Letter Counting in Groups

Isaac Cheng, Karn Chutinan, Anastasia Lee

Mentored by Dev Sinha

September 26, 2024

Contents

1. Introduction

1.1. Group Presentations and the Word Problem

Throughout mathematics, groups are used to describe symmetry in all of its many forms: geometric shapes, algebraic objects, or even topological structures. To be able to compare and study these very different ideas of symmetry, groups are often described based on generators and relations between them.

Consider the group D_8, the symmetries of the square. One basic symmetry is the 90-degree clockwise rotation (which we will call r) and another is horizontal reflection (which we will call s). As either rotating a square four times by 90 degrees or performing two horizontal reflections is the same as doing nothing at all, we write $r^4 = 1$ and $s^2 = 1$. Rotation and reflection also interact with each other according to the equation $srsr = 1$. It turns out that these relations are enough to completely characterize the group as $D_8 = \langle r, s \mid r^4 = s^2 = srsr = 1 \rangle$.

Now, we can write an element of D_8 as a **word** in terms of these generators, for example $rsrrsr^{-1}s^2r$, where r^{-1} denotes the inverse of r (a 90-degree counterclockwise rotation, or equivalently three 90-degree clockwise rotations). By repeatedly applying our relations, including the facts that $r^{-1}r = 1$, $srsr = 1 \iff rsr = s^{-1}$, and $s^2 = 1 \iff s = s^{-1}$, we can see that $rsrrsr^{-1}s^2r = rsrrsr^{-1}r = rsrrs = s^{-1}rs = srs = r^{-1}$.

Correctly applying relations to reduce a word to its minimal form requires some degree of ingenuity, so a natural question is whether there is some easier way. That is, given some word w in a group $G = \langle S \mid R \rangle$ where S is a set of generators and R a set of relations between them, is there an algorithm that is guaranteed to reduce w to its minimal form within finite time? Specifically, we wonder whether an algorithm can determine whether $w = 1$ in finite time, a question known as the **word problem**. The word problem is solved for many classes of groups, including finite groups, but using relatively complicated algebraic machinery. In this project, we will demonstrate a theory that solves the word problem for solvable finite groups using basic combinatorial functions.

1.2. Combinatorial Possibilities

A basic approach to the word problem might be to count the number of letters appearing in a word. For example, in $D_8 = \langle r, s \mid r^4 = s^2 = srsr = 1 \rangle$, we might check that the number of times the letter r appears (henceforth denoted R) is divisible by 4 and the number of times the letter s appears (henceforth denoted S) is even. Unfortunately, this fails to account for noncommutativity - $rssr = r^2$, but $srsr = 1$!

While the idea of counting letters doesn't work immediately, it offers a good foundation upon which to build. Continuing to use capital-letter notation to denote counting functions for letters (A counts as, B counts bs, etc.), let $[\![A]\!]B$ count the number of bs occurring after as, reversing sign for inverses. For instance, $[\![A]\!]B(abaab^{-1}) = 1 - 3 = -2$, since the first b occurs after one a and the second b is a inverse occurring after 3 as. We call

this function a "weight one letter-braiding function," as will be defined later.

These functions allow us to distinguish elements even in nonabelian groups. In fact, it will later be shown that for any word $w \in D_8$, the functions $R, S, \llbracket R \rrbracket R + \llbracket S \rrbracket R : D_8 \to \mathbb{Z}_2$ all vanish on w if and only if w is equal to the identity in D_8, solving the word problem for that group.

1.3. Purpose of this Project

For this project, we explored "letter braiding functions" of the form $\llbracket A \rrbracket B$ and their generalizations, specifically with regard to the word problem and the problem of distinguishing groups. It will be demonstrated that for certain groups G, if a word vanishes under all well-defined letter braiding functions from G to the ring $\mathbb{Z}_{|G|}$, it must be equal to the identity in G.

2. Understanding Letter Braiding Functions

In order to be more precise with our descriptions of groups and relations, we'll define a more formal framework.

2.1. Groups

We begin with the standard definitions of groups, subgroups, conjugates, cosets, and quotient groups.

Definition 2.1 — A **group** $(G, \cdot)$ is a set G together with a binary operation $\cdot : G \times G \to G$, such that for all g_1, g_2, g_3 in G,

1. There exists an element $1 \in G$ such that for all $g \in G$,
$$1 \cdot g = g \cdot 1 = g,$$
2. For any g_1, g_2, g_3 in G,
$$(g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3),$$
3. For any $g \in G$, there exists a unique g^{-1} such that
$$g \cdot g^{-1} = g^{-1} \cdot g = 1.$$

We will omit the $\cdot$ and simply write 'Let G be a group.'

Additionally, if for all $g, h \in G$ we have $gh = hg$, then we say G is **abelian**.

Definition 2.2 — A **subgroup** H of G is a group H whose underlying set is a subset of G.

Definition 2.3 — Given an element $x \in G$, the conjugate of x by g is the element gxg^{-1}. x and y are said to be **conjugate** if and only if the conjugate of y by some element g is x.

It can be checked that conjugacy is an equivalence relation.

Definition 2.4 — Given an element $g \in G$, the left **coset** of a subgroup H of G is the set
$$gH := \{gh : h \in H\}.$$
The right coset Hg is defined similarly.

Definition 2.5 — A subgroup H in G is called **normal** if for every g in G, the cosets gH and Hg are equal.

We state the following proposition without proof:

Proposition 2.6. If H is normal in G, and g_1 and g_2 are in H, then the set

$$g_1H \cdot g_2H = \{g_1h_1g_2h_2 : h_1, h_2 \in H\}$$

is equal to g_1g_2H.

Definition 2.7 — If H is normal in G, we define the quotient group G/H as follows:

1. The underlying set is the left cosets of G,
2. If g_1, g_2 are elements of two cosets, then the multiplication of the two cosets is just g_1g_2H.

It is well known that such a group is well-defined.

We define the combinatorial objects we will be working with now.

Definition 2.8 — A **word** w over a set S is a finite expression of the form

$$s_1^{\varepsilon_1}s_2^{\varepsilon_2}\dots s_n^{\varepsilon_n},$$

where each s_i is in S and each ε_o is ± 1. The length of the word is the integer n.

Since we want some semblance of a group structure, we now define reductions:

Definition 2.9 — A **reduction** of the word is a removal of ss^{-1} or $s^{-1}s$ pair to make a shorter word w, of length $n-2$. We observe that any chain of reductions must eventually terminate, since length is a nonnegative integer. As such we define a **completely reduced form** of w to be the terminating word at the end of a chain of reductions

$$w \to w_1 \to \cdots \to w_k.$$

For example if $S = \{a, b, c\}$, we have $w = ab^{-1}cc^{-1}b$ reduces to a.
The following can be proven with induction:

Proposition 2.10. All chains of reductions of a word w lead to the same completely reduced form.

As such, it makes sense to talk about **the reduced form** of a word w, and we say that two words are **equivalent** if they have the same reduced form.

Definition 2.11 — We now define the **free group on n elements**, F_n, to be words over $\{g_1, \dots g_n\}$ up to equivalence, with an operation $\cdot$ defined as follows: if $w_1 = s_1^{\varepsilon_1} \dots s_k^{\varepsilon_k}$ and $w_2 = t_1^{\varepsilon_1} \dots t_\ell^{\varepsilon_\ell}$, then

$$w_1 w_2 = s_1^{\varepsilon_1} \dots s_k^{\varepsilon_k} t_1^{\varepsilon_1} \dots t_\ell^{\varepsilon_\ell}.$$

It is well known that this gives a well defined group under this operation; associativity is easily checked using 2.10, identity is the empty word, and inverse is just reversing the order of letters and reversing signs.

Definition 2.12 — A **group presentation** is a set S of generators and a set $\mathcal{R}$ of words over S. We take the free group on $|S|$ elements, whose generators are the set S, and let $\mathcal{R}'$ be the subgroup generated by all conjugates of elements of $\mathcal{R}$. It is verifiable that $\mathcal{R}'$ is normal in $F_{|S|}$. The group induced by this group presentation is $F_{|S|}/\mathcal{R}'$, and we denote it by

$$\langle g_1, g_2, \dots g_n \mid r_1 = 1, r_2 = 1, \dots, r_n = 1 \rangle$$

where the g_i are in S, and the r_i are in $\mathcal{R}$. We may shorten either side to S or $\mathcal{R}$ respectively.

Example 2.1

The group D_8 is given by

$$D_8 \cong \langle r, s : r^4 = 1, s^2 = 1, srsr = 1 \rangle.$$

Here the set S is $\{r, s\}$ and the set R is $\{rrrr, ss, srsr\}$.

2.2. Towards Letter Braiding Functions

These definitions are due to [2]. Throughout this section, we let R denote an arbitrary commutative ring. We will be considering the $\mathbb{Z}$-module of homomorphisms from G to R.

