Associativity of the Commutator Operation in Groups

Fernando Guzmán

Department of Mathematical Sciences, Binghamton University, Binghamton N.Y., 13902-6000

E-mail address: `fer@math.binghamton.edu`

URL: `http://math.binghamton.edu/fer`

Abstract. The study of associativity of the commutator operation in groups goes back to the work of F. W. Levi in 1942. In the 1960's Richard J. Thompson created a group F whose elements are representatives of the generalized associative law for an arbitrary binary operation. In 2006, Geoghegan and Guzmán proved that a group G is solvable if and only if the commutator operation in G eventually satisfies ALL instances of the associative law, and also showed that many non-solvable groups do not satisfy any instance of the generalized associative law. We will address the question: Is there a non-solvable group which satisfies SOME instance of the generalized associative law? For finite groups, we prove that the answer is no.
AMS Classification 2000: Primary: 20D05 ; Secondary: 20F16, 20N02, 20F38.

1. Introduction

In 1942 F. W. Levi [3] proved that the commutator operation in a group is associative if and only if the group is nilpotent of class ≤ 2. That, in a sense, settled the question of associativity of the commutator operation in groups.

In the 1960's Richard Thompson [4], studying the logical connections between the three-variable associative law and the generalized associative law, created the group F whose elements can be thought of as (equivalence classes of) instances of the generalized associative law. More details on this below. Thompson's notes were circulated but never published. In 1996, Cannon, Floyd and Parry [1] published what became the main introductory reference to Thompson's groups. We refer the reader to that paper for full details.

In 2006 Geoghegan and Guzmán [2] used subgroups of Thompson's group F to "*measure*" the associativity of a binary operation. That is, a non-associative binary operation may still satisfy some instances of the generalized associative law. In particular, for a group which is not nilpotent of class ≤ 2, its commutator operation may satisfy some instances of the generalized associative law, and this reopens the issue of the associativity of the commutator operation in groups, in this generalized sense.

For any magma S, the set $\mathrm{Assoc}(S)$ of instances of the generalized associative law that are (eventually) satisfied by S, forms a subgroup of F. It is proved in [2] that when we take S to be a group G with its commutator operation $[\cdot,\cdot]$, G is a solvable group if and only if $\mathrm{Assoc}(G) = F$. Moreover, for many non-solvable groups G it was shown that $\mathrm{Assoc}(G) = 1$. That opened the question of whether there is a non-solvable group G for which

$$1 \lneqq \mathrm{Assoc}(G) \lneqq F$$

i.e. such that the commutator operation of G satisfies some instance of the generalized associative law.

In this paper, we answer the question for finite groups. We prove:

Theorem. *If the commutator operation in a finite group G satisfies some instance of the generalized associative law, then G is solvable.*

2. **The Generalized Associative Law and Thompson's group** F

Given a magma $(S,\star)$, i.e. a non-empty set S with a binary operation $\star$, the *Generalized Associative Law* states that for any $n \geq 1$, any two parenthizations of the expression

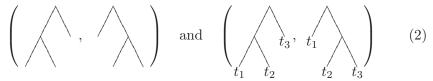$$X_1 \star X_2 \star \cdots \star X_n \tag{1}$$

evaluate the same in S. An *instance* of the Generalized Associative Law consists of a choice of $n \geq 1$, and two parenthizations for (1). When the two parenthizations are different we may emphasize this fact by referring to it as a *non-trivial* instance of the Generalized Associative Law. The first non-trivial instance of the Generalized Associative Law occurs with $n = 3$, and it is the usual (three-variable) Associative Law.

There are several alternative ways to construct/look at Thompson's group F. Here we sketch one of them, referring the reader to [1] for full details.

We can encode a parenthezation of a product of n factors, using a binary tree with n leaves. Each instance of the generalized associative law, is then encoded with a pair of binary trees having the same number of leaves. The set $\mathcal{T}$ of all pairs of binary trees, with both trees in each pair having

the same number of leaves, is the beginning of one of the constructions of Thompson's group F.

