Verification of Identities*

Sridhar Rajagopalan†  Leonard J. Schulman‡

March 13, 1999

Abstract

We provide an $O(n^2 \log \frac{1}{\delta})$ time randomized algorithm to check whether a
given operation $\circ : S \times S \rightarrow S$ is associative (where $n = |S|$ and $\delta > 0$ is the
error probability required of the algorithm.) We prove that (for any constant $\delta$)
this performance is optimal up to a constant factor even in case the operation
is “cancellative”. No sub-$n^3$ time algorithm was previously known for this task.

More generally we give an $O(n^c)$ time randomized algorithm to check whether
a collection of $c$-ary operations satisfy any given “read-once” identity.

Key words and phrases: “Computer Aided Verification”, “Randomized Al-
gorithms”.

AMS subject classification: primary 68Q25, secondary 20N02, 20N05.

Abbreviated title: Verification of Identities.

---

†IBM Almaden Research Center. sridhar@almaden.ibm.com This research was conducted while
the author was a post-doctoral fellow at DIMACS and Princeton University.
‡College of Computing, Georgia Institute of Technology, Atlanta GA 30332-0280.
schulman@cc.gatech.edu
1 Introduction

Let a set $S$ be given, along with a binary operation $\circ : S \times S \to S$. In this paper, we consider the complexity of checking whether $\circ$ is associative. We provide an algorithm for this problem, and then show how our method extends without essential modification to provide a means of checking general “read-once” identities.

Throughout, let $n = |S|$. We provide a randomized, one-sided error algorithm which in time $O(n^2 \log \frac{1}{\delta})$ computes whether the operation $\circ$ is associative. If $\circ$ is associative then the algorithm does not err in its response, while if $\circ$ is not associative then the error probability is bounded by $\delta$. It is assumed that the operation $\circ$ is computable in unit time; the same assumption is made throughout the paper, and all runtimes scale linearly in the actual time required for computation of the given operations.

The techniques we develop are more general and can be used to check whether collections of operators satisfy various identities. For instance: given a finite state machine, $\delta : S \times \Sigma \to S$, verify that the machine is “asynchronous”, namely $\delta(\delta(s,a),b) = \delta(\delta(s,b),a)$ for all $s \in S$ and $a,b \in \Sigma$. This test is sometimes called the diamond test.

In case the identity is not satisfied, our method also provides a witness (e.g., for associativity, a triple $a,b,c$ such that $(a \circ b) \circ c \neq a \circ (b \circ c)$) with only a logarithmic slowdown in the time complexity.

Associativity: Prior Work

Prior to our work, except in special cases, no method for verifying associativity was known that was better than the naive $O(n^3)$-time algorithm of examining whether $a \circ (b \circ c) = (a \circ b) \circ c$ for all $a,b,c \in S$.

F. W. Light observed in 1949 (see [2]) that, if $R \subseteq S$ is a set of generators of $S$, i.e. a collection such that every element of $S$ is representable as a product of elements of $R$, then it suffices to test all triples $a,b,c$ in which $b$ is an element of $R$. This however does not result in a sub-$n^3$ algorithm for this problem, for two reasons: first, the operation may require a large set of generators, as much as $n$ (and there are such examples which are associative, so the algorithm would in fact examine all triples) — see example 8.1; second, even if the operation does have a small set of generators, we are not aware of any rapid method for obtaining a set of generators of (even close to) minimal size.

Associativity: Cancellative Operations

An operation $\circ$ on a finite set $S$ is left (resp. right) cancellative if for every $a$ and $b$, there is an $x$ (which must be unique) such that $x \circ a = b$ (resp. $a \circ x = b$).
A cancellative operation is both left and right cancellative. In other words, every row and column of the table for the operation is a permutation of $S$.