Our functions of interest can be described purely combinatorially. For example given something like $\llbracket A \rrbracket B$, we wish to count the pairs of a, b such that a is before b, reversing sign based off of the number of inverses are in the pair.

Definition 2.13 — Given F_n, the **indicator homomorphism** h_1 to R counts the number of g_1's in a given word w. Here, g_1^{-1} counts as -1; this gives rise to a well-defined function on F_n. We define similar homomorphisms h_i for each $i = 2, \dots, n$. In an arbitrary group these may not stay well-defined; in this case we just call these **letter counting functions**.

For given letters we may use a capital letter to denote the indicator homomorphism instead of say, h_a. That is, A is the homomorphism on F_n which counts a's in a given word w.

Now we'll try to capture the essence of counting "a's before b's." One way to think of it is that for each b, we count the number of a's before it, and add that count to the total. This motivates a method of computation; we keep track of the number of a's by summing the value of A on the letters before the b. More concretely, it looks like this, and summing the last row yields $\llbracket A \rrbracket B = 1$:

	a	a	b	b	a	b^{-1}	a
A	1	1	0	0	1	0	1
B	0	0	1	1	0	-1	0
$\int_{[i-1]} A$	0	1	2	2	2	3	3
$\llbracket A \rrbracket B$	0	0	2	2	0	-3	0

Table 1: Example tabulation

Here the $\int_{[i-1]} A$ symbol stands for summing the value of A on the first $i = 1$ letters.

More formally,

Definition 2.14 — For a discrete function f on naturals to some ring, define

$$\int_{[n]} f = \sum_{i=1}^{n} f(i).$$

However, when we try to count something like $\llbracket A \rrbracket A$, we run into a problem with inverses. We know that $aa^{-1} = a$, but when we apply our current definition we get

$$\llbracket A \rrbracket A(aa^{-1}a) = -1, \qquad \llbracket A \rrbracket A(a) = 0.$$

It turns out we need to count a^{-1} as "before itself." This will make sense once we establish a graphic interpretation of these functions.

Definition 2.15 — We define $[i]_w$ for a fixed word w to be $[i]$ if the ith letter is an inverse, and $[i-1]$ otherwise.

This allows a^{-1} to 'count with itself.' Notably we have $\llbracket A \rrbracket A(aa^{-1}a) = 0$, which is consistent with $\llbracket A \rrbracket A(a) = 0$.

We now define the framework for letter braiding functions.

Definition 2.16 — We define **formal braiding symbols** inductively over a set of n variables, L_1, L_2, …, L_n as follows: all the L_i are formal braiding symbols, any linear combination of symbols is a formal braiding symbol, and given two symbols f and g, then $\llbracket f \rrbracket g$ is also a formal braiding symbol.

Additionally, the **weight** of a symbol is defined as the maximum amount of braiding products that occur in any term of the symbol, so $\llbracket A \rrbracket \llbracket B \rrbracket C$ has weight 2, for example.

We also call a symbol **pure** if we don't ever use addition in generating the symbol inductively.

Now we place homomorphisms into these symbols.

Definition 2.17 — Fix a word $w \in F_n$. Given two discrete-valued functions f and g, whose domains are both $[n]$, we define their **left braiding product** over w as

$$\llbracket f \rrbracket g(i) = \left(\int_{[i]_w} f \right) g(i).$$

Definition 2.18 — For a fixed word $w \in F_n$ and braiding symbol f, we assign a **associated function** $w^* f$ inductively; if f is a homomorphism, then $w^* f(n)$ just returns the valuation of f on the nth letter of w. Otherwise,

$$w^* \left(\llbracket f \rrbracket g \right) = \llbracket w^* f \rrbracket w^* g.$$

Associated functions are called pure if their corresponding symbols are pure.

Note that inductively $w^* f$ and $w^* g$ are discrete functions, so we use the braiding product definition. This splits $w^* f$ into more summations, but keeps $w^* g$.

Definition 2.19 — The **letter braiding function** for a formal braiding symbol f takes a word w to $\int_{[n]} w^* f$, where n is the length of w.

For concreteness, we will do a sample computation of $\llbracket\llbracket A\rrbracket B\rrbracket\llbracket C\rrbracket C$.

		a	b	c^{-1}	c	b	c	a	b^{-1}	c	Sum
	A	1	0	0	0	0	0	1	0	0	2
	B	0	1	0	0	1	0	0	-1	0	1
	C	0	0	-1	1	0	1	0	0	1	2
	$\int_{[i]_w} A$	0	1	1	1	1	1	1	2	2	
	$\int_{[i]_w} C$	0	0	-1	-1	0	0	1	1	1	
$(\int A)B$	$w^*\llbracket A\rrbracket B$	0	1	0	0	1	0	0	-2	0	0
$(\int C)C$	$w^*\llbracket C\rrbracket C$	0	0	1	-1	0	0	0	0	1	1
	$\int_{[i]_w} w^*\llbracket A\rrbracket B$	0	0	1	1	1	2	2	0	0	
$\left(\int(\int A)B\right)(\int C)C$	$w^*\llbracket\llbracket A\rrbracket B\rrbracket\llbracket C\rrbracket C$	0	0	1	-1	0	0	0	0	0	0

Table 2: Example tabulation

For now, we state the following proposition without proof; this is to ensure that letter braiding functions do work as functions on groups.

Proposition 2.20. Letter braiding functions on words descend to well-defined functions on free groups.

2.3. A Combinatorial Interpretation

It turns out that these letter braiding functions have a nice combinatorial interpretation, which may make their evaluation seem much more 'natural.'

First, some notation:

Definition 2.21 — For a given word $w = s_1^{\varepsilon_1}\dots s_n^{\varepsilon_n}$, the set of **ordered letters** is

$$L(w) := \{(i, \varepsilon_i, s_i) : 1 \le i \le n\}.$$

To motivate our general description, we consider a braiding symbol like

$$\sigma = \llbracket\dots\llbracket L_1\rrbracket L_2\dots\rrbracket L_n.$$

Fix a word w, and we consider multisubsets M of $L(w)$ which have size n and the only ordered letters $(k, \varepsilon_k, s_k) \in M$ that appear more than once satisfy $\varepsilon_k = -1$. If $M = \{(k_1, \varepsilon_{k_1}, s_{k_1})\dots(k_n, \varepsilon_{k_n}, s_{k_n})$, and $k_1 \le k_2 \le \dots \le k_n$, then we consider M to **match** the symbol σ if $s_{k_i} = L_i$ for all $1 \le i \le n$.

Intuitively, we are counting the number of (not necessarily contiguous) 'substrings' of the word w which are L_1 to L_n in order, but counting inverses more, and then accounting for signs.

We have the following proposition, which we will prove later:

Proposition 2.22. Let σ be a pure braiding symbol and w a word. Denote by $\mathcal{M}_\sigma$ to be the multisets of $L(w)$ which match σ. Then the evaluation of the associated function f of σ on w turns out to be

$$\sum_{M \in \mathcal{M}_\sigma} \prod_{(k,\varepsilon_k,s_k) \in M} \varepsilon_k.$$

Again the interior product is just counting the number of negative signs. This allows us to actually generalize to inverses.

Definition 2.23 — For a formal pure braiding symbol f, we construct an **associated graph**, which is a labeled directed graph $G(f)$ with a **terminal vertex** as follows: if f is a formal variable, say L_1, we construct a 1-vertex directed graph with terminal vertex labeled L_1. Otherwise, $f = [\![g]\!]h$, and we take the directed graph by connecting the terminal vertex of $G(g)$ towards the terminal vertex of $G(h)$. The terminal vertex of $G(f)$ is the terminal vertex of $G(h)$.

We sill similarly say the associated graph $G(f)$ for a braiding function f with associated symbol σ is $G(\sigma)$. We omit the proof of the following fact:

Proposition 2.24. For any pure braiding symbol f, $G(f)$ has no cycles. Hence, it is possible to interpret $G(f)$ as a poset.

Definition 2.25 — Now for a word w, and pure braiding function f, we say a coloring of $G(f)$ using elements of $L(f)$ is valid if for any adjacent vertices $v_1 \to v_2$ with colors $(k_1, \varepsilon_{k_1}, s_{k_1})$ and $(k_2, \varepsilon_{k_2}, s_{k_2})$ respectively, we have

1. In $G(f)$, v_1 is labeled with s_{k_1}, likewise for v_2,
2. If $\varepsilon_{k_2} = -1$, $k_1 \leq k_2$, otherwise $k_1 < k_2$.

The **value** of the coloring is then the product of the ε_i on all the labels.

Intuitively, this corresponds to laying the graph beneath the word such that the label of each vertex matches up with the word above it. Strict order is preserved unless the letter above an edge is an inverse.

Here's the theorem:

Theorem 2.26. Let f be a pure letter braiding function. Then $f(w)$ is equal to the sum of the values of all valid colorings of $G(f)$ using $L(w)$.