In order to define a binary operation on pairs of trees, an equivalence relation is introduced, by which two pairs of trees are declared to be equivalent when one pair is obtained from the other by doing a "*common expansion*" on both trees. For example, if t_1, t_2, t_3 are any binary trees then the following two pairs of trees

$$\left(\bigwedge , \bigwedge \right) \quad \text{and} \quad \left(\bigwedge_{t_1 \ t_2} t_3 , \ t_1 \bigwedge_{t_2 \ t_3} \right) \tag{2}$$

are equivalent, since the second pair is obtained from the first pair by expanding both trees the same way, i.e. by "*hanging*" t_1 from the first leaf, t_2 from the second leaf and t_3 from the third leaf. Note that the left pair encodes the three-variable associative law. In general, if (s, t) is a pair of binary trees with n leaves each, and $t_1, \ldots, t_n$ are n binary trees, then $(s(t_1, \ldots, t_n), t(t_1, \ldots, t_n))$ is a pair of trees with k leaves each, where $k = k_1 + \cdots + k_n$ and k_i denotes the number of leaves of t_i. We call the pair $(s(t_1, \ldots, t_n), t(t_1, \ldots, t_n))$ a *common expansion* of the pair (s, t). We consider the equivalence relation on $\mathcal{T}$ generated by declaring each pair (s, t) equivalent to each of its common expansions $(s(t_1, \ldots, t_n), t(t_1, \ldots, t_n))$. The equivalence class of (s, t) is denoted by $\langle s, t \rangle$. It can be shown that in each equivalence class, there is exactly one representative pair which is not a common expansion of any other pair. We call such pair of trees a *reduced pair*, and it can be shown that each pair is a common expansion of the reduced pair representative in its class. A *free caret* in a binary tree, is a caret on two consecutive leaves. The first tree in the left pair of (2) has a free caret on the first and second leaves. The second tree of that pair has a free caret on the second a third leaves. When the two trees in a pair both have a free caret with the same two leaves, we say that they have a "*matching*" free caret. Reduced pairs are characterized by the fact that the two trees do not have any matching free caret. The left pair in (2) above is a reduced pair of trees.

Given two elements $\langle s, t \rangle, \langle u, v \rangle \in F$ the product is defined as

$$\langle s, t \rangle \cdot \langle u, v \rangle = \langle s', v' \rangle$$

where $\langle s, t \rangle = \langle s', t' \rangle$, $\langle u, v \rangle = \langle u', v' \rangle$, and $t' = u'$. Clearly, the identity element of F is $\langle t, t \rangle$, and the inverse of $\langle s, t \rangle$ is $\langle t, s \rangle$.

Because the elements of F are equivalence classes, they don't encode individual instances of the Generalized Associative Law, rather equivalence classes of such instances. We say that a magma "*eventually satisfies*" an

instance of the Generalized Associative Law, if it satisfies some common expansion of it, that is, an instance in the same equivalence class.
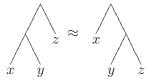
3. **Commutator Notation**

Let G be a group $x, y \in G$. We will use the standard notation

$$[x, y] := x^{-1} y^{-1} x y$$

for the commutator of x and y. We will use binary trees to denote general commutator expressions, according to the following recursive definition. If s and t are commutator expressions, then the binary tree

denotes the commutator expression $[s, t]$. This way, the 3-variable associative law for the commutator operation in a group can be written as

A *tree-like* expression in a group G is a commutator expression, where each leaf of the corresponding binary tree is labelled with either a variable, or an element of the group G. We will refer to the elements of G in a tree-like expression as *constants*. In particular, if we take the trivial binary tree, the one having a single leaf, then any element of G is a constant tree-like expression in G.

Given two tree-like expressions s and t in a group G, and a subset $X \subseteq G$, we say that X satisfies the identity $s \approx t$, and write $X \models s \approx t$, if we get equality whenever we substitute elements of X for the variables of s and t. Given a tree-like expression t in a group G, and a subset $X \subseteq G$, we denote by $t(X)$ the set of all values obtained by substituting elements of X for the variables of t.

A subset $X \subseteq G$ of a group is said to be *normal* if it is closed under conjugation by every element of G, and it is said to be *inverse* if it contains the inverse of each of its elements. We denote by B_p the full binary tree of height p. The set $B_p(G)$ is a normal, inverse, generating set of the p-th derived group $G^{(p)}$.

