The Word Problem in Quandles

Benjamin Fish
Advisor: Rena Levitt

April 5, 2013

1 Introduction

A word over an algebra A is a finite sequence of elements of A, parentheses, and operations of A defined recursively: Given each n-ary operation $\circ$ of A, if a_1, a_2, ..., a_n are words, then $\circ(a_1, a_2, \ldots, a_n)$ is also a word. The word problem over an algebra A is the following decision problem:

Question. Given a pair of words w_1,w_2 over A, is $w_1 = w_2$ in A?

The related isomorphism problem is the decision problem that asks if two algebras are isomorphic.

The word problem was first introduced by Dehn in 1911. He proved that it was solvable for some groups, but it was proved undecidable in the general case by Novikov in 1955 [7]. Since then, the word problem has been proved either decidable or undecidable for many algebras, including many types of groups [2] [4]. See [8] for more. However, far littler is known about the word problem in quandles.

A quandle is an algebraic structure that arises from the Reidemeister moves for knots. It is a complete invariant of knots, but the isomorphism problem is extremely difficult in quandles. The word problem over quandles is a step in the direction of the isomorphism problem. While the word problem in general is undecidable in quandles, the word problem for free quandles (analogous to free groups) is decidable.

In 1984, Winker introduced a notion of Cayley graphs for quandles [10]. This was motivated by the use of Cayley graphs for groups. A Cayley graph of a group is a representation of that group that depicts its structure: There is a vertex for every element in the group and an edge from elements x to y labeled with a generator g of that group if $xg = y$ [5]. Cayley graphs for groups are important for the following reason:

Theorem. *The word problem for a group with a generating set S is solvable if and only if there exists an algorithm to construct any finite portion of the Cayley graph of the group with generating set S.*

Thus a good translation of Cayley graphs for groups into one for quandles may help solve the word problem in quandles.

In this paper, we improve the algorithm introduced by Winker to create Cayley graphs for quandles. We guarantee that finite quandles' Cayley graphs are constructed in finite time, and that every element in the quandle is included in the graph, two guarantees that the original algorithm does not make. Using this improved algorithm, we are able to 1) classify certain classes of quandles and 2) solve the word problem for certain classes of quandles. In doing so, we introduce a technique that can be used to solve the word problem for other quandles. Those classes of quandles include quandles made from Coxeter groups, and quandles with a single relation. Coxeter groups and single-relator quandles were chosen because of their nice presentations and the fact that the word problem in Coxeter groups and single-relator groups is solved [8] [9]. Our main results are the following: First, an algorithm for constructing Cayley graphs that satisfies all of our desired properties. It always constructs the Cayley graph, and will do so in finite time if the quandle is finite. Furthermore, if it constructs any finite portion of the graph in finite time for a given quandle, it is guaranteed to solve the word problem for that quandle. Our second main result is that all one-relator involutory quandles on two generators have a solvable word problem.

Section 1.1 introduces quandles and quandle presentations. Section 1.2 introduces Cayley graphs for quandles. Section 2 gives an improved algorithm for finding Cayley graphs for quandles, and then proves its correctness and effectiveness. Section 3 gives some examples of quandle presentations, and how to use the algorithm to classify and solve the word problem. Specifically, it introduces a translation of groups to quandles, and solves the word problem for the quandles associated with certain Coxeter groups. In Section 3.1, we classify one-relator involutory quandles on two generators, and give more general results on how to solve the word problem in other quandles using the algorithm.

1.1 Quandles and Presentations

A quandle is defined as the following:

Definition 1. A quandle $(Q, *, /)$ is a set Q together with two binary operations $*, /$ such that
I. Idempotence: $x * x = x$
IIA. Right Cancellation A: $(x * y)/y = x$
IIB. Right Cancellation B: $(x/y) * y = x$
III. Right Self-Distributivity: $(x * y) * z = (x * z) * (y * z)$
for all $x, y, z \in Q$.

We call the two operations multiplication and division respectively. Despite what these names might imply, the two operations are not at all similar to the multiplication and division found in the integers and are simply called so for the sake of convenience.

As a slight abuse of notation, we typically call Q the quandle. Note that due to cancellation, the $/$ operation is uniquely defined by the $*$ operation. In other words, we only need to define the $*$ operation to well-define a quandle.

If an algebraic structure satisfies only axioms II and III, it is called a rack. If the two operations $*$ and $/$ define the same operation, then it is called an involutory quandle. That is,

Definition 2. An involutory quandle $(Q, *)$ is a set Q together with a binary operation $*$ such that
I. Idempotence: $x * x = x$
II. Right Cancellation: $(x * y) * y = x$
III. Right Self-Distributivity: $(x * y) * z = (x * z) * (y * z)$
for all $x, y, z \in Q$.