Proof. It suffices to show that $w^*f(n)$ is equal to the sum of the values of all colorings of $G(f)$ with the terminal vertex colored with (n, ε_n, s_n). Observe the base case of indicator homomorphisms is not hard; indeed, for a given indicator homomorphism h_1 we have that $w^*h_1(n)$ is ε if the nth letter of w is (n, ε, l_1), and 0 otherwise, which coincides with the definition of valid covering with terminal vertex colored with (n, ε_n, s_n).

We now induct downwards. As f is pure, we can let $f = \llbracket g \rrbracket h$ for other pure braiding symbols g and h, and recall that

$$w^*f = \llbracket w^*g \rrbracket w^*h.$$

We treat the two branches separately. Let $G'(g)$ denote the part of $G(f)$ that came from $G(g)$, and similarly define $G'(h)$. Then the number of valid colorings of $G(f)$ with terminal vertex colored (n, ε_n, s_n) is as follows: it's the number of valid colorings of $G(h)$ whose terminal vertex is colored as (n, ε_n, s_n), multiplied by the number of valid colorings of $G(g)$ whose terminal vertex is before n, or if $\varepsilon_n = -1$, the terminal vertex could be at n. These are induced by $G'(h)$ and $G'(g)$ respectively.

The former by inductive hypothesis is equal to w^*h. The latter coincides directly with the quantity $\int_{[n]_w} w^*g$. Particularly, this is precisely the braiding product $w^*\llbracket g \rrbracket h$, as desired. □

Proposition 2.22 is a corollary of this result.

Example 2.2

Our above definition matches up with the multiset description of $\llbracket \ldots \llbracket L_1 \rrbracket L_2 \ldots \rrbracket L_n$, since that graph is just a directed path $L_1 \to L_2 \to \cdots \to L_n$.

Example 2.3

On the word *aabcabcba*, we could go through the effort of computing $\llbracket a \rrbracket \llbracket c \rrbracket b$ manually, but we can instead enumerate per b. The first b gives 0, second b gives $3 \cdot 1$, and the last one gives $3 \cdot 2$. So we have the value of 9 on this word.

This now allows us to compute things combinatorially, very fast:

Proposition 2.27. $\llbracket A \rrbracket B(w) + \llbracket B \rrbracket A(w) = A(w)B(w)$.

Proof. On the one hand, the right hand side counts pairs of a and b up to sign. On the other hand the left hand side counts pairs of a and b where a is 'before' b, and b is 'before' a. The only issue is the sign; for each b it turns out we count $A(w)$ for both functions. We are done. □

More interestingly we can prove Proposition 2.20 with no computation now, but it is a bit long to formalize.

Proof. It suffices to verify the statement for various pure letter braiding functions, since all others are linear combinations of these.

Let f be a pure letter braiding function. We wish to show that for any words w and w', $f(ww') = f(waa^{-1}w') = f(wa^{-1}aw')$. In this case, let the corresponding ordered letters of a and a^{-1} in $L(waa^{-1}w')$ be $(k, 1, a)$ and $k+1, -1, a)$. We will handle $f(ww') = f(waa^{-1}w')$ by finding sets of colorings whose values sum to 0.

We will use the 'number of ways to place graph beneath the word' interpretation. Indeed, every possible placement consists of vertices colored with $(k, 1, a)$, then some other vertices colored with $(k+1, -1, a)$. Let the former set of vertices be V, and denote by $C_{\notin V}$ the set of pairs (v, c) where c is the coloring of vertex v, and v is not in V. This is to keep track of coloring that uses a vertex of V, where we fix what happens outside of V.

Consider the set of valid graph colorings using $L(waa^{-1}w')$ where a vertex v is colored with $(k, 1, a)$ or $(k+1, -1, a)$ if and only if that vertex is in V, and call it S. Then viewing the subgraph induced by V as a poset, this has some set of maximal elements. Let the set of vertices in the original graph corresponding to these elements be M. Visually, these are the 'farthest left.'

Observe that for each vertex $v \in V$, but $v \notin M$, we are forced to color it with $(k+1, -1, a)$. This is because if there existed no edge going into v in the subgraph induced by V, then v would be in M. As such, let $v' \to v$ be the edge going into v, and if v is colored $(k, 1, a)$, then v' would need to be colored some (n, ε_n, s_n) for $n < k$. However, both k and $k+1$ don't satisfy this condition, contradiction.

Now for every vertex $v \in M$, we can color it with either $(k, 1, a)$ or $(k+1, -1, a)$. This provides no contradictions, and since every other vertex not in V has a position either less than k or greater than $k+1$, it follows that we can extend this using $C_{\notin V}$ to a valid coloring.

Since exactly half of these give an odd number of vertices underneath a, and the other half give an even number of vertices underneath a^{-1}, the sum of the values of S is 0.

Summing over all possible pairs $(V, C_{\notin V})$, where V is nonempty, we find that the sum of the contributions of every valid coloring that has $(k, 1, a)$ or $(k+1, -1, a)$ is 0. To finish this case, note that every valid coloring of $G(f)$ using $L(waa^{-1}w')$ can be bijected to a valid coloring of $G(f)$ using $L(ww')$ by recoloring any letter $(n, \varepsilon_n, s_n) \in L(waa^{-1}w')$ with $n \geq k+2$ with the corresponding $(n-2, \varepsilon_{n-2}, s_{n-2}) \in L(ww')$.

The equality $f(wv) = f(wa^{-1}av)$ follows from taking the set of minimal elements instead, and then arguing from there. □

Lemma 2.28. For all $n \in \mathbb{N}$ and any fixed word w, we have

$$\underbrace{[\![\ldots[\![A]\!]A\ldots]\!]A}_{n\ A\text{'s}}(w) = \binom{A(w)}{n}.$$

Proof. First, notice that the value of $\underbrace{[\![\ldots[\![A]\!]A\ldots]\!]A}_{n\ A\text{'s}}$ depends only on the a's in w, since A sends all other generators to 0. Thus, $\underbrace{[\![\ldots[\![A]\!]A\ldots]\!]A}_{n\ A\text{'s}}(w) = \underbrace{[\![\ldots[\![A]\!]A\ldots]\!]A}_{n\ A\text{'s}}(w_a)$, where w_a is w with only the a's extracted. By Proposition 2.20, $\underbrace{[\![\ldots[\![A]\!]A\ldots]\!]A}_{n\ A\text{'s}}$ is a well-defined function on free groups, so we can simplify w_a with the relation $a \cdot a^{-1} = a^{-1} \cdot a = 1$ until we are left with $w_a = a^k$ for some $k \in \mathbb{Z}$.

Now consider the evaluation of $\underbrace{[\![\ldots[\![A]\!]A\ldots]\!]A}_{n\ A\text{'s}}(a^k)$. If $k = 0$, then $\underbrace{[\![\ldots[\![A]\!]A\ldots]\!]A}_{n\ A\text{'s}}(1) = 0 = \binom{0}{n}$, as desired. If $k \in \mathbb{N}$, we have:

	a^k							
	$\widehat{\ }a$	$\widehat{\ }a$	$\widehat{\ }a$	$\widehat{\ }a$	$\widehat{\ }a$	$\cdots$	$\widehat{\ }a$	Sum
A	1	1	1	1	1	$\cdots$	1	$\binom{k}{1}$
$[\![A]\!]A$	0	1	2	3	4	$\cdots$	$k-1$	$\binom{k}{2}$
$[\![[\![A]\!]A]\!]A$	0	0	1	$1+2$	$1+2+3$	$\cdots$	$\sum_{i=1}^{k-2} i = \binom{k-1}{2}$	$\binom{k}{3}$
$[\![[\![[\![A]\!]A]\!]A]\!]A$	0	0	0	1	$1+(1+2)$	$\cdots$	$\sum_{i=2}^{k-2} \binom{i}{2} = \binom{k-1}{3}$	$\binom{k}{4}$
$[\![[\![[\![[\![A]\!]A]\!]A]\!]A]\!]A$	0	0	0	0	1	$\cdots$	$\sum_{i=3}^{k-2} \binom{i}{3} = \binom{k-1}{4}$	$\binom{k}{5}$
$\vdots$						$\vdots$		
$\underbrace{[\![\ldots[\![A]\!]A\ldots]\!]A}_{n\ A\text{'s}}$	0	0	0	0	0	$\cdots$	$\sum_{i=n-2}^{k-2} \binom{i}{n-2} = \binom{k-1}{n-1}$	$\binom{k}{n}$

Table 3: Evaluating $\underbrace{[\![\ldots[\![A]\!]A\ldots]\!]A}_{n\ A\text{'s}}(a^k)$ for $k \in \mathbb{N}$