In the special case that the operation is cancellative, Light's observation is quite useful. Only $O(n^2)$ time is required to test whether $\circ$ is cancellative. Moreover, we will show in section 5 that, given a cancellative multiplication table, one can deterministically compute a set of generators of size $\lceil \log_2 n \rceil + 1$ in $O(n^2)$ time. Thus, Light's observation results in an $O(n^2 \log n)$ deterministic algorithm for verifying associativity in the cancellative case. Consequently there is a deterministic $O(n^2 \log n)$ time algorithm to test whether a given set and operation form a group. (Alternatively, a random set of elements may be chosen; a set of $\log_2 n$ elements will be a set of generators with probability at least $1 - \exp(-c)$. This implies a randomized algorithm analogous to the deterministic one.)

We show in section 5 that any cancellative nonassociative operation has at least $n - 2$ nonassociative triples. Therefore, the procedure of checking $O(n^2 \log \frac{n}{2})$ random triples will succeed with probability $1 - \delta$, providing essentially the same guarantee as the randomized version of Light's method, while being even simpler. (Both run in time $O(n^2 \log \frac{n}{2})$.)

However, just as Light's observation fails to be of use for general operations, because small sets of generators may not exist or may be hard to find, the random sampling approach also fails to be of use for general operations, in this case because for every $n \geq 3$ there exists an operation (noncancellative, of course) with just one nonassociative triple. See example 8.2.

**Associativity: Lower Bound**

We show that any randomized algorithm requires time $\Omega(n^2)$ to verify associativity, even in the cancellative case. Thus the runtime of our randomized algorithm as a function of input size is tight up to a constant factor.

*Comment on terminology:* The pair $(S, \circ)$ is known in the Algebra literature as a groupoid [1, 2], a term which is unfortunately also used elsewhere in that literature to mean something entirely different [4, 6].

When $\circ$ is cancellative $(S, \circ)$ is referred to in [1] as a quasigroup.

## 2 Algorithm for Checking Associativity

We define the structure $S/2 = (\mathbb{Z}/2)[S]$ as follows. The elements of $S/2$ are sums of elements of $S$, with coefficients in $\mathbb{Z}/2$ (i.e. $\sum_{s \in S} \alpha_s s$ for $\alpha_s \in \mathbb{Z}/2$). $S/2$ is equipped with the following operations:

1. Addition: $\sum_{s} \alpha_s s + \sum_{s} \beta_s s = \sum_{s} (\alpha_s + \beta_s) s$. 


2. Scalar multiplication: \( \beta \sum_s \alpha_s s = \sum_s (\beta \alpha_s) s \) and \( (\sum_s \alpha_s s) \beta = \sum_s (\beta \alpha_s) s \) for \( \beta \in \mathbb{Z}/2 \).

3. The operation \( \circ: (\sum_s \alpha_s s) \circ (\sum_s \beta_s s) = \sum_s \alpha_s \beta_s (r \circ s) \)

Thus \( S/2 \) is what would be known as the “group algebra” of \((S, \circ)\) over \( \mathbb{Z}/2 \), were \((S, \circ)\) a group. (In the present circumstance “groupoid algebra” may be an appropriate term.)

Henceforth, we will denote members of \( S/2 \) by bold letters, and coefficients by the corresponding Greek lowercase: e.g., \( \mathbf{a} = \sum_s \alpha_s s \).

Our method for checking associativity in \((S, \circ)\) is to repeat the following, \( O(\log \frac{1}{\delta}) \) times:

- **Check the associative identity for three random elements of \( S/2 \).**

  In other words, select random (uniformly i.i.d.) \( \mathbf{a}, \mathbf{b}, \mathbf{c} \in S/2 \), and check that \( (\mathbf{a} \circ \mathbf{b}) \circ \mathbf{c} = \mathbf{a} \circ (\mathbf{b} \circ \mathbf{c}) \).

**Theorem 2.1** The algorithm above, running in time \( O(n^2 \log \frac{1}{\delta}) \), will determine whether \( \circ \) is associative, with error probability at most \( \delta \) if it is not associative, and no error if it is associative.