It is always possible to left-associate words in a quandle. If a word is left-associated, and no idempotence or right cancellation rule can be applied, it is said to be in normal form. For example,

$$a * ((b * c) * a) = ((((a * a) * c) * b) * c) * a = (((a * c) * b) * c) * a.$$

When written in left-associated form, in involutory quandles and only when it is clear, we typically write it without explicitly showing the operations. For example, $(((a * c) * b) * c) * a$ may be written as $acbca$.

There are many ways to make quandles, including from groups. The following are a few examples:

Example 3. Let $(\mathcal{G}, \circ)$ be a group. Then $(\mathcal{G}, *, /)$ where $*$ is defined as $x * y \triangleq y^{-1}xy$ and $/$ is defined as $x/y \triangleq yxy^{-1}$ is a quandle called the conjugation quandle of $\mathcal{G}$.

Proof. It is not difficult to prove that this is a quandle by showing that it satisfies the axioms of a quandle:

I. $x * x = x^{-1}xx = x$.

II. $(x * y)/y = (y^{-1}xy)/y = y(y^{-1}xy)y^{-1} = x$.

III. $(x/y) * y = (yxy^{-1}) * y = y^{-1}(yxy^{-1})y = x$.

IV.

$$\begin{aligned}(x * y) * z &= (y^{-1}xy) * z \\ &= z^{-1}(y^{-1}xy)z\end{aligned}$$

and

$$\begin{aligned}(x * z) * (y * z) &= (z^{-1}xz) * (z^{-1}yz) \\ &= (z^{-1}yz)^{-1}(z^{-1}xz)(z^{-1}yz) \\ &= (z^{-1}y^{-1}z)(z^{-1}xz)(z^{-1}yz) \\ &= z^{-1}y^{-1}xyz.\end{aligned}$$

Then $(x * y) * z = (x * z) * (y * z)$. □

Another way to create quandles from groups is through the cyclic group:

Example 4. Let $\mathbb{Z}_p$ be the cyclic group of order p. Then the dihedral quandle $(\mathbb{Z}_p, *, /)$ is the quandle defined by $x * y \triangleq 2y - x \pmod p$ and $x/y \triangleq 2y - x \pmod p$.

Note this is an involutory quandle.

For more examples, we consider quandle presentations, analogous to group presentations: A quandle presentation is a set of generators G and a set of relations R, written $\langle G|R\rangle$. This is a quandle whose underlying set are equivalence classes of words. To define this, we first need to define some useful sets of words (languages), relations, and how to derive new relations from them.

First, an alphabet is a set of symbols. If we let A be the alphabet consisting of the symbols $(,), *, /$ and elements of a set G, then a word over A is a sequence of those symbols. Since we don't want nonsense words, we define a couple of useful languages:

Definition 5. $L(G)$ is the language over A defined recursively in the following manner:

1. For every g in G, g is in $L(G)$.
2. Given two words x and y in $L(G)$, $(x) * (y)$ and $(x)/(y)$ are also in $L(G)$.

Typically, parentheses are omitted when it is clear they are unnecessary. For example, given a, b in G, $(a) * (b)$ is typically written $a * b$.

As mentioned above, words of a quandle have a normal form:

Definition 6. $N(G)$ is defined as the subset of $L(G)$ where x is in $N(G)$ if

$$x = (\ldots((g_1 \circ g_2) \circ g_3) \circ \ldots \circ g_{n-1}) \circ g_n$$

for g_i in G and $\circ$ in $\{*, /\}$, such that $g_1 \neq g_2$ and x contains no subexpressions of the forms $(y * g_i)/g_{i+1}$ or $(y/g_i) * g_{i+1}$ where y is a subexpression and $g_i = g_{i+1}$.

There is an onto normalization function $n : L(G) \to N(G)$. This function is defined by repeated uses of the axioms to first get an arbitrary word into left-associated form, and then use the first two axioms to cancel any repeated generators. For this paper, it will also be useful to consider words that have been left-associated and right-cancelled, but idempotence has not been 'used'. We will call this form canonicalization:

Definition 7. $C(G)$, the set of canonicalized words, is defined as the subset of $L(G)$ where x is in $C(G)$ if

$$x = (\dots((g_1 \circ g_2) \circ g_3) \circ \dots \circ g_{n-1}) \circ g_n$$

for g_i in G and $\circ$ in $\{*, /\}$, such that x contains no subexpressions of the forms $(y * g_i)/g_{i+1}$ or $(y/g_i) * g_{i+1}$ where y is a subexpression and $g_i = g_{i+1}$.