While taking partial sums and summing the rows of the table, we make use of the well-known Hockey Stick Identity, which states that

$$\sum_{i=k}^{n} \binom{i}{k} = \binom{n+1}{k+1}.$$

If $k \in -\mathbb{N}$, we have:

	a^k					
	$a^{-1}\frown$	$a^{-1}\frown$	$a^{-1}\frown$	$\cdots$	$a^{-1}\frown$	Sum
A	-1	-1	-1	$\cdots$	-1	$\binom{k}{1}$
$[\![A]\!]A$	1	2	3	$\cdots$	$-k$	$\binom{k}{2}$
$[\![[\![A]\!]A]\!]A$	-1	-3	-6	$\cdots$	$\sum_{i=1}^{-k} -i = -\binom{-k+1}{2}$	$\binom{k}{3}$
$[\![[\![[\![A]\!]A]\!]A]\!]A$	1	4	10	$\cdots$	$\sum_{i=2}^{-k+1} \binom{i}{2} = \binom{-k+2}{3}$	$\binom{k}{4}$
$[\![[\![[\![[\![A]\!]A]\!]A]\!]A]\!]A$	-1	-5	-15	$\cdots$	$\sum_{i=3}^{-k+2} -\binom{i}{3} = -\binom{-k+3}{4}$	$\binom{k}{5}$
$\vdots$				$\vdots$		
$\underbrace{[\![\ldots[\![A]\!]A\ldots]\!]A}_{n\ A\text{'s}}$	$(-1)^n$	$(-1)^n n$	$(-1)^n\binom{n+1}{n-1}$	$\cdots$	$(-1)^n\binom{-k+n-2}{n-1}$	$\binom{k}{n}$

Table 4: Evaluating $\underbrace{[\![\ldots[\![A]\!]A\ldots]\!]A}_{n\ A\text{'s}}(a^k)$ for $k \in -\mathbb{N}$

We similarly use the Hockey Stick Identity to take the partial sums, and when using it to sum the rows, we get $(-1)^n\binom{-k+n-1}{n}$. So, it suffices to show that $(-1)^n\binom{-k+n-1}{n} = \binom{k}{n}$. Expanding the left-hand side yields

$$\begin{aligned}(-1)^n\binom{-k+n-1}{n} &= (-1)^n\frac{(-k+n-1)(-k+n-2)\cdots(-k)}{n!}\\ &= (-1)^n\frac{(-1)(k)(-1)(k-1)\cdots(-1)(k-n+1)}{n!}\\ &= \frac{k(k-1)\cdots(k-n+1)}{n!}\\ &= \binom{k}{n},\end{aligned}$$

completing the algebraic proof. □

Combinatorially, for $k \in \mathbb{N}$, we can interpret $\underbrace{[\![\ldots[\![A]\!]A\ldots]\!]A}_{n\ A\text{'s}}(a^k)$ as the number of instances of $n-1$ a's after an a. This is equivalent to counting the number of ways to choose n A's out of the $A(w)$ total A's, which can be done in $\binom{A(w)}{n}$ ways.

For $k \in -\mathbb{N}$, from the graph interpretation, we simply want to count the number of ways we can lay a directed path of n vertices underneath k letters such that order is (not strictly) preserved. This is equivalent to counting the number of multisubsets of $\{1, 2, \ldots, -k\}$ with size n; each one of these corresponds to a unique directed path which preserves order, where we go from a directed path to the multiset by taking the position

of the letters its colored by, and vice versa.

Of course, it should be remarked that there are precisely $\binom{-k+n-1}{n}$ ways of doing this; a proof of this can be found by a standard 'stars and bars' argument. Notably from the algebra above, we have that this is equal to $(-1)^n\binom{k}{n}$, but then after multiplying by the proper sign, which is $(-1)^n$, we get the result of $\underbrace{\llbracket \ldots \llbracket A \rrbracket A \ldots \rrbracket A}_{n\ A\text{'s}}(a^k) = \binom{-k}{n} = \binom{A(w)}{n}$.

2.4. Letter Braiding Functions as Polynomials

It turns out we can define a notion of 'degree,' analogous to polynomial degree, which corresponds to the notion of weight.

Definition 2.29 — We say a function f on a group G has **degree** n if the function

$$\Delta_{w'} f(w) = f(w'w) - f(w)$$

is degree at most $n-1$ for all $w' \in G$, and there's some word v such that $\Delta_v f$ is degree $n-1$ exactly. We define constant functions to be degree 0, and then inductively create functions of higher degree.

If a function never reduces to degree 0, then we say it is **non-polynomial.**

Particularly, nontrivial homomorphisms to abelian groups are degree 1, as they satisfy $f(w'w) = f(w') + f(w)$.

We state the following without proof:

Proposition 2.30. Weight n letter braiding functions have degree $n+1$.

We will provide the following example to show why this 'should be' true:

Example 2.4
The function $\llbracket A \rrbracket B$ is degree 2.

Proof. We omit checking that $c + dB(w)$ is degree at most 1 for all constants c in d in the ring. This is very direct.

We claim that $\llbracket A \rrbracket B(w'w) = \llbracket A \rrbracket B(w') + A(w')B(w) + \llbracket A \rrbracket B(w)$. Indeed, combinatorially we consider the pairs of letters (a, b) in $\llbracket A \rrbracket B(w'w)$. Then one of three cases occur, as the pairs are ordered;

1. Both appear in w',
2. Both appear in w',
3. The a appears in w' and the b appears in w.

The first two cases are handled by $\llbracket A \rrbracket B(w')$ and $\llbracket A \rrbracket B(w)$. The third is counted completely by $A(w')B(w)$. So the claim is true.

Now since $\llbracket A \rrbracket B(w')$ and $A(w')$ are constant for fixed w', it follows that this function is degree 1. For a function that gives exactly degree 1, take $w' = a$. □

Similar statements should hold for higher 'degree' functions.

2.5. Computing Well-Defined Letter Braiding Functions

Firstly a definition:

Definition 2.31 — Let $\mathcal{R}$ be a set of words in F_n, and let S be any set. Take the group presentation $G := \langle g_1, g_2, \ldots, g_n \mid \mathcal{R} \rangle$. Then we say a function $f : F_n \to R$ is **well-defined** on G if f is constant on cosets of the subgroup of F_n generated by $\mathcal{R}$.

A naive guess would be that a letter braiding function which vanishes on all relations should be well-defined on a group. But while vanishing on relations is necessary, it is not sufficient. For example, consider a group with a relation $a = b$, or $r = ab^{-1} = 1$. Then the function $\llbracket A \rrbracket B + A$ vanishes on r. But then for example $b^{-1}a$ is also equal to the identity element, and the count $\llbracket A \rrbracket B + A$ on $b^{-1}a$ will equal to 1, not zero. More generally, if $r = 1$ then all conjugates of r will also equal the identity element, as $grg^{-1} = gg^{-1} = 1$. So, one must check that a function vanishes not only on relations but on all of their conjugates.

One natural question is whether or not this is enough to ensure that a function is well-defined. However, we note it is enough:

Proposition 2.32. If a letter braiding function f vanishes on a word w and all of its conjugates in F_n, then f is well-defined on $\langle g_1, g_2, \ldots g_n \mid w = 1 \rangle$.

Notice that well-defined letter braiding functions up to any weight i can form a module if we choose a ring for their codomain. That is, suppose f, g are well-defined letter braiding functions of weight up to i from a group G to some ring R. If $f(id) = g(id) = 0$, where id is the identity in G, then $(f + g)(id) = 0$ and $kf(id) = 0$ for all $k \in R$.

One can programmatically compute a basis for the module of well-defined letter braiding functions $G \to R$ up to some weight i. In broad terms, a program should evaluate all formal braiding symbols of weight up to i on the relations of G, and then use linear algebra to find the basis of linear combinations of the formal braiding symbols that vanish on the relations of G, which is the basis of the intended module. However, the program must also verify that these letter braiding functions also vanish on all conjugates of the relation, which is impossible to check in entirety. Nonetheless, checking the functions up to 4th level conjugates (any more becomes computationally difficult) appears to produce provably well-defined functions in almost every case.

3. Towards the Word Problem

We will now demonstrate that letter braiding functions allow us to solve the word problem for some specific groups.

3.1. $\mathbb{Z}_2 \times \mathbb{Z}_4$

Consider $\mathbb{Z}_2 \times \mathbb{Z}_4 = \langle a, b \mid a^2 = b^4 = aba^{-1}b^{-1} = 1 \rangle$.

Proposition 3.1. The functions A, B, and $\llbracket B \rrbracket B$ solve the word problem for $\mathbb{Z}_2 \times \mathbb{Z}_4$ going to $\mathbb{Z}_2$.