First note that checking that \( (\mathbf{a} \circ \mathbf{b}) \circ \mathbf{c} = \mathbf{a} \circ (\mathbf{b} \circ \mathbf{c}) \) can be done in time \( O(n^2) \). For, \( \mathbf{a} \circ \mathbf{b} \) can be computed by brute force in time \( O(n^2) \); moreover the result of this computation is another vector \( \mathbf{d} \in S/2 \), so the subsequent computation \( \mathbf{d} \circ \mathbf{c} \) can again be computed by brute force in time \( O(n^2) \). For the same reason, \( \mathbf{a} \circ (\mathbf{b} \circ \mathbf{c}) \) can be computed in time \( O(n^2) \); and finally, the two sides can be compared in time \( O(n) \).

We now show that a single run of the above process succeeds in detecting nonassociativity with probability at least \( 1/8 \); by repeating the process \( \log \frac{1}{\delta} \) times, the dependence of the running time on \( \delta \) follows.

It is easily verified that \( \circ \) is associative on \( S \) if and only if it is associative on \( S/2 \) as well (for a proof see lemma 3.1).

Let \( S^k \) denote the \( k \)-wise Cartesian product of \( S \) with itself. A minor of \( S^k \) is a set \( A_1 \times A_2 \times \cdots \times A_k \subseteq S^k \) (where each \( A_i \) is a subset of \( S \)). If \( H \) is a commutative group (expressed additively), \( g \) is a function \( g : S^k \to H \), and \( T \) is a subset of \( S^k \), then define \( g(T) = \sum_{t \in T} g(t) \).

**Lemma 2.1** Let \( H \) be any commutative group, and let \( g : S^k \to H \) be a nonzero function. Then the fraction of minors \( T \) of \( S^k \) for which \( g(T) \neq 0 \) is at least \( 2^{-k} \).

**Proof:** Fix \( \tau = (t_1, \ldots, t_k) \) such that \( g(\tau) \neq 0 \). Let \( T_1 \subseteq S - \{t_1\}, \ldots, T_k \subseteq S - \{t_k\} \). For \( 1 \leq i \leq k \) define \( T_i^0 = T_i \) and \( T_i^1 = T_i \cup \{t_i\} \). Observe that \( \{\tau\} = \cap_{i=1}^k ((T_1^0 \times \cdots \times T_i^0 \times \cdots \times T_k^0) - (T_1^1 \times \cdots \times T_i^1 \times \cdots \times T_k^1)) \) and that for \( \varepsilon = (\varepsilon_1, \ldots, \varepsilon_k) \)
(with $\varepsilon_i \in \{0, 1\} \forall i$), $T_{\varepsilon_1}^i \times \ldots \times T_{\varepsilon_k}^{i_k} = \bigcap_i T^1_i \times \ldots \times T^1_{i-1} \times T^1_i \times \ldots \times T^1_{i+1} \times \ldots \times T^1_k$.

Hence inclusion-exclusion gives

$$g(\tau) = (-1)^k \sum_i (-1)^{\varepsilon_i} g(T_{\varepsilon_1}^i \times \ldots \times T_{\varepsilon_k}^{i_k}).$$

Since $g(\tau) \neq 0$, there exists a minor $T_{\varepsilon_1}^i \times \ldots \times T_{\varepsilon_k}^{i_k}$ such that $g(T_{\varepsilon_1}^i \times \ldots \times T_{\varepsilon_k}^{i_k}) \neq 0$.

Thus the minors of $S^k$ are partitioned into sets of $2^k$ minors, each identified by some $T_1, \ldots, T_k$, and in each such set there is at least one minor $T$ for which $g(T)$ is nonzero.

The error probability claimed in the theorem follows from the lemma by defining:

$$g : S^3 \to S/2$$

$$g(i, j, k) = (i \circ j) \circ k - i \circ (j \circ k).$$

In the last line we have abbreviated notation by using the natural embedding $S \to S/2$ sending $s \in S$ to $1s + \sum_{r \neq s} 0r$. \hfill \Box

Remark: There is a similarity to the argument showing that Freivalds’ [3] checker for matrix multiplication succeeds with probability at least 1/2. In that case, given matrices $A, B$ and $C$, a random vector $v$ is multiplied by $AB - C$; if $AB \neq C$ then there is some vector $u$ such that $u(AB - C) \neq 0$, and to every vector $w$ such that $w(AB - C) = 0$, we associate the vector $w + u$, and note that $(w + u)(AB - C) \neq 0$.

