]$ and $r \in R$, it holds that $x \odot (r \odot c_y(r)) = x \odot y$, but f is not necessarily associative (i.e., $(x \odot r) \odot c_y(r)$ may not equal $x \odot (r \odot c_y(r))$). Nevertheless, since F is "close to being associative" (per being rarely rejected by Step 3), one may hope that $(x \odot r) \odot c_y(r) = x \odot (r \odot c_y(r))$ typically holds (i.e., holds for $1 - O(\epsilon')$ fraction of the r's in R). This is indeed proved in Claim 4.2.3, but our proof (as well as other proofs that follow) uses the following claim.

Claim 4.2.2 (c_v is almost uniformly distributed in $[n]$): *For every $v \in [n]$, when selecting uniformly $r \in R$, it holds that $c_v(r)$ is $4\epsilon'$-close to the uniform distribution over $[n]$.*

Proof: Fixing v, we call $r \in R$ good if $c_v(r) \in C$, and observe that good r's yield different $c_v(r)$'s (because $c \stackrel{\text{def}}{=} c_v(r) = c_v(r')$ imply $r \odot c = v = r' \odot c$, which implies $r = r'$ since $c \in C$). Recalling

[5] It seems that a more natural (and "symmetric") form of a self-corrector is given by

$$\text{Plurality}_{r,s \in R}\{s \odot ((c_x(s) \odot r) \odot c_y(r))\}.$$

However, as shown in the auxiliary claim of Claim 4.2.5, this more complicated form is equivalent to the one we use below (i.e., in Eq. (3)).

that (by Claim 4.2.1) the number of v-entries in columns that is not in C is at most $2\epsilon' \cdot n$, it follows that at most $2\epsilon' n$ rows in R have a v-entry in a column that is not in C, Hence,

$$\Pr_{r\in R}[c_v(r) \in C] \geq \frac{|R| - 2\epsilon' n}{|R|} > 1 - 3\epsilon';$$

that is, $r \in R$ is good with probability at least $1 - 3\epsilon'$. Recalling that different good r's (in R) have different values of $c_v(r)$, it follows that, for a uniformly distributed $r \in R$, the distribution of $c_v(r)$ is $3\epsilon'$-close to be uniform over some set of $|R|$ values. The claim follows. ■

Claim 4.2.3 (g is supported by a strong majority): *For every $a, b \in [n]$,*

$$\Pr_{r,r'\in R}\left[(a \odot r) \odot c_b(r) = (a \odot r') \odot c_b(r')\right] = 1 - O(\epsilon'). \tag{4}$$

Equivalently, for every $a, b \in [n]$, it holds that $\Pr_{r\in R}[(a \odot r) \odot c_b(r) = g(a, b)] = 1 - O(\epsilon')$.

The following proof (as well as subsequent ones) makes essential use of the specific way in which associativity is checked in Step 3. Specifically, the fact that (w.h.p.) none of the invocations of Step 3a rejects implies that

$$\Pr_{r,s\in[n]}\left[(\forall a \in [n])\, f(a, f(r, s)) = f(f(a, r), s)\right] \;>\; 1 - \epsilon'. \tag{5}$$

We shall use the fact that Eq. (5) implies that, for every function $\alpha : [n] \times [n] \to [n]$, it holds that

$$\Pr_{r,s\in[n]}\left[f(\alpha(r, s), f(r, s)) = f(f(\alpha(r, s), r), s)\right] \;>\; 1 - \epsilon'. \tag{6}$$

Analogously, by Steps 3b and 3c, we get

$$\Pr_{r,s\in[n]}\left[f(r, f(\alpha(r, s), s)) = f(f(r, \alpha(r, s)), s)\right] \;>\; 1 - \epsilon' \tag{7}$$

$$\Pr_{r,s\in[n]}\left[f(r, f(s, \alpha(r, s)))) = f(f(r, s), \alpha(r, s))\right] \;>\; 1 - \epsilon'. \tag{8}$$

Proof: To gain some intuition, note that Eq. (8) immediately implies that Eq. (4) holds for any fixed $b \in [n]$ and a uniformly distributed $a \in [n]$. Furthermore, fixing $a \in [n]$ and considering a random $b \in [n]$, one may use Claim 4.2.1 to show that $c_b(r)$ is distributed almost independently of r, and then derive Eq. (4) by using Eq. (6). The actual claim, which refers to any $a, b \in [n]$, is proved by a more complicated argument, which employs the foregoing ideas more extensively.