Proof. First, let us verify that the functions vanish on words that are the identity. We can use the commutativity relation to write $w = a^{A(w)}b^{B(w)}$, so if $w = 1$, we must have $2 \mid A(w)$ and $4 \mid B(w)$. Then clearly $A(w) \equiv B(w) \equiv 0 \pmod 2$. Moreover, $4 \mid B(w)$ implies that $\llbracket B \rrbracket B = \binom{B(w)}{2} = \frac{B(w)(B(w)-1)}{2} \equiv 0 \pmod 2$, since $\frac{B(w)}{2} \equiv 0 \pmod 2$. Thus, A, B, and $\llbracket B \rrbracket B$ all vanish on w if it is the identity.

Now suppose that all three functions vanish on w. Then $2 \mid A(w)$, $2 \mid B(w)$, and $2 \mid \llbracket B \rrbracket B$. Since $\llbracket B \rrbracket B = \frac{B(w)(B(w)-1)}{2}$, we must have either $4 \mid B(w)$ or $4 \mid B(w) - 1$. But we know $2 \mid B(w)$, so we can conclude that $4 \mid B(w)$. Finally, using commutativity, we can write $w = a^{A(w)}b^{B(w)}$, and given that $2 \mid A(w)$ and $4 \mid B(w)$, we can simplify to $w = 1$ using the relation $a^2 = b^4 = 1$. □

3.2. Q_8

Next, consider $Q_8 = \langle a, b \mid a^4 = a^2b^{-2} = bab^{-1}a = 1 \rangle$.

Proposition 3.2. The functions A, B, and $\llbracket A \rrbracket A + \llbracket B \rrbracket A + \llbracket B \rrbracket B$ solve the word problem for Q_8 going to $\mathbb{Z}_2$.

Proof. One can verify that these functions vanish on the relations of $\mathbb{Z}_2 \times \mathbb{Z}_4$. Let us further show that they vanish on all conjugates of the relations.

We can use the relations in the following manner to simplify any given word by separating the a's and the b's:

1. We can write $w = ba^{x_1}ba^{x_2} \cdots ba^{x_n}$ for $x_i \in \mathbb{Z}$.
   Using $a^2 = b^2$, we can ensure that none of the powers of b throughout the word are greater than 1. Moreover, if the word starts with a, we can use $ba = a^{-1}b$ to make it start with b.

2. We can move a's to the left of b's.

Lemma 3.3. For $x \in \mathbb{Z}$, $ba^x = a^{-x}b$.

Proof. We can look at the three cases: $x = 0$, $x \in \mathbb{N}$, and $x \in -\mathbb{N}$. For $x = 0$, it is clearly true that $b = b$. Then for $x \in \mathbb{N}$ we proceed by induction on x. For the base case, $x = 1$, the equation follows from the relation $bab^{-1}a = 1$. Now assume that $ba^k = a^{-k}b$ for some $x = k \in \mathbb{N}$. Then

$$ba^{k+1} = ba^k a = a^{-k}ba = a^{-k}a^{-1}b = a^{-(k+1)}b,$$

as desired.

We proceed similarly for $x \in -\mathbb{N}$. For the base case, $x = -1$, we have

$$\begin{aligned} 1 &= bab^{-1}a \\ a^{-1} &= bab^{-1} \\ ba^{-1} &= b^2ab^{-1} = a^3b^{-1} = ab^2b^{-1} = ab. \end{aligned}$$

Now assume that $ba^k = a^{-k}b$ for some $x = k \in -\mathbb{N}$. Then

$$ba^{k-1} = ba^k a^{-1} = a^{-k}ba^{-1} = a^{-k}ab = a^{-(k-1)}b,$$

as desired. □

3. We can gather the a's to the left of the word.
By repeatedly using Lemma 3.3, we can write $ba^{x_1}ba^{x_2}\cdots ba^{x_n}$ as $a^{\sum_{i=1}^n (-1)^i x_i}b^n$:

$$\begin{aligned} ba^{x_1}ba^{x_2}\cdots ba^{x_n} &= a^{-x_1}ba^{-x_2}ba^{-x_3}b\cdots a^{-x_n}b \\ &= a^{-x_1}a^{x_2}ba^{x_3}b\cdots a^{x_n}b^2 \\ &= a^{-x_1}a^{x_2}a^{-x_3}b\cdots a^{-x_n}b^3 \\ &\vdots \\ &= a^{-x_1+x_2-x_3+\cdots+(-1)^n x_n}b^n. \end{aligned}$$

Therefore, for any word w, we have $w = a^{\sum_{i=1}^n (-1)^i x_i}b^n$. Note that $\sum_{i=1}^n (-1)^i x_i = A(w)$ and $n = B(w)$ are constant across all conjugates of w, so given that the functions vanish on all relations of Q_8, they also must vanish on all conjugates of the relations. Moreover, by Proposition 2.32, the functions are well-defined on Q_8.

Now assume that a word w vanishes on the functions A, B, and $[\![A]\!]A + [\![B]\!]A + [\![B]\!]B$. We seek to show that necessarily $w = 1$. First, since w vanishes on A and B, we have that $A(w), B(w) \equiv 0 \pmod 2$. Next, by Lemma 2.28,

$$[\![A]\!]A(w) = \binom{A(w)}{2} = \frac{A(w)(A(w)-1)}{2} = \frac{A(w)^2}{2} - \frac{A(w)}{2}.$$

We know that $A(w) \equiv 0 \pmod 2$, so $\frac{A(w)^2}{2} \equiv 0 \pmod 2$. Thus, $[\![A]\!]A(w) \equiv -\frac{A(w)}{2} \pmod 2$.

Similarly, $[\![B]\!]B(w) \equiv -\frac{B(w)}{2} \pmod 2$. Since there are no a's after b's in $a^{\sum_{i=1}^{n}(-1)^i x_i} b^n$, $[\![B]\!]A(w) = 0$. So,

$$([\![A]\!]A + [\![B]\!]A + [\![B]\!]B)(w) \equiv -\frac{A(w)}{2} - \frac{B(w)}{2} \equiv 0 \pmod 2,$$

which is equivalent to $A(w) + B(w) \equiv 0 \pmod 4$.

Now, with this condition as well as $A(w), B(w) \equiv 0 \pmod 2$, the only possible cases are $A(w) \equiv B(w) \equiv 0 \pmod 4$ and $A(w) \equiv B(w) \equiv 2 \pmod 4$. In both cases, we will obtain $w = 1$ after simplifying with $a^4 = 1$ and $a^2 = b^2$. Thus, if a word vanishes on all three functions, it must be the identity. □

3.3. D_8

Recall that $D_8 = \langle a, b \mid a^4 = b^2 = baba = 1 \rangle$.

Proposition 3.4. The functions A, B, and $[\![A]\!]A + [\![B]\!]A$ solve the word problem for D_8 going to $\mathbb{Z}_2$.

Proof. Again, one can verify that these functions are well-defined on the relations of D_8. Now notice that from $baba = 1$, we get $ba = a^{-1}b^{-1} = a^{-1}b^{-1}b^2 = a^{-1}b$, so by the same process as in Proposition 3.2, we can write $w = a^{\sum_{i=1}^{n}(-1)^i x_i} b^n$. By the same reasoning as before, from this form we can see that the functions must vanish on all conjugates of the relations and are therefore well-defined on D_8.

Now suppose that a word w vanishes on A, B, and $[\![A]\!]A + [\![B]\!]A$. Then we have $A(w), B(w) \equiv 0 \pmod 2$. Since there are no a's after b's in the form $w = a^{\sum_{i=1}^{n}(-1)^i x_i} b^n$, we have $[\![A]\!]A(w) \equiv 0 \pmod 2$, which is equivalent to $A(w) \equiv 0 \pmod 4$. Finally, from the relations $a^4 = 1$ and $b^2 = 1$, we obtain $w = 1$, as desired. □

3.4. M_{27}

Consider $M_{27} = \mathbb{Z}_9 \rtimes \mathbb{Z}_3 = \langle a, b \mid a^9 = b^3 = bab^{-1}a^{-4} = 1 \rangle$.

Proposition 3.5. The functions A, B, and $[\![[\![A]\!]A]\!]A + [\![B]\!]A$ solve the word problem for M_{27} going to $\mathbb{Z}_3$.

Proof. One can verify that these functions are well-defined on the relations of M_{27}. Now, we must show that all conjugates of the relations vanish on these functions. In order to do so, we can separate the a's and b's similarly to how we did in Proposition 3.2:

1. We can write $w = ba^{x_1}ba^{x_2} \cdots ba^{x_n}$.
   Using $b^3 = 1$, we can ensure that none of the powers of b are greater than 2. Moreover, from $bab^{-1}a^{-4} = 1$, we have $b = a^4ba^{-1}$, so $b^2 = ba^4ba^{-1}$, meaning that we can ensure all powers of b are at most 1. Lastly, from $bab^{-1}a^{-4} = 1$, we obtain $a = b^{-1}a^4b = b^2a^4b = ba^4ba^3b$, so if w starts with a, we can manipulate it to start with b.

2. We can move a's to the left of b's. Similarly to Lemma 3.3, we can inductively prove that $ba^x = a^{4x}b$ for any $x \in \mathbb{Z}$.