3 Generalizations

We have presented our method in the context of testing for associativity, that being a concrete and significant case, and the original problem we considered. However the method can be extended mutatis mutandis to more general situations.

Let $\{D_i\}_{i=1}^c$ be a collection of finite sets. If $\circ$ is a function on domain $\prod_{i=1}^c D_i$, we say that $c$ is the degree of $\circ$. An identity is an equation involving one or more operations $\circ_i$, and which is required to hold for all instantiations of the variables, e.g. $\forall a, b, d, e \in D_1, c \in D_2$

$$\circ_1(\circ_2(a, b), c, \circ_2(d, e)) = \circ_1(\circ_2(e, d), c, \circ_2(b, a))$$

or in the case of associativity,

$$\circ(\circ(a, b), c) = \circ(a, \circ(b, c)) \forall a, b, c \in S.$$
Note that each side of the equation may be viewed as a formula (in the circuit sense); we let $\ell$ be the total number of operations (internal nodes) occurring in the formulas. Thus $\ell = 6$ in the first example, and $\ell = 4$ for associativity. We let $k$ be the total number of distinct variables occurring in the identity; thus $k = 5$ in the first example, and $k = 3$ for associativity.

A “read-once” identity is one in which every variable occurs exactly once on each side of the equation.

We have the following theorem:

**Theorem 3.1** Let $\{D_i\}_{i=1}^k$ be finite sets, and let $\{\sigma_j\}$ be a collection of operators, defined on various products of these sets; let the arguments to operator $\sigma_j$ be drawn from the sets $D_{\sigma(j,1)}, ..., D_{\sigma(j,|\sigma_j|)}$ where $c_j$ is the degree of $\sigma_j$. Let $M = \max_j \prod_{i=1}^{c_j} |D_{\sigma(j,i)}|$. A read-once identity with $\ell$ operations on $k$ variables $\{x_i\}_{i=1}^k$ (each ranging in the corresponding $D_i$) can be verified in time $O(M \log(1/\delta) 2^{4\ell})$ with failure probability at most $\delta$.

For comparison, suppose for simplicity that all the sets are of size $n$, this runtime is $O(n^{\max\{c_j\}} \log(1/\delta) 2^{4\ell})$ whereas the naive method requires time $O((kn)^*)$. The key gain is that the exponent of $n$ is the maximum degree of the operations, rather than the number of variables in the identity.

This theorem relies upon the following:

**Lemma 3.1** A read-once identity holds in $S$ if and only if it holds in $S/2$.

**Proof:** If the identity holds in $S/2$ then in particular it holds when each variable in the equation is a “singleton” element of the algebra, i.e. an element which has coefficient 1 on some element of $S$, and coefficient 0 elsewhere.

For the converse, note that each side of the identity can be expanded as a summation, over $(x_1, ..., x_k) \in D_1 \times ... \times D_k$, of terms each in exactly the same form as that side of the identity. Each corresponding pair of terms are equal in $S$, hence the summations are equal. \qed

Some examples:

**Multiple Domains** Let a finite state machine $(S, \Sigma, \delta)$ be given. Here $S$ is the set of states of the machine (let $|S| = n$), $\Sigma$ is the tape alphabet (let $|\Sigma| = m$), and $\delta : S \times \Sigma \to S$ is the transition function. Then the degree of the operator $\delta$ is 2, and it is possible to check in time $O(nm)$ (rather than the obvious $O(nm^2)$) that the operation of the machine is not dependent on the order of the input. (In other words, whether $\forall s \in S, a, b \in \Sigma, \delta(\delta(s, a), b) = \delta(\delta(s, b), a)$.) Testing for this property of the machine is sometimes called the diamond test in the CAD literature. We do this by extending $\mathbb{Z}/2$ with both $S$ and $\Sigma$. Then, we can extend $\delta$ implicitly to $\delta : (\mathbb{Z}/2)[S] \times (\mathbb{Z}/2)[\Sigma] \to (\mathbb{Z}/2)[S]$. We then verify the identity for a random $s, a$ and $b$. 