4. Basic Results

The commutator operation satisfies a number of identities in all groups, some of which we need to refer to explicitly.

$$[xy, z] = [x, z]^y \cdot [y, z] \tag{3}$$
$$[x, yz] = [x, z] \cdot [x, y]^z \tag{4}$$
$$[y, x] = [x, y]^{-1} = [x^y, y^{-1}] = [x^{-1}, y^x] \tag{5}$$

If G is a group and s, t, s', t' are tree-like expressions such that (s', t') is a common expansion of (s, t), then $G \models s \approx t$ clearly implies that $G \models s' \approx t'$. The converse does not hold; but if the common expansion from (s, t) to (s', t') involves hanging trees of height at most p then when $G \models s' \approx t'$ it follows that $B_p(G) \models s \approx t$. We can take this one step further when one side of the identity is trivial.

Lemma 1. *Let G be a group, t a tree-like expression and t' the result of hanging from each non-constant leaf of t a tree of height at most p.*
If $G \models t' \approx 1$ then $B_p(G) \models t \approx 1$ and $G^{(p)} \models t \approx 1$.
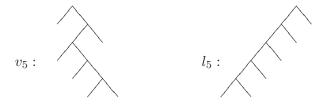
Proof. Since every tree of height at most p can be expanded to a full binary tree B_p of height p, from $G \models t' \approx 1$ we immediately get $B_p(G) \models t \approx 1$. Now, the commutator identities (3) and (4) and the fact that $B_p(G)$ is a normal, inverse, generating set of $G^{(p)}$, yield $G^{(p)} \models t \approx 1$. $\qquad\square$

The next lemma follows by a straight forward calculation. When evaluating t and s in $G/Z(G)$ the constants are to be replaced by their corresponding cosets.
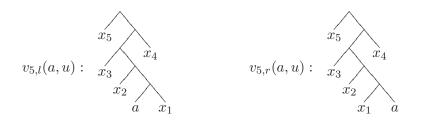
Lemma 2. *Let G be a group, $Z(G)$ its center, and let s, t be tree-like expressions in G. Let x be a variable, different from all variables that occur in s and t.*

$$G/Z(G) \models s \approx t \quad iff \ G \models \underset{s \quad x}{\bigwedge} \approx \underset{t \quad x}{\bigwedge} \quad iff \ G \models \underset{x \quad s}{\bigwedge} \approx \underset{x \quad t}{\bigwedge}$$

Each non-trivial binary tree has at least one free caret; if it has exactly one free caret, we call it a **vine**. The **left vine** of height n, denoted l_n is the vine of height n, whose free caret holds the two leftmost leaves of the vine. In the following picture we see a vine v_5 of height 5 and the left vine l_5.

$v_5:$ $l_5:$

Given a vine v_n of height n, $a \in G$, $u = (x_1, x_2, \ldots, x_n)$ we denote by $v_{n.l}(a, u)$ (resp. $v_{n.r}(a, u)$) the tree-like expression in G obtained by placing a at the left (resp. right) leaf of the free caret in v_n and $x_1, \ldots, x_n$ at the other leaves of v_n from bottom to top. In the previous example we have

$v_{5,l}(a, u):$ $v_{5,r}(a, u):$

Lemma 3. *Let v_n be a vine of height n, G a group, $a \in G$, and $u = (x_1, x_2, \ldots, x_n)$. Then*

$$v_{n,l}(a, u) = l_{n,l}(\overline{a}, \overline{u}) = l_{n,l}(a, \widehat{u})^{\pm 1}$$

where $\overline{a} \in \{a, a^{-1}\}$ and each $\overline{x_i}, \widehat{x_i}$ are conjugates of x_i. Similarly,

$$v_{n,r}(a, u) = l_{n,l}(\overline{a}, \overline{u}) = l_{n,l}(a, \widehat{u})^{\pm 1}$$

where $\overline{a} \in \{a, a^{-1}\}$ and $\overline{x_i}, \widehat{x_i}$ are conjugates of x_i.