Fixing any $a, b \in [n]$, we consider r and r' that are distributed independently and uniformly in R. Then, by Claim 4.2.2, we have $\Pr_{r\in R}[c_b(r) \in C] \geq 1 - 4\epsilon'$, whereas $c_b(r')$ is defined per $r' \in R$. In this case (i.e., $c_b(r) \in C$), there exists a unique $d = d_b(r, r')$ such that $d \odot c_b(r) = c_b(r')$. Hence, it holds that

$$\Pr_{r,r'\in R}[r' \odot c_b(r') = r' \odot (d_b(r, r') \odot c_b(r))] \;\geq\; 1 - 4\epsilon'. \tag{9}$$

Recalling that $c_b(r)$ is $4\epsilon'$-close to uniform (and r' is ϵ'-close to uniform), by Eq. (7) we have

$$\Pr_{r,r'\in R}\left[r' \odot (d_b(r, r') \odot c_b(r)) = (r' \odot d_b(r, r')) \odot c_b(r)\right] \;\geq\; 1 - 6\epsilon'. \tag{10}$$

Combining $r \odot c_b(r) = r' \odot c_b(r')$ with Eq. (9)&(10), we get

$$\Pr_{r,r'\in R}\left[r \odot c_b(r) = (r' \odot d_b(r,r')) \odot c_b(r)\right] \;\geq\; 1 - 10\cdot\epsilon'. \tag{11}$$

Recalling that $c_b(r) \in C$ means that there is a unique value w such that $w \odot c_b(r) = b$ (i.e., $w = r$), it follows that $\Pr_{r,r'\in R}[r = r' \odot d_b(r,r')] \;=\; 1 - O(\epsilon')$. Hence, for every $a, b \in [n]$, we have

$$\Pr_{r,r'\in R}\left[(a \odot r) \odot c_b(r) = (a \odot (r' \odot d_b(r,r'))) \odot c_b(r)\right] \;=\; 1 - O(\epsilon'). \tag{12}$$

Assuming that $(a \odot (r' \odot d_b(r,r'))) \odot c_b(r)$ equals $(a \odot r') \odot (d_b(r,r') \odot c_b(r))$, which in turn equals $(a \odot r') \odot c_b(r')$, would have established the claim; that is, under this assumption, Eq. (12) implies $\Pr_{r,r'\in R}[(a \odot r) \odot c_b(r) = (a \odot r') \odot c_b(r')] \;=\; 1 - O(\epsilon')$. However, associativity is only guaranteed by Eq. (6)–(8) *when two of the three operands are uniformly and independently distributed.* Thus, towards applying the associativity guaranteee, we first argue that the joint distributions of $(r', d_b(r,r'))$ and $(d_b(r,r'), c_b(r))$ are (each) almost uniformly distributed in $[n] \times [n]$.

Recall that, for $r', r \in R$ such that $c_b(r) \in C$, it holds that $d_b(r,r') \odot c_b(r) = c_b(r')$ uniquely determines $d_b(r,r')$ as a function of $c_b(r)$ and $c_b(r')$. Considering uniformly distributed $r, r' \in R$, we show that the distribution of $d_b(r,r')$ is almost uniform (over $[n]$) and independent of the distribution of r' (resp., $c_b(r)$).

- Fixing an arbitrary $r' \in R$ and letting r be uniformly distributed in R, we wish to show that $d_b(r,r')$ is almost uniformly distributed in $[n]$.

  We first recall that (by Claim 4.2.1) the rectangle $R \times C$ contains at least $n' \stackrel{\text{def}}{=} |R| - 2\epsilon' n \geq (1 - 3\epsilon') \cdot n$ entries that hold the value $c_b(r')$, whereas each row in R (resp., column in C) contains a single $c_b(r')$ value. Hence, $R \times C$ contains an n'-by-n' square, denoted $R' \times C'$, such that each row in R' (resp., column in C') contains a single $c_b(r')$ value that appears in a column in C' (resp., a row in R'). It follows that, for a uniformly distributed $c \in C'$, the unique d that satisfies $d \odot c = c_b(r')$ is uniformly distributed in R'.