6
More generally, if $\circ$ is an operator from $X \times Y \to Z$, it can be naturally extended to an operator from $(\mathbb{Z}/2)[X] \times (\mathbb{Z}/2)[Y] \to (\mathbb{Z}/2)[Z]$, i.e.

$$(\sum_x \alpha_x x) \circ (\sum_y \beta_y y) = \sum_{xy} \alpha_x \beta_y \circ y = \sum_z \gamma_z z$$

where $\gamma_z = \sum_{x \circ y = z} \alpha_x \beta_y$. The algorithm extends to this setting. (In particular, if the identity is false, the probability of detecting this in one round is at least $1/8$.)

**Multiple Operations** Consider two binary operations $\cup$ and $\cap$, and a unary operation $'$ over $S$ (representing perhaps some kind of complementation). We wish to verify that for every $a, b, c \in S$, $(a \cap (b \cup c))' = a' \cup (b' \cap c')$. This is done in quadratic time by verifying the identity at random points over $(\mathbb{Z}/2)[S]$.

## 4 Witness Identification

The above method disproves identities without producing an explicit counterexample. However, if desired, it can easily be used in order to produce such a counterexample. (For example, for the purpose of debugging.) We illustrate this in the case of associativity.

Let $a, b, c$ be a nonassociative triple in $S/2$. Let $A$, $B$ and $C$ be the sets indexed by $a, b$ and $c$ (i.e. those elements occurring with coefficient 1). We know that there is a nonassociative triple in $A \times B \times C$. Because of theorem 3.1, we can choose a new triple to be tested by the algorithm as follows: $a', b', c'$ are i.i.d., with $a'_i = 0$ if $a_i = 0$, and otherwise $a'_i$ is uniformly selected in $\{0, 1\}$. $b'$ and $c'$ are chosen similarly. The probability that $a', b', c'$ is a nonassociative triple is at least $1/8$. We repeat this process until a nonassociative triple $a', b', c'$ is found; then continue in like manner. Each of $A$, $B$ and $C$ approximately halve in each successful round (analysis below), hence the number of rounds will be $O(\log n)$ with high probability. Thus, with a logarithmic increase in cost, we can pinpoint a nonassociative triple in $S$. Below we carry out the analysis up to the point that each of $A$, $B$ and $C$ has been reduced to size $n^{2/3}$ or less; at that point all triples may be examined, in time $O(n^2)$.

This argument, like theorem 2.1, generalizes in all the ways described in the previous section.

**Theorem 4.1** Beginning with sets $A, B, C$ containing some nonassociative triple, the probability that more than $(1 + \frac{1}{\log^{1/3} n})\frac{\log n}{3} \log_2 n$ rounds of the above process are required to reduce all sets to size at most $n^{2/3}$ is at most $\exp(-\Omega(\log^{1/3} n))$.

**Proof:** After one round we have new sets $A', B', C'$; if the round was successful (i.e. $a', b', c'$ was a nonassociative triple) these will usually be strict subsets
of $A$, $B$ and $C$, while if the round was unsuccessful then $A' = A$, $B' = B$ and $C' = C$.

We are interested in the distribution of the random variables $|A'|/|A|, |B'|/|B|$ and $|C'|/|C|$ (in each round). With probability at most $7/8$ the round is unsuccessful and the ratios equal 1.