3. We can gather the a's to the left of the word.
   By repeatedly using $ba^x = a^{4x}b$, we can write $ba^{x_1}ba^{x_2}\cdots ba^{x_n}$ as $a^{4x_1+16x_2+\cdots+4^n x_n}b^n$.

So, we can write $w = a^{\sum_{i=1}^{n} 4^i x_i} b^n$ for any w. Since $\sum_{i=1}^{n} 4^i x_i = A(w)$ and $n = B(w)$ are constant across all conjugates of w, and the functions vanish on all relations of M_{27}, they must also vanish on all conjugates of the relations, meaning that they are well-defined.

Now suppose that a word w vanishes on A, B, and $[\![[\![A]\!]A]\!]A + [\![B]\!]A$. Then $A(w), B(w) \equiv 0 \pmod 3$. Moreover, there are no a's after b's, so $[\![B]\!]A = 0$, leaving us with

$$[\![[\![A]\!]A]\!]A = \binom{A(w)}{3} = \frac{A(w)(A(w)-1)(A(w)-2)}{6} \equiv 0 \pmod 3.$$

This is equivalent to $A(w)(A(w)-1)(A(w)-2) \equiv 0 \pmod{18}$; in particular, this product is divisible by 9. Since $A(w) \equiv 0 \pmod 3$, we cannot have $A(w)-1 \equiv 0 \pmod 3$ nor $A(w) \equiv 0 \pmod 3$, so $A(w) \equiv 0 \pmod 9$. Therefore, simplifying with $a^9 = 1$ and $b^3 = 1$ yields $w = 1$, as desired. □

3.5. Finite Abelian 2-groups

We can now proceed to prove that the word problem is solved for all finite abelian 2-groups. First, for $k \in \mathbb{N}$, consider $\mathbb{Z}_{2^k} = \langle a \mid a^{2^k} = 1\rangle$. Firstly, we need a lemma:

Lemma 3.6. As a polynomial, $\binom{n}{2^k} \equiv 0 \pmod 2$ if and only if $n \equiv 0, 1, \ldots, 2^k - 1 \pmod{2^{k+1}}$.

Proof. Note that if $n < 2^k$, the result is true. It suffices to verify the result for positive integers, as $\binom{n}{2^k}$ is polynomial in n and hence eventually periodic modulo 2^k. Henceforth assume $n \geq 2^k$. Recall that the maximum power of 2 dividing $n!$ is $\sum_{i=1}^{\infty} \left\lfloor \frac{n}{2^i} \right\rfloor$. As such the maximum power of 2 dividing $\frac{n!}{(2^k)!(n-2^k)!}$ is

$$\sum_{i=1}^{c} \left(\left\lfloor \frac{n}{2^i} \right\rfloor - \left\lfloor \frac{2^k}{2^i} \right\rfloor - \left\lfloor \frac{n-2^k}{2^i} \right\rfloor \right),$$

where we cut the sum off at the smallest integer c such that 2^c is larger than n. Note that $c > k$ by size of n. We split the sum into two parts; if $i \leq k$, we have

$$\begin{aligned}\sum_{i=1}^{k} \left(\left\lfloor \frac{n}{2^i} \right\rfloor - \left\lfloor \frac{2^k}{2^i} \right\rfloor - \left\lfloor \frac{n-2^k}{2^i} \right\rfloor \right) &= \sum_{i=1}^{k} \left(\left\lfloor \frac{n}{2^i} \right\rfloor - 2^{j-i} - \left\lfloor \frac{n}{2^i} \right\rfloor + 2^{k-i} \right) \\ &= 0.\end{aligned}$$

It suffices to show that for $i > k$,

$$\left\lfloor \frac{n}{2^i} \right\rfloor - \left\lfloor \frac{n-2^k}{2^i} \right\rfloor = 0$$

if and only if $n \equiv 2^k, 2^k + 1 \ldots, -1 \pmod{2^{k+1}}$. However, note that this is the number of multiples of 2^i smaller than n, and if n satisfies the above congruence, then n and $n - 2^k$ have the same number of multiples of 2^{k+1} beneath them. As such for all i larger than $k+1$, the result is true too. Hence for n satisfying these, the binomial coefficient is odd.

Now if $n \equiv 0, 1, \ldots 2^k \pmod{2^{k+1}}$, we have that the number of multiples of 2^{k+1} beneath n is greater than the number of multiples of 2^{k+1} beneath $n - 2^k$, so we have a positive quantity, and the binomial coefficient is even. □

Proposition 3.7. For $k \in \mathbb{N}$, the functions A, $[\![A]\!]A$, $[\![[\![A]\!]A]\!]A$, ..., $\underbrace{[\![\ldots[\![A]\!]A\ldots]\!]A}_{2^{k-1}\ A\text{'s}}$ solve the word problem for $\mathbb{Z}_{2^k}$ going to $\mathbb{Z}_2$.

Proof. Again, we first verify that these functions are well-defined on the relations of $\mathbb{Z}_{2^k}$. Since these functions are periodic with period 2^n for some n less than or equal to 2^k, it follows they are all periodic with larger period 2^k, so they are well defined.

Now suppose a word w vanishes on all of these functions. Then, write $w = a^n$. We will show inductively that $n \equiv 0 \pmod{2^k}$. The base case of $i = 1$ is trivial.

Suppose $n \equiv 0 \pmod{2^{m-1}}$. Then, since $\binom{n}{2^{m-1}} \equiv 0 \pmod 2$, it follows that $n \equiv 0, 1, \ldots, 2^{m-1} - 1 \pmod{2^m}$. However, the only residues modulo 2^m that are compatible with n are 0 and 2^{m-1}, so $n \equiv 0 \pmod{2^m}$.

By induction, it follows that $n \equiv 0 \pmod{2^k}$, and hence $a^n = 1$. □

Lemma 3.8. Given finite abelian groups G, H for which the word problem is solved, the word problem is also solved for $G \times H$.

Proof. Since G and H are abelian, $G \times H$ is also abelian. Then, for each element of $G \times H$, we can separate the generators of G and the generators of H using this commutativity. Moreover, for each word $w = 1$, both parts must be the identity in their respective groups. Since the word problem is solved for G and H, there exist functions that vanish on each part of the word if and only if that part is equivalent to the identity. Then since the two parts of the word are independent, taking these functions together solve the word problem for $G \times H$; the entire set of functions will vanish on a word w if and only if $w = 1$. □

Remark 3.9. It is a important result in group theory, known as the **structure theorem**, that for any abelian group G there exist prime powers $q_1, q_2, \ldots$ and an integer $n \in \mathbb{N}$ such that $G \cong \mathbb{Z}^n \times \mathbb{Z}_{q_1} \times \mathbb{Z}_{q_2} \times \ldots$. If G is finite, then there are no factors of $\mathbb{Z}$; we have $G \cong \mathbb{Z}_{q_1} \times \ldots \times \mathbb{Z}_{q_n}$ for some prime powers $q_1, \ldots, q_n$.

Theorem 3.10. The word problem is solved for all finite abelian 2-groups.

Proof. By the structure theorem, any finite abelian 2-group G satisfies $G \cong \mathbb{Z}_{2^{k_1}} \times \mathbb{Z}_{2^{k_2}} \times \cdots \times \mathbb{Z}_{2^{k_n}}$. From Proposition 3.7 and Lemma 3.8, since the word problem is solved for each of the $\mathbb{Z}_{2^{k_i}}$, we have that the word problem is solved for G. □

3.6. Observations

In all of these examples, there is a common technique: so long as there exists some form of "pseudo-commutativity," such as $ba = a^{-1}b$ in Q_8 and D_8 or $ba = a^4b$ in M_{27}, we can separate the generators and find functions that will vanish on each section if and only if the entire word is the identity. Such a relation must exist for a group to be finite, in order to prevent elements $baba\ldots$ from being irreducible. We wonder whether for any finitely presented finite group, one can construct a set of well-defined letter braiding functions that will solve the word problem for that group simply by looking at the relations (without any computation).

4. Solving the Word Problem with Augmentation Filtration

It turns out that letter braiding functions from a group to a ring exactly describe a core underlying algebraic structure of the group. Using this structure and some ring theory, we will demonstrate that well-defined letter braiding functions solve the word problem for all p-groups.

Definition 4.1 — Let G be a group and R be a ring. We define the **group algebra** RG to be the set of all finite linear combinations $r_1g_1 + \ldots + r_ng_n$, where $r_1, \ldots, r_n \in R$ and $g_1, \ldots, g_n \in G$. The product operation on RG can be defined as the linear extension of the product operation in G, using the usual distributive operations from R. We can say that $G \subseteq RG$ according to the identity $g = eg$ for all $g \in G$, where e is the multiplicative identity in R, and we will write 1 for the identity in RG, which is the product of e and the identity in G.