  Next, we note that distinct r's in R such that $c_b(r) \in C$ must have distinct $c_b(r)$'s values. Hence, for a random $r \in R$ such that $c_b(r) \in C$, the value $c_b(r)$ is uniformly distributed in $S \stackrel{\text{def}}{=} \{c_b(u) \in C : u \in R\}$. Using the conclusion of the previous paragraph, it follows that, for a random $r \in R$ such that $c_b(r) \in S \cap C'$, the value $d_b(r,r')$ (which solves $d \odot c_b(r) = c_b(r')$) is uniformly distributed in some $|S \cap C'|$-subset of $[n]$. Using $\Pr_{r\in R}[c_b(r) \in S \cap C'] \;=\; 1 - O(\epsilon')$, it follows that $d_b(r,r')$ is almost uniformly distributed in $[n]$.

- Fixing an arbitrary $r \in R$ such that $c_b(r) \in C$ and letting r' be uniformly distributed in R, we wish to show that $d_b(r,r')$ is almost uniformly distributed in $[n]$.

  In this case (i.e., $c_b(r) \in C$), the value of $d_b(r,r')$ is determined by $c_b(r')$, which is almost uniformly distributed in $[n]$, such that different values of $c_b(r')$ yield differnt values of $d_b(r,r')$ (since the column $c_b(r) \in C$ contains different values).

Having estabished that, for uniformly distributed $r, r' \in R$, the distribution of $d_b(r,r')$ is almost uniform (over $[n]$) and independent of the distribution of r' (resp., $c_b(r)$), we observe that, for every $a, b \in [n]$, with probability $1 - O(\epsilon')$ over the choice of $r, r' \in$R, it holds that

$$\begin{aligned}
(a \odot (r' \odot d_b(r,r')) \odot c_b(r) &= ((a \odot r') \odot d_b(r,r')) \odot c_b(r) \\
&= (a \odot r') \odot (d_b(r,r') \odot c_b(r)) \\
&= (a \odot r') \odot c_b(r')
\end{aligned}$$

where the first (resp., second) equality uses Eq. (6) and relies on the fact that r' and $d_b(r,r')$ (resp., $d_b(r,r')$ and $c_b(r)$) are almost independently distributed in $[n]$. Hence, we have

$$\mathrm{Pr}_{r,r'\in R}\left[(a\odot(r'\odot d_b(r,r')))\odot c_b(r)=(a\odot r')\odot c_b(r')\right]\ \geq\ 1-O(\epsilon'). \tag{13}$$

Combining Eq. (12) with Eq. (13), the claim follows. ■

Towards the key claims. Using the foregoing claims, we prove the key features of g; that is, that it is close to f, cancellative, and associative. (Recall that $f(a,b)=a\odot b$.)

Claim 4.2.4 (g is $O(\epsilon')$-close to f): $\mathrm{Pr}_{a,b\in[n]}[g(a,b)=f(a,b)]\ =\ 1-O(\epsilon')$. *Furthermore, for every $b\in[n]$, it holds that* $\mathrm{Pr}_{a\in[n]}[g(a,b)=f(a,b)]\ =\ 1-O(\epsilon')$.

Proof: We shall prove the furthermore claim; that is, for every $b\in[n]$, we shall show that $g(\cdot,b)$ is $O(\epsilon')$-close to $f(\cdot,b)$. This is proved by combining the following two facts.

- Using Claim 4.2.3, *for every $a,b\in[n]$, we have* $\mathrm{Pr}_{r\in R}[g(a,b)=f(f(a,r),c_b(r))]\ =\ 1-O(\epsilon')$, *where* $f(r,c_b(r))=b$.
- Using Eq. (8), *for every $b\in[n]$, we have* $\mathrm{Pr}_{a\in[n],r\in R}[f(f(a,r),c_b(r))=f(a,f(r,c_b(r)))]\ >\ 1-\epsilon'$.