We have shown earlier that for any $a \in A, b \in B, c \in C$ for which $(a, b, c)$ is nonassociative, and for any $R \subseteq A - \{a\}, S \subseteq B - \{b\}, T \subseteq C - \{c\}$, at least one of the eight triples $(R, R \cup \{a\}) \times (S, S \cup \{b\}) \times (T, T \cup \{c\})$ is a successful choice. In the worst case from the point of view of the sizes of $A', B'$, and $C'$, the successful triple is always $(R \cup \{a\}) \times (S \cup \{b\}) \times (T \cup \{c\})$. Therefore as an upper bound on our analysis we can suppose that the distribution of the sizes of $A', B'$ and $C'$ in successful rounds are independent, and binomially distributed in the ranges $[1, |A|], [1, |B|], [1, |C|]$.

We focus on just one of these sequences of ratios, e.g. $|A'|/|A|$; the union bound accounts for a factor of 3 in the probability of failure. The probability that less than fraction $1 - \frac{1}{2\log^{1/3} n}$ of the $(1 + \frac{1}{\log^{1/3} n})\log_2 n$ rounds, are successful, is at most $\exp(-\Omega(n^{1/3} \log n))$. The probability that there exists any successful round in which $|A'|/|A| > (1 + \frac{1}{\log n})/2$ is at most $\log n \exp(-\Omega(n^{2/3} / \log^2 n))$. Therefore with probability at least $1 - \exp(-\Omega(n^{1/3} \log n))$ there are at least $\frac{1}{5}(1 + \frac{1}{\log^{1/3} n})\log_2 n$ rounds in which $|A'|/|A| \leq (1 + \frac{1}{\log n})/2$. The size of the set remaining after these rounds is at most $n((1 + \frac{1}{\log n})/2)^{1/5(1 + \frac{1}{\log^{1/3} n})\log_2 n} \leq n^{2/3} \exp(-\Omega(n^{2/3} \log n))$.

5 The cancellative case

We claim the following concerning this case:

**Theorem 5.1** Let $\circ$ be cancellative.

1. If $\circ$ is nonassociative then it has at least $n - 2$ nonassociative triples.

2. It is possible to compute a generating set of size $[\log_2 n] + 1$ in quadratic time.

The first observation implies that picking a random triple and checking it for associativity works essentially optimally for cancellative operations. The second observation implies an $O(n^2 \log n)$ deterministic algorithm for verifying associativity for cancellative binary operations by taking into account Light’s observation mentioned in the introduction.

**Proof:**

1. Let $(a, b, c)$ be nonassociative and let $a = a' \circ a''$. Consider the following cycle: $((a' \circ a'') \circ (b \circ c)) = ((a' \circ a'') \circ b) \circ c = (a' \circ (a'' \circ b)) \circ c = a' \circ (a'' \circ (b \circ c)) = (a' \circ a'') \circ (b \circ c)$. 

8
Each of these equalities is an application of the associative identity, and since the first fails, one of the other four must fail as well. In other words if \((a, b, c)\) is nonassociative then at least one of the following must be nonassociative:

(i) \((a', a'', b)\)
(ii) \((a', a'' \circ b, c)\)
(iii) \((a'', b, c)\)
(iv) \((a', a'', b \circ c)\)

Since \(\circ\) is cancellative, \(a\) can be written as \(a' \circ a''\) in \(n\) different ways. For each of these, associativity fails in at least one of the four categories above. Thus, there is a category for which there are \(n/4\) failures. Each category identifies either \(a'\) or \(a''\), so there can be no duplications among the nonassociative triples listed for that category.

To improve to a bound of \(n - 2\) note that there must be a nonassociative triple in which not all three elements are equal. This follows once we know that there are at least two triples in one of the above categories. Now we suppose without loss of generality that there is a nonassociative triple \((a, b, c)\) for which \(b \neq c\); if this is not the case then there is one for which \(a \neq b\), and the remainder of the argument can be “reflected” accordingly.

For each of the \(n\) choices of \(a', a''\) such that \(a' \circ a'' = a\), fix a category (i-v) in which the triple is nonassociative. Observe that among the triples listed in categories (i),(ii) and (iv) there can be no duplication, since \(a'\) is identified as the first element in each of those triples. The question remains, how much duplication there can be between list (iii) and lists (i),(ii) and (iv).

Since \(b \neq c\), there can be no duplication between lists (i) and (iii).