Proof. When $n = 1$, it follows from the commutator identities (5). When $n > 1$, it follows from those same identities and induction. There are four cases to consider, two for $v_{n,l}(a, u)$ and two for $v_{n,r}(a, u)$. If

$$v_{n,l}(a, x_1, \ldots, x_n) = [x_n, v_{n-1,l}(a, x_1, \ldots, x_{n-1})] = \text{} \quad ,$$

by induction, $v_{n-1,l}(a, x_1, \ldots, x_{n-1})^{-1} = l_{n-1,l}(\overline{a}, \overline{x_1}, \ldots, \overline{x_{n-1}})$ where $\overline{a} \in \{a, a^{-1}\}$ and each $\overline{x_i}$ is a conjugate of x_i, $i = 1, \ldots n - 1$. Now, using (5),

we get

$$v_{n,l}(a, x_1, \ldots, x_n) \;=\; [x_n, l_{n-1,l}(\overline{a}, \overline{x_1}, \ldots, \overline{x_{n-1}})^{-1}] \;=$$

$$=\; [l_{n-1,l}(\overline{a}, \overline{x_1}, \ldots, \overline{x_{n-1}}), \overline{x_n}] \qquad =$$

$$=\; l_{n,l}(\overline{a}, \overline{x_1}, \ldots, \overline{x_{n-1}}, \overline{x_n})$$

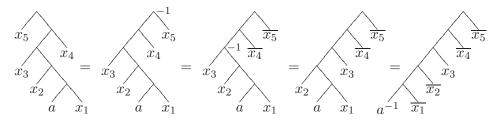where $\overline{x_n} = (x_n)^{l_{n-1,l}(a', x_1', \ldots, x_{n-1}')^{-1}}$. The other case for $v_{l,n}$

$$v_{n,l}(a, x_1, \ldots, x_n) = [v_{n-1,l}(a, x_1, \ldots, x_{n-1}), x_n] =$$

and the two cases for $v_{n,r}$ are treated similarly. The second equality of the statement, i.e.

$$l_{n,l}(\overline{a}, \overline{u}) = l_{n,l}(a, \widehat{u})^{\pm 1}$$

is obtained using (5) repeatedly to "*pull*" the inverse out of the commutator when the exponent of a is $\;^{-1}$. $\qquad\qquad\qquad\qquad\qquad\square$

We illustrate the first part of this lemma with the following example:

As an immediate consequence of Lemma 3 we obtain:

Proposition 4. *Let G be a group, $a, b \in G$, and v_n a vine of height n. If b centralizes $l_{n,l}(a, u)$ for all u, then it also centralizes $v_{n,l}(a, u)$ and $v_{n,r}(a, u)$ for all u.*

5. **The Main Result**

Before we prove the main result of this paper, we will need a counting argument which can be expressed in terms of coloring of the leaves of a full binary tree.

Given two leaves in a full binary tree, we'll refer to the distance to their closest common ancestor, as the *distance* between the leaves.

Proposition 5. *Suppose we color the leaves of a full binary tree of height $nj + 1$ so that the coloring satisfies the following condition: any two leaves at a distance*

$$d \equiv 1 \pmod{j}$$

must have different color. Then the number of colors has to be $\geq 2^n$.

Proof. The proof is by induction on n. When $n = 1$, any leaf on the left subtree is at a distance $j + 1$ from any leaf on the right subtree. Hence at least two colors are needed. For $n > 1$, consider the 2^j subtrees, $t_1, \ldots, t_{2^j}$ which are full binary of height $(n-1)j + 1$. Each one of them needs at least 2^{n-1} colors. Each of the leaves on the leftmost subtree, t_1, are at a distance $nj + 1$ from each of the leaves on the rightmost subtree, t_{2^j}, hence all the 2^{n-1} colors used for t_1 must be different from the 2^n used for t_{2^j}, and we need 2^n colors. $\qquad\square$

Although we don't need it here, it is not hard to see that this lower bound is tight, i.e. one can always do the coloring with 2^n colors.
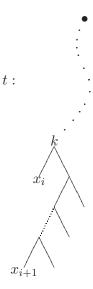
We now get to the main result of this paper.