Combining these two facts (and recalling that $f(r,c_b(r))=b$), for every $b\in[n]$, we have $\mathrm{Pr}_{a\in[n]}[g(a,b)=f(a,b)]\ =\ 1-O(\epsilon')$. ■

Claim 4.2.5 (g is cancellative):

1. *For every $a,a',b\in[n]$, if $g(a,b)=g(a',b)$, then $a=a'$.*
2. *For every $a,b,b'\in[n]$, if $g(a,b)=g(a,b')$, then $b=b'$.*

Proof: We start with Part 1. By Claim 4.2.3, $\mathrm{Pr}_{r\in R}[(a\odot r)\odot c_b(r)=g(a,b)]=1-O(\epsilon')$, and the same holds for a'. Hence, $g(a,b)=g(a',b)$ implies

$$\mathrm{Pr}_{r\in R}\left[(a\odot r)\odot c_b(r)=(a'\odot r)\odot c_b(r)\right]=1-O(\epsilon'). \tag{14}$$

Recalling that $\mathrm{Pr}_{r\in R}[c_b(r)\in C]=1-O(\epsilon')$ and that in this case there is a unique w such that $w\odot c_b(r)=b$, it follows that

$$\mathrm{Pr}_{r\in R}\left[(a\odot r)=(a'\odot r)\right]=1-O(\epsilon')>\epsilon',$$

which implies that $a=a'$ (since $a\odot r=a'\odot r$ for some $r\in C$).

Turning to Part 2, we first establish it for $a\in R$. Analogously to Eq. (14), we observe that $g(a,b)=g(a,b')$ implies

$$\mathrm{Pr}_{r\in R}\left[(a\odot r)\odot c_b(r)=(a\odot r)\odot c_{b'}(r)\right]=1-O(\epsilon'). \tag{15}$$

Now, if $a\in R$, then $\mathrm{Pr}_{r\in R}[a\odot r\in R]\geq\frac{|R|-(n-|R|)}{|R|}>1-2\epsilon'$ follows (since each element appears once in row a). In this case, $c_b(r)=c_{b'}(r)$ follows for some $r\in R$ (which satisfies $a\odot r\in R$), which implies $r\odot c_b(r)=r\odot c_{b'}(r)$, and it follows that $b=b'$ (since $r\odot c_b(r)=b$ and ditto for b'). We now turn to the general case (of $a\in[n]$), but first prove an auxiliary claim.

Auxliliary claim: For every $x, y \in [n]$, it holds that $\Pr_{r \in R}[r \odot g(c_x(r), y) = g(x, y)] = 1 - O(\epsilon')$.

Proof: Fixing $x, y \in [n]$, we observe that, with probability $1 - O(\epsilon')$ over the choices of $r, s \in R$, it holds that

$$\begin{aligned} r \odot g(c_x(r), y) &= r \odot ((c_x(r) \odot s) \odot c_y(s)) \\ &= (r \odot (c_x(r) \odot s)) \odot c_y(s) \\ &= ((r \odot c_x(r)) \odot s) \odot c_y(s) \\ &= (x \odot s) \odot c_y(s) \\ &= g(x, y) \end{aligned}$$

where the first and last equality are due to Claim 4.2.3, whereas the second and third equalities are guranateed by Eq. (7) (and Claim 4.2.2). This completes the proof of the auxliliary claim.

The general case of Part 2. For any $a, b, b' \in [n]$, if $g(a, b) = g(a, b')$, then (by the auxiliary claim) we have

$$\Pr_{r \in R}\left[r \odot g(c_a(r), b) = r \odot g(c_a(r), b')\right] \;=\; 1 - O(\epsilon').$$

Using the hypothesis that $r \in R$, this implies

$$\Pr_{r \in R}\left[g(c_a(r), b) = g(c_a(r), b')\right] \;=\; 1 - O(\epsilon').$$

Observing that (by Claim 4.2.2) it holds that $\Pr_{r \in R}[c_a(r) \in R] = 1 - O(\epsilon')$ and applying the foregoing special case (which refers to a first operand in R), we infer that $b = b'$. ■

Claim 4.2.6 (g is associative): *For every $a, b, c \in [n]$, it holds that $g(a, g(b, c)) = g(g(a, b), c)$.*

Proof: We prove the claim in two steps, first establishing it only for a random $a \in [n]$, and then inferring that it holds for any $a \in [n]$. Specifically, we first prove that, for every $y, z \in [n]$,

$$\Pr_{r \in R}\left[g(r, g(y, z)) = g(g(r, y), z)\right] = 1 - O(\epsilon'). \tag{16}$$