If there is a duplication between a triple \((a_2', a_2'' \circ b, c)\) in list (ii) and a triple \((a_2'', b, c)\) in list (iii), then the value of \(a_2''\) is implied by the value of \(a_2'' \circ b\); this in turn implies the value of \(a_2'\), which must be equal to \(a_2''\), and this in turn implies the value of \(a_2'\). Hence there can be only one triple common to lists (ii) and (iii).

A common triple to lists (iii) and (iv) is possible only if \(b \circ c = c\); in this case if triples \((a_2', b, c)\) in list (iii) and \((a_2', a_4'', b \circ c)\) in list (iv) are equal, then \(a_4''\) must equal \(a_4''\). This which implies the value of \(a_4'\), which in turn is equal to \(a_4'\), and this implies the value of \(a_4'\). Hence there can be only one triple common to (iii) and (iv).

Hence there are at least \(n - 2\) nonassociative triples.

The construction in section 6 will show that this bound is tight to within a constant factor.
2. Let $A \subseteq S$ be arbitrary. Denote by $\langle A \rangle$ the closure of $A$ under $\circ$. Thus a generating set $G$ is one such that $\langle G \rangle = S$. Let $A$ be any set such that $\langle A \rangle \neq S$ and let $b \in S - \langle A \rangle$. Then, $|\langle A \cup \{b\} \rangle| \geq 2|\langle A \rangle|$. For, right cancellativity implies that the elements of $b \circ \langle A \rangle$ are all distinct; left cancellativity and the fact that $\langle A \rangle$ is closed imply that each element of $b \circ \langle A \rangle$ is outside of $\langle A \rangle$.

This implies the existence of a size $\lfloor \log_2 n \rfloor + 1$ generating set. Moreover, observe that we can keep track of the closure (in a greedy manner) in time $O(n^2)$.

If an operation $\circ$ on a finite set $S$ is both cancellative and associative, then $(S, \circ)$ is a group. Hence the above process of first testing for cancellativity, and then for the associative identity, yields:

**Theorem 5.2** There is a deterministic $O(n^2 \log n)$ time algorithm to test whether $(S, \circ)$ is a group.

\[ \square \]

### 6 Lower Bound for Verifying Associativity

An $\Omega(n^2)$ lower bound for checking for associativity is immediate: if some product of elements is not examined, it may be changed to destroy associativity. Indeed, this points out that a sub-$O(n^2)$ algorithm can be foiled by simply changing the product of a random pair of elements. Using Yao’s lemma [7], the $\Omega(n^2)$ lower bound holds for randomized algorithms (even those tolerating constant error probability) as well.

With some more attention, we obtain an $\Omega(n^2)$ lower bound which holds even in case $\circ$ is assumed to be cancellative. (In the cancellative case, as we have pointed out, there are simpler quadratic time algorithms than ours to check associativity.) Again the method is to change just a few values of $\circ$; and again Yao’s lemma implies that the lower bound holds for randomized algorithms.

The argument is as follows. Suppose the algorithm is deterministic. Let $S$ be the hypercube $(\mathbb{Z}/2)^m$, and let $\circ$ be vector addition. Suppose there exist $a, b, c \in S$, with $a \neq 0^m$, such that the entries $b \circ c$, $b \circ (a + c)$, $(a + b) \circ c$, $(a + b) \circ (a + c)$ are not examined by the algorithm. Then the behavior of the algorithm on $\circ$ is indistinguishable from its behavior on the operation $\circ'$ which we obtain by modifying the following entries of $\circ$:

1. $b \circ' c = a + b + c$
2. $b \circ' (a + c) = b + c$
3. \((a + b) \circ' c = b + c\)

4. \((a + b) \circ' (a + c) = a + b + c\)

Observe that \(\circ'\) is still cancellative, but it is not associative because

\[ (c \circ' (a + b)) \circ' c = (a + b + c) \circ' c = a + b \]

while

\[ c \circ' ((a + b) \circ' c) = c \circ' (b + c) = b. \]