Theorem 6. *If the commutator operation in a finite group G satisfies some instance of the generalized associative law, then G is solvable.*

Proof. Let s', t' be two binary trees with the same number of leaves, so that $s' \approx t'$ is an instance of the generalized associative law. Let G be a finite group such that $G \models s' \approx t'$. Let (s, t) be a reduced pair of trees, which represents the equivalence class $\langle s', t' \rangle \in F$. Since (s', t') is obtained from (s, t) by a common expansion, there is a $p \geq 0$ such that $B_p(G) \models s \approx t$. In fact, p can be taken to be the max of the heights of the trees used to expand (s, t) into (s', t'). Label the variables of both s and t with variable names $x_1, x_2, \ldots$ from left to right, and let $\overset{\displaystyle\wedge}{\underset{x_i \; x_{i+1}}{}}$ be the leftmost free caret in either s or t; without lost of generality, let's say in s. Let k be the lowest common ancestor of x_i and x_{i+1} in t, and r the subtree of t rooted at k. Since t has no free caret to the left of x_{i+1}, the left child of r has no free caret, and hence it has to be the leaf x_i; x_{i+1} is the first leaf of the right

child of r. So r is an expansion of $\overset{\displaystyle\frown}{\underset{x_i \quad l_{j,l}(x_{i+1}, y)}{}}$ for variables $y_1, \ldots, y_j$. Consider the path from the root of t to the leaf x_{i+1}. This path goes through the vertex k, and

determines a vine $v_{m,l}(x_i, l_{j,l}(x_{i+1}, y), z_2, \ldots, z_m)$, so that t is an expansion of this vine. Let p' be the max of the heights of the trees hanged at $y_1, \ldots, y_j, z_2, \ldots, z_m$ to get t. For any a and b that commute, putting them in the place of variables x_i and x_{i+1}, evaluates s to 1, and therefore, by Lemma 1

$$B_{p+p'}(G) \models v_{m,l}(b, l_{j,l}(a, y), z_2, \ldots, z_m) \approx 1$$
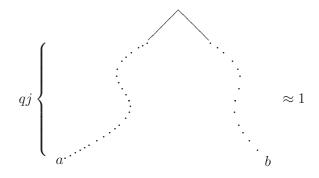
and

$$G^{(p+p')} \models v_{m,l}(b, l_{j,l}(a, y), z_2, \ldots, z_m) \approx 1$$

Repeated application of Lemma 2 yields

$$G^{(p+p')}/Z_{m-1}(G^{(p+p')}) \models [b, l_{j,l}(a, y)] \approx 1$$

Let $H = G^{(p+p')}/Z_{m-1}(G^{(p+p')})$. Thus, we have shown the following fact for H: if b commutes with a then b centralizes $l_{j,l}(a, u)$ for all $u_i \in H$. Applying this fact to b and $l_{j,l}(a, u)$, we conclude that b centralizes $l_{2j,l}(a, u)$, and by induction on q we get that b centralizes $l_{qj,l}(a, u)$. By Proposition 4, we get that for any vine v_{qj}, b centralizes $v_{qj,l}(a, u)$ and $v_{qj,r}(a, u)$. Now, applying the fact to $v_{qj,l/r}(a, u)$ and b, induction, and Proposition 4, we conclude that $v_{qj,l/r}(a, u)$ commutes with $w_{qj,l/r}(b, u')$ for any vines v_{qj} and w_{qj}, and any $u, u' \in H$. So, in a full binary tree of height $qj + 1$, if two of the leaves, one in the left subtree and one in the right subtree, are labelled with elements that commute, the whole tree evaluates to 1.

Choose n such that $2^n > |H|$. Let B_{nj+1} be the full binary tree of height $nj + 1$. By Proposition 5, if we label the leaves with elements of H, there must be at least two leaves at a distance $qj + 1$ having the same label, for some $q \leq n$. The full binary tree B_{qj+1} containing these two leaves evaluates to 1, and so does B_{nj+1}. That means $B_{nj+1}(H) = 1$ and by Lemma 1, $H^{(nj+1)} = 1$; so H is solvable, and so is G. $\qquad\square$

References

[1] J. W. Cannon, W. J. Floyd and W. R. Parry, Introductory notes on Richard Thompson's groups, Enseign. Math. (2) **42** (1996), n. 3-4, 215–256. MR 1426438 (98g:20058).

[2] Ross Geoghegan and Fernando Guzmán, Associativity and Thompson's group, Topological and asymptotic aspects of group theory, Contemp. Math., vol. 394, Amer. Math. Soc., Providence, RI, 2006, pp.113–135, MR 2216710 (2007a:20067).

[3] F. W. Levi, Groups in which the commutator operation satisfies certain algebraic conditions, J. Indian Math. Soc. (N.S.) **6** (1942), 87–97. MR 0007417 (4,133i).

[4] Richard J. Thompson, Handwritten notes.