Fixing $y, z \in [n]$ and using the furthermore part of Claim 4.2.4 (as well as Claim 4.2.3), we observe that with probability $1 - O(\epsilon')$ over the choice of $r, s \in R$, it holds that

$$\begin{aligned} g(r, g(y, z)) &= f(r, g(y, z)) \\ &= f(r, (y \odot s) \odot c_z(s)). \end{aligned}$$

Hence,

$$\Pr_{r,s \in R}\left[g(r, g(y, z)) = r \odot ((y \odot s) \odot c_z(s))\right] = 1 - O(\epsilon'). \tag{17}$$

Next, using Eq. (7) (as well as Claim 4.2.2), with probability $1 - O(\epsilon')$ over the choice of $r, s \in R$, it holds that

$$\begin{aligned} r \odot ((y \odot s) \odot c_z(s)) &= (r \odot (y \odot s)) \odot c_z(s) \\ &= ((r \odot y) \odot s) \odot c_z(s) \\ &= g(f(r, y), z) \\ &= g(g(r, y), z) \end{aligned}$$

where the first and second equalities are guranateed by Eq. (7) (while relying on the randomness of r and $c_z(s)$ (resp., r and s)), whereas the third equality use Claim 4.2.3, and the last equality uses the furthermore part of Claim 4.2.4. Hence,

$$\Pr_{r,s\in R}[r\odot((y\odot s)\odot c_z(s)))=g(g(r,y),z)]=1-O(\epsilon'). \tag{18}$$

Combining Eq. (17)&(18), we establish Eq. (16).

Using Eq. (16), we now establish the actual claim. In particular, we shall apply Eq. (16) four times, while observing that in each of these application the first operand is uniformly (or almost uniformly) distributed in R. Specifically, we observe that, for every $x,y,z\in[n]$, with probability $1-O(\epsilon')$ over the choice of $r\in R$, it holds that

$$\begin{aligned} g(r,g(x,g(y,z))) &= g(g(r,x),g(y,z)) \\ &= g(g(g(r,x),y),z) \\ &= g(g(r,g(x,y)),z) \\ &= g(r,g(g(x,y),z)), \end{aligned}$$

where the second transition we rely on the randomness of $g(r,x)$, which follows from Part 1 of Claim 4.2.5 (i.e., for every x, the mapping $r\mapsto g(r,x)$ is a bijection). Hence,

$$\Pr_{r\in R}[g(r,g(x,g(y,z)))=g(r,g(g(x,y),z))]=1-O(\epsilon'). \tag{19}$$

Using Part 2 of Claim 4.2.5, it follows that $g(x,g(y,z))=g(g(x,y),z)$. ■

Conclusion. Claim 4.2.4 asserts that f is ϵ-close to g, whereas Claims 4.2.5 and 4.2.6 assert that g is cancellative and associative. Hence, g describes a finite cancellative semigroup, which means that g describes a group, and it follows that f is ϵ-close to representing a group. ■

Corollary 4.3 (testing the property of being a group) (Theorem 1.2, restated): *There exists a $(\widetilde{O}(n)/\epsilon)$-time tester for the property of being a group. Furthermore, the tester has one-sided error.*

Theorem 4.4 (testing the property of being an Abelian group): *There exists a $(\widetilde{O}(n)/\epsilon)$-time tester for the property of being an Abelian group. Furthermore, the tester has one-sided error.*

Proof: Our tester first invokes Construction 4.1 and rejects if it has rejected. Otherwise, it checks that $g:[n]\times[n]\to[n]$ as defined in Eq. (3) represents an Abelian group by selecting a random pair of elements in $[n]$ and checking whether they commute (under g). Here we use the well known fact that asserts that if a group is not Abelian, then at least 3/8 of the element pairs do not commute.[6]