Now, for any fixed \(a\), there is a set \(R \subseteq S\) of size \(|R| = |S|/2\) such that for all \(r, r' \in R, r + a \neq r'\). Letting \(b\) and \(c\) range over \(R\), we obtain \(|S|^2/4\) disjoint quadruples \(\{b, c\}, \{b, a + c\}, \{a + b, c\}, \{a + b, a + c\}\) in \(S \times S\). If the algorithm does not examine at least one value of \(\circ\) in each quadruple then it cannot distinguish \(\circ\) from \(\circ'\). Hence the algorithm must perform at least \(n^2/4\) operations.

We now note that theorem 5.1(1) is tight up to a constant factor. For, if we pick \(a, b, c\) independently, uniformly at random, then since \(\circ'\) is cancellative, each one of the four pairs \(\{a, b\}, \{a \circ' b, c\}, \{b, c\}\) and \(\{a, b \circ' c\}\) is uniformly distributed in \(S \times S\). Therefore the probability that \((a, b, c)\) is a nonassociative triple for \(\circ'\) is at most \(16/n^2\); in other words, the number of nonassociative triples is at most \(16n\).

7 Grigni’s modification

M. Grigni has pointed out that for large \(k\) (the number of variables in the identity) it is useful to run our algorithm using \(S/p = (\mathbb{Z}/p)[S]\) for prime \(p\) such that \(p > k\) (in place of \(S/2\)), where \(k\) is the number of variables occurring in the identity. At the modest price of keeping track of larger coefficients, and using a little more randomness, the probability of detecting failure of the identity in one run of the process is now at least \(1 - k/p\), rather than \(2^{-k}\). This is for the following reason: the result of the computation in our algorithm is a pair of elements of \(S/p\), one corresponding to the LHS of the identity being checked, and another corresponding to the RHS. Subtracting one from the other, the algorithm obtains an element \(\sum \omega_s, s \in S/p\), and reports nonassociativity if any \(\omega_s\) is nonzero. Each such \(\omega_s\) is defined by a polynomial over \(\mathbb{Z}/p\) in the (random) variables (of which each is random in \(\mathbb{Z}/p\)) used in the algorithm. The degree of each of these polynomials is most \(k\), and if the operation is not associative then at least one polynomial is nonzero. By a lemma of J. T. Schwartz [5], the fraction of assignments on which a nonzero polynomial is 0 is at most \(k/p\).
8 Examples

Example 8.1 Let $S = \{1, 2, \cdots n\}$, with $x \circ y = x$ for every $(x, y)$. The only set of generators is $S$. Note also that $\circ$ is associative.

Example 8.2 Let $S = \{1, 2, \cdots n\}$, with $n$ at least 3. Let $1 \circ 2 = 2$. Let $x \circ y = 3$ for every $(x, y) \neq (1, 2)$. The only non-associative triple is $(1, 1, 2)$.

For $n = 2$ any nonassociative operation has at least two nonassociative triples.

9 Open Questions

While a nonassociative operation has a short witness for this property — and we are able, in fact, to efficiently find such a witness — we do not know of any short witness for associativity, or any sub-$n^3$ time (randomized or deterministic) algorithm which, if the given operation is associative, proves this fact. (An exception is cancellative operations which, from theorem 5.2, we know to have associativity witnesses of length $O(n^2 \log n)$.)

It remains to make any progress on verification of identities which are not read-once. A key example is the “distributive” identity $a \cap (b \cup c) = (a \cap b) \cup (a \cap c)$; it is not known whether this can be verified in less than cubic time.

Finally, we recall the interesting question encountered in the context of Light’s observation, as to whether there is an efficient algorithm which, given a set $S$ and a binary operation $\circ$ on $S$, finds a set of generators for $S$ of optimal or nearly optimal size.

10 Acknowledgments

We wish to acknowledge helpful conversations with Manuel Blum, Michelangelo Grigni, Sampath Kannan and Thomas Wilke. We also wish to thank Kousha Etessami for first bringing the question to our attention.

References