Definition 4.2 — Let RG be a group algebra. We note that there is a homomorphism $\phi : RG \to R$ such that $\phi(g) = 1$ for all $g \in G$. Let the **augmentation ideal** of RG, written as JG, be the kernel of this map. That is, each element of RG is of the form $\sum r_ig_i$ for $r_i \in R, g_i \in G$, and JG consists of exactly the elements such that $\sum r_i = 0$. Note that $g - 1 \in JG$ for all $g \in G$, since $1 + (-1) = 0$.

Lemma 4.3. *JG is a two-sided ideal. That is, for all $a, b \in JG, c \in RG$, we have $a + b, ac, ca \in JG$.*

Proof. Suppose $a, b \in JG$ and $c \in RG$. Now, $\exists r_1, \ldots, r_n, s_1, \ldots, s_m, t_1, \ldots, t_p \in R$ and $g_1, \ldots, g_n, h_1, \ldots, h_m, i_1, \ldots, i_p \in G$ such that $a = r_1g_1 + \cdots + r_ng_n$, $b = s_1h_1 + \cdots + s_mh_m$, and $c = t_1i_1 + \cdots + t_pi_p$. Since $a, b \in JG$, we also have $r_1 + \cdots + r_n = s_1 + \cdots s_m = 0$.

Notice that the sum of the coefficients of $a + b = r_1g_1 + \cdots r_ng_n + s_1h_1 + \cdots + s_mh_m$ is $r_1 + \cdots + r_n + s_1 + \cdots + s_m = 0 + 0 = 0$, so JG is closed under addition. Also, the sum of the coefficients of $ac = (r_1g_1 + \cdots + r_ng_n)(t_1i_1 + \cdots + t_pi_p)$ is $(r_1 + \cdots + r_n)(i_1 + \cdots + i_p) = 0(i_1 + \cdots + i_p) = 0$, so JG absorbs under right multiplication. Similarly, JG absorbs under left multiplication, so it is an ideal. □

Remark 4.4. We note that the product of two ideals is another ideal, where we define product by $AB = \{a * b : a \in A, b \in B\}$ for all ideals A, B in some ring. This means that we can create an **augmentation filtration** on RG by raising JG to successive powers J^2G, J^3G, etc. Because each power of J^nG is an ideal, as it is a product of ideals, we have $i \leq j \implies J^iG \supseteq J^jG$ for all i, j.

We offer the following example for clarity.

Example 4.1

Let $G = C_4 = \langle a \mid a^4 = 1 \rangle$ and $R = \mathbb{Z}_2$. We have the group algebra $RG = \{0, 1, a, a^2, a^{-1}, 1+a, 1+a^2, 1+a^{-1}, a+a^2, a+a^{-1}, a^2+a^{-1}, 1+a+a^2, 1+a+a^{-1}, 1+a^2+a^{-1}, a+a^2+a^{-1}, 1+a+a^2+a^{-1}\}$.

Now, we see that $JG \subset RG$ is the set of all elements with an even number of terms, so $JG = \{0, 1+a, 1+a^2, 1+a^{-1}, a+a^2, a+a^{-1}, a^2+a^{-1}, 1+a+a^2+a^{-1}\}$. Computing all multiplications, we have that $J^2G = \{0, 1+a^2, a+a^{-1}, 1+a+a^2+a^{-1}\}$, $J^3G = \{0, 1+a+a^2+a^{-1}\}$, and $J^4G = \{0\}$. We also note that for $n > 4$, $J^nG = J^4G = \{0\}$.

The following is a result in ring theory:

Lemma 4.5. Let $JG \supseteq J^2G \supseteq \cdots$ be the augmentation filtration of a group G with order p^k for some prime p over the ring $\mathbb{Z}_p$. This filtration terminates in finitely many steps with the zero ideal.

Remark 4.6. Note that any J^nG consists of linear combinations $\sum r_i g_i$ of elements of G with scalars in R. We may extend any letter braiding function $f : G \to R$ to a letter braiding function $f : J^nG \to R$, where $f\left(\sum r_i g_i\right) = \sum r_i f(g_i)$.

Now, we are ready to connect letter braiding functions to the augmentation filtration. The following theorem is due to Nir Gadish, one of the creators of letter braiding functions, in the paper [1]. The proof of this statement is omitted, as it involves a significant amount of topological and algebraic machinery.

Theorem 4.7. Let RG be a group algebra with R a PID domain, and let $G = \mathcal{D}_1(G) \trianglerighteq \mathcal{D}_2(G) \trianglerighteq \cdots$ be the augmentation filtration of G over R. For all i, the set of homomorphisms $J^{i+1}G/J^{i+2}G \to R$ is exactly the set of well-defined weight i letter braiding functions $G \to R$.

That is, consider some weight i letter braiding function $f : G \to R$. This function extends to a letter braiding function $f : RG \to R$ as previously mentioned. Now, there exists some $\phi \in \mathrm{Hom}(J^{i+1}G/J^{i+2}G \to R)$, such that for all $a \in J^{i+1}G$, $(\phi \circ \pi)(a) = f(a)$, where π is the quotient map $J^{i+1}G \to J^{i+1}G/J^{i+2}G$.

Conversely, for every $\phi \in \mathrm{Hom}(J^{i+1}G/J^{i+2}G \to R)$, there exists some letter braiding function $f : G \to R$ (which extends to $f : RG \to R$) such that for all $a \in J^{i+1}G$, $(\phi \circ \pi)(a) = f(a)$.

This theorem will allow us to solve the word problem for all p-groups, since their augmentation filtration over $\mathbb{Z}_p$ terminates. First, we need several lemmas.

Lemma 4.8. Let $n, k \in \mathbb{N}$. The only element $a \in \mathbb{Z}_n^k$ such that $\phi(a) = 0$ for all $\phi \in \text{Hom}(\mathbb{Z}_n^k, \mathbb{Z}_n)$ is the identity $0 \in \mathbb{Z}_n^k$.

Proof. Suppose $a \in \mathbb{Z}_n^k$ is nonzero. We may write $a = (x_0, \ldots, x_k)$ for $x_0, \ldots, x_k = \mathbb{Z}_n$. Suppose $a \in \mathbb{Z}_n^k$ is nonzero, so $\exists x_i$ nonzero. Now let $\pi : \mathbb{Z}_n^k \to \mathbb{Z}_n$ be the projection map preserving the ith coordinate of each vector in $\mathbb{Z}_n^k$. We have $\pi(a) = x_i \neq 0$. Thus, if $\phi(a) = 0$ for all $\phi \in \text{Hom}(\mathbb{Z}_n^k, \mathbb{Z}_n)$, $a = 0$. □

Lemma 4.9. Let RG be the group algebra of a p-group G over $\mathbb{Z}_p$. Every ideal $I \subseteq RG$ is isomorphic to $\mathbb{Z}_p^k$ for some k.

Proof. Since Z_p is a field, RG is a vector space with basis G over Z_p. Thus it is isomorphic to $Z_p^{|G|}$. Now, any ideal $I \subseteq RG$ is a subspace of $Z_p^{|G|}$, meaning it must be isomorphic to Z_p^k for some k. □

Corollary 4.10. Let $JG \supseteq J^2G \supseteq \cdots$ be the augmentation filtration of a p-group G over $\mathbb{Z}_p$. For all i, $J^iG/J^{i+1}G \cong \mathbb{Z}_p^k$ for some k.

Now, we are ready to prove that letter braiding functions solve the word problem for p-groups.

Theorem 4.11. Let G be a p-group. Suppose $L(G, \mathbb{Z}_p)$ is the set of all well-defined letter braiding functions from G to $\mathbb{Z}_p$. For any word $w \in G$, $f(w) = 0$ for all $f \in L(G, \mathbb{Z}_p)$ if and only if w is the identity in G.

Proof. By construction, all functions in $L(G, \mathbb{Z}_p)$ vanish if w is the identity. Therefore we just need to prove the converse.

For any i, let $L_i(G, \mathbb{Z}_p)$ be the set of all well-defined weight i letter braiding functions $G \to \mathbb{Z}_p$. Suppose $w \in G$ such that $f(w) = 0$ for all $f \in L(G, \mathbb{Z}_p)$, so by extending f linearly to $f : \mathbb{Z}_pG \to \mathbb{Z}_p$ we have $f(w-1) = f(w) - f(1) = f(w) = 0$ for all $f \in L(\mathbb{Z}_pG, \mathbb{Z}_p)$. Finally, let $J^1G \supseteq J^2G \supseteq \cdots$ be the augmentation filtration of G over $\mathbb{Z}_p$. We will proceed by induction on J^iG; as a base case, note that we have $w - 1 \in JG$ since $w \in G$.