[6] An elementrary proof of a weaker lower bound (of 1/4) proceeds as follows. Suppose that more than 3/4 of the pairs are commuting. Then, for every $a\in[n]$,

$$\Pr_{r,s\in[n]}[r\odot(a\odot s)\neq(a\odot s)\odot r]<1/4 \quad\text{and}\quad \Pr_{r,s\in[n]}[a\odot(s\odot r))\neq a\odot(r\odot s)]<1/4$$

which implies $\Pr_{r,s\in[n]}[(r\odot a)\odot s\neq(a\odot r)\odot s]<1/2$. Hence, $\Pr_{r\in[n]}[r\odot a\neq a\odot r]<1/2$ holds for every $a\in[n]$. It follows that, for every $a,b\in[n]$,

$$\Pr_{r\in[n]}[a\odot(b\odot r)\neq(b\odot r)\odot a]<1/2 \quad\text{and}\quad \Pr_{r\in[n]}[b\odot(r\odot a)\neq b\odot(a\odot r)]<1/2$$

which implies $\Pr_{r\in[n]}[(a\odot b)\odot r\neq(b\odot a)\odot r)]<1$. Hence, $a\odot b=b\odot a$ holds for every $a,b\in[n]$.

Note that g can be evaluated, at random, by making $2+n$ queries to f (i.e., $g(x,y) \leftarrow f(f(x,r),c_y(r))$ for a uniformly chosen $r \in [n]$, where $c_y(r)$ is computed by finding c such that $f(r,c) = y$), and recall that if f represents a group then Construction 4.1 always accepts and $g = f$. Hence, our tester always accepts when f represents an Abelian group. Turning to the case that f is ϵ-far from representing an Abelian group, we prove that our tester rejects with probability at least $1/3$ (and obtain the desired tester by error reduction).

We actually prove the counterpositive. Assuming that f is accepted with probability at least $2/3$, it follows (by Theorem 4.2) that f is ϵ-close to representing a group. Furthermore, in that case (by Claim 4.2.3), the random evaluation of g yields its correct value, with probability at least $1 - O(\epsilon)$, whereas g is ϵ-close to f (see Claim 4.2.4) and represents a group (see Claims 4.2.5 and 4.2.6). Lastly, note that the group represented by g must be Abelian, because otherwise our tester would have rejected with probability at least $(1 - O(\epsilon)) \cdot 3/8 > 1/3$. ■

Acknowledgments

We are grateful to Alex Lubotzky for inspiring discussions.

References

[1] N. Alon and Y. Roichman. Random Cayley graphs and expanders. *Random Structures Algorithms*, Vol. 5 (2), pages 271–284, 1994.

[2] E. Ben-Sasson, O. Goldreich, P. Harsha, M. Sudan, and S. Vadhan. Robust PCPs of Proximity, Shorter PCPs, and Applications to Coding. *SIAM Journal on Computing*, Vol. 36 (4), pages 889–974, 2006. Extended abstract in *36th STOC*, 2004.

[3] M. Blum, M. Luby and R. Rubinfeld. Self-Testing/Correcting with Applications to Numerical Problems. *Journal of Computer and System Science*, Vol. 47, No. 3, pages 549–595, 1993.

[4] F. Ergun, S. Kannan, R. Kumar, R. Rubinfeld, and M. Viswanathan. Spot-Checkers. *Journal of Computer and System Science*, Vol. 60 (3), pages 717–751, 2000.

[5] S. Evra, S. Gadot, O. Klein, abd I. Komargodski. Verifying Groups in Linear Time. *ECCC*, TR23-195, 2023.

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

[7] O. Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017.

[8] F. Le Gall and Y. Yoshida. Property testing for cyclic groups and beyond. *J. Comb. Opt.*, Vol. 26, pages 636–654, 2013.

[9] A. McIver and P.M. Neumann. Enumerating finite groups. *Quart. J. Math. Oxford*, Vol. 38, pages 473–488, 1987.

[10] P.M. Neumann. An enumeration theorem for finite groups. *Quart. J. Math. Oxford*, Vol. 20, pages 395–401, 1969.

[11] L. Pyber. Enumerating finite groups of given order. *Ann. of Math.*, Vol. 137 (12), pages 203–220, 1993.

[12] S. Rajagopalan and L.J. Schulman. Verification of Identities. *SIAM Journal on Computing*, Vol. 29 (4), pages 1155–1163, 2000.

[13] G. Rothblum, S. Vadhan, and A. Wigderson. Interactive Proofs of Proximity: Delegating Computation in Sublinear Time. In *45th ACM Symposium on the Theory of Computing*, pages 793–802, 2013.

[14] R. Rubinfeld and M. Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal on Computing*, Vol. 25 (2), pages 252–271, 1996.