Now, suppose $w - 1 \in J^{i+1}G$ for some $i \in \mathbb{N} \cup \{0\}$. Recall from Theorem 4.7 that $L_i(G, \mathbb{Z}_p) = \text{Hom}(J^{i+1}G/J^{i+2}G, \mathbb{Z}_p)$, so because $f(w - 1) = 0$ for all $f \in L_i(\mathbb{Z}_pG, \mathbb{Z}_p)$, we have $(\phi \circ \pi)(w - 1)\phi(\pi(w - 1)) = 0$ for all $\phi \in \text{Hom}(J^{i+1}G/J^{i+2}G, \mathbb{Z}_{|G|})$ where $\pi : J^{i+1}G \to J^{i+1}G/J^{i+2}G$ is the quotient map.

Recall that $J^{i+1}G/J^{i+2}G \cong \mathbb{Z}_p^k$ for some k by Corollary 4.10, so because $\phi(\pi(w-1)) = 0$ for all $\phi \in \text{Hom}(J^{i+1}G/J^{i+2}G, \mathbb{Z}_p) = \text{Hom}(\mathbb{Z}_p^k, \mathbb{Z}_p)$ we know that $\pi(w - 1)$ is the identity in $J^{i+1}G/J^{i+2}G$ by Lemma 4.8. Thus $\pi(w-1) = J^{i+2}G$, meaning that $w-1 \in J^{i+2}G$.

By induction, $w - 1 \in J^k G$ for all k, and since G is a p-group, there exists some n such that $J^n G = \{0\}$ by Lemma 4.5. Thus, $w - 1 = 0$, so $w = 1$, the identity in G. □

5. Conclusion

Though we have explored the word problem, we are still curious about the underlying structures which can reveal exactly which letter braiding functions will solve the word problem. Many questions arise which we hope to explore in the future, such as: How do letter braiding functions follow from relations? Can we look at a presentation and immediately see the well-defined letter braiding functions? How do letter braiding functions and relations encode each other? We look forward to further investigation into these fascinating functions, particularly with regard to understanding their topological roots.

6. Appendices

A. Letter Linking

It turns out that there are yet more interpretations of letter braiding. One such interpretation is **letter linking**, which is a less general version of letter braiding. This has roots in topology, and is detailed more in [3].

For a given word, we can consider a **list** of letters. This is a multiset of ordered letters as defined by Definition 2.21, except we make them into 4-tuples $(n, \varepsilon_n, s_n, \sigma_n)$, with the last element being ± 1, arbitrarily. The value of the fourth element is called the **extrinsic sign**; the point of this will become clear later.

We say two lists for a given word are **simply equivalent** if we can remove or add pairs of tuples which are identical, but have opposite extrinsic sign. Also the standard list for a generator a, denoted Λ_a, just takes each letter with s_i matching a exactly once, all with extrinsic sign $+1$. As such we say a list is **homogeneous** if all letters come from the same generator. We let L_a denote an arbitrary homogeneous list with generator a.

We define the **total sign** for an element $\ell = (n, \varepsilon_n, s_n, \sigma_n)$ in a list L to be $\varepsilon_n \cdot \sigma_n$, and denote it by $\operatorname{sign}(\ell)$. We can filter to a generator a, where we set $\operatorname{sign}_a(\ell) = 0$ if s_n is not the desired letter. The a-count of a list L is now

$$\sum_{\ell \in L} \operatorname{sign}_a(\ell) = 0.$$

Now an **interval** I is a nonempty set of consecutive letters. Following this, we define **oriented intervals,** which have first and last ordered letters $\partial_0 I$ and $\partial_1 I$ respectively. The signs of the endpoints, denoted σ_0^I and σ_1^I, is simply the total sign of $\partial_0 I$ and $\partial_1 I$ respectively. We require the endpoint signs to differ. Thus we assign a direction to intervals, which will be important later.

Given a generator a and a list L whose a-count is 0, we define a **cobounding** $d^{-1}L$ to be a set of disjoint oriented intervals $\{I_k\}$ such that each element of L occurs exactly once as one of the endpoints of some I_k. Given a letter ℓ and an interval I, we define the **signed intersection** $\ell \cap_{\rightarrow} I$ to be empty if ℓ is not contained in I, and otherwise ℓ with extrinsic sign multiplied by σ_0^I.

For fixed distinct generators a and b, and L a list for which $d^{-1}L_a$ exists, we define a **linking list** $d^{-1}L_a \wedge L_b$ as the union of all $\ell \cap I_k$, where $\{I_k\}$ ranges over the cobounding of L_a and ℓ ranges over L_b. That is, all of the b's, but with extrinsic sign 'relative' to the orientation of the interval. We can obtain the b-count of this interval, and if it's zero, we can cobound it and link it with another list as much as we want.

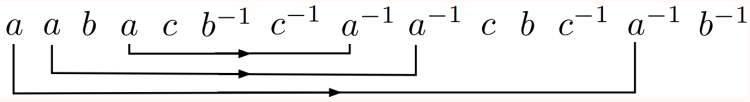
Algorithmically, this process resembles the following procedure. We take the example from [3].

Example 6.1

We will compute the a-count of $d^{-1}(d^{-1}\Lambda_a \wedge \Lambda_b) \wedge \Lambda_a$ on the following word. Start with

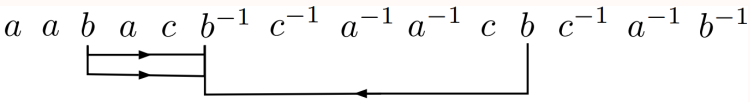
$$aabacb^{-1}c^{-1}a^{-1}a^{-1}cbc^{-1}a^{-1}b^{-1}.$$

Firstly, we pair a and a^{-1} arbitrarily, and create intervals. The orientation is from the positive sign to the negative sign, as follows:



For each b, we count the number of intervals passing under it; $+1$ for each positively oriented interval, -1 for each negatively oriented interval.

Using these counts, we create a 'cobounding' of b's using these numbers; for each b with number n, it must be a boundary of exactly n oriented intervals. This gives the following count:



Finally, we count the a's in each of these intervals, accounting for sign. In order, we obtain $2 - (-1) - (-1) = 4$.

It should be noted that there is some analog between $[\![[\![A]\!]B]\!]A$ and this. In fact, the evaluation of this letter linking function is 4 on this word. We presume that this is where these functions came from, but apparently the topological uses are much deeper.

Even more, the use of these functions are limited to things that vanish. However, we note that using k coboundings makes these vanish on γ_n (see 4.11) for $n \leq k$, but not necessarily γ_{k+1}, so there is some use in this regard.

B. Graphical Approach

There's actually a strange way to compute functions like $[\![A]\!]B$.

Definition 6.1 — An **oriented curve with boundary** is a curve with a boundary which can be totally ordered. We let the negative end be the smallest endpoint and the positive end be the largest endpoint.

The canonical way to orient curves is by parameterizing them, with the negative and positive ends being the obvious ones.

We can give each curve a **labeling** at each endpoint, which is just an element of some fixed ring. The curve's **weight** is then the label at the positive end minus the label at the negative end.

Definition 6.2 — For a word w with length n, fix a circle with n equally spaced points on it, and label one point with the first letter of w, and continue labelling going counterclockwise. We call these **letter points**.

If h is a homomorphism to R which vanishes on w, then a **graph realization** R_h of h is the finite union of smooth oriented curves, labeled such that

1. The sum of weights of curves with a boundary at a fixed letter point ℓ is $h(\ell)$,
2. The sum of weights of curves at any other boundary point is 0.

Particularly, multiple curves can end at a letter point.

Now for homomorphisms g and h that vanish on w, consider two graph realizations of g and h on the word w.

Definition 6.3 — Two oriented curves C_1 and C_2 in the plane intersect **n**eatly at some point if their tangent vectors are perpendicular at that point.

The **s**ign of a neat intersection point is $+1$ if the tangent vector of C_1 and that of C_2 can be rotated and translated together to the origin where C_1 is on the positive x-axis, and C_2 is on the positive y-axis. Otherwise, the sign is -1.

If C_1 and C_2 are weighted, define the **v**alue of their intersection point to be its sign multiplied by the product of the labels of the curve.

We state the following proposition without proof.

Proposition 6.4. Set up graph realizations as above. Then the value of $\llbracket g \rrbracket h$ is precisely equal to the sum of the value of intersections of curves of R_f and R_g.

The current proof of this requires significant topological machinery. However, we hope to find a more elementary proof.

References

[1] Nir Gadish. *Letter-braiding: a universal bridge between combinatorial group theory and topology*. 2023. arXiv: 2308.13635 [math.GR]. URL: https://arxiv.org/abs/2308.13635.

[2] Nir Gadish et al. *Infinitesimal calculations in fundamental groups*. 2024. arXiv: 2403.20264 [math.AT]. URL: https://arxiv.org/abs/2403.20264.

[3] Jeff Monroe and Dev Sinha. *Linking of letters and the lower central series of free groups*. 2022. arXiv: 2006.00989 [math.GR]. URL: https://arxiv.org/abs/2006.00989.