Given a word x in $L(G)$, we write x' to be the canonicalized form of x.

Now we define how to derive relations over these language from other relations:

Definition 8. A derivation $\Delta(X)$ over a language L and set of relations R is a finite sequence of relations in L such that each relation $p_i = q_i$ can be derived using the previous relations $p_1 = q_1, p_2 = q_2, \dots p_{i-1} = q_{i-1}$ in the sequence and R using a set of rules X.

In this paper, we will use a few different rule sets. To describe a derivation, it suffices to describe the set of rules and the language L. For a language L, we call $X(R)$ the set of relations for which there is a derivation $\Delta(X)$ that includes that relation.

Example 9. We call D the type of derivation, over the language $L(G)$ for some set of generators G, under the following rules:

Reflexivity: For any word w in $L(G)$, derive $w = w$.

Symmetry: Given a relation $p = q$ derivable from R, derive $q = p$.

Transitivity: Given relations $p = q$ and $q = r$ derivable from R, derive $p = r$.

Right Multiplication: Given a relation $p = q$ derivable from R, and for any word w in $L(G)$, derive $p * w = q * w$.

Left Multiplication: Given a relation $p = q$ derivable from R, and for any word w in $L(G)$, derive $w * p = w * q$.

Using D and a given set of relations, we can derive new relations. For example, using the relation $a*b = c$ and the relations deriving from the quandle axioms, we can derive $(b*(a*b))*(c*(a*b)) = b$:

1. $a * b = c$ (primary relation)
2. $(b * c) * (a * b) = (b * c) * c$ (left multiplication of 1. by $b * c$)
3. $(b * c) * c = b$ (involutory quandle axiom)
4. $(b * c) * (a * b) = b$ (transitivity of 2. and 3.)
5. $(b * (a * b)) * (c * (a * b)) = (b * c) * (a * b)$ (involutory quandle axiom)
6. $(b * (a * b)) * (c * (a * b)) = b$ (transitivity of 4. and 5.)

This is then a derivation $\Delta(D)$ over the language $L(G)$ and $R = A_G \cup \{((a * b), c)\}$, where $G = \{a, b, c\}$ and A_G is defined below.

Definition 10. Given a generating set G, A_G is the set of relations of the following forms:
$x * x = x$
$(x * y) * y = x$
$(x * y) * z = (x * z) * (y * z)$ for x, y, z in $L(G)$.

We now have the tools to define a presentation of a quandle. We start with the presentation of an involutory quandle:

Definition 11. The involutory quandle given by the presentation $\langle G|R\rangle$ is defined to be $L(G)/D(R \cup A_G)$.

That is, $\langle G|R\rangle$ is the underlying set $L(G)$ where equality of words is given by the set of relations $D(R \cup A_G)$. It is not difficult to check that this is an involutory quandle. In the non-involutory case, we need to use the non-involutory quandle axioms instead:

Definition 12. The quandle given by the presentation $\langle G|R\rangle$ is defined to be $\mathcal{L}(G)/D(R \cup \mathcal{A}_G)$ where $\mathcal{L}$ is L with the addition of right division and left division and $\mathcal{A}_G$ is the set of relations of the following forms:
$x * x = x$
$(x * y)/y = x$
$(x/y) * y = x$
$(x * y) * z = (x * z) * (y * z)$ for x, y, z in $\mathcal{L}(G)$.

See [3] or [6] for other equivalent constructions. This will be our main source of quandles.

It is possible to create quandle presentations from knots. The underlying set will be the arcs of a given diagram for the knot, and the operations are defined by the crossings:

Example 13. Let K be a diagram of an oriented knot. Let A be the set of arcs of K. Let R be the following set of relations: Where b is the overcrossing arc and a and c are the undercrossing arcs, let $a/b = c$ be in R if it is a right-handed crossing and let $a * b = c$ be in R if it is a left-handed crossing.

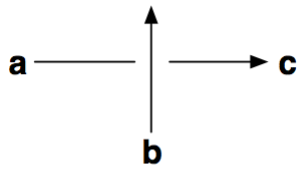

Figure 1: Left-handed crossing

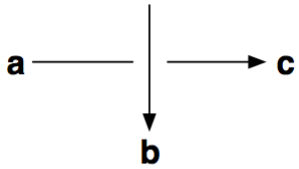

Figure 2: Right-handed crossing

Then the knot quandle is the quandle given by the presentation $\langle A|R\rangle$.

It turns out that this quandle is a complete invariant of knots. This is a primary motivation for the use of quandles. In fact, the quandle axioms come directly from the Reidemeister moves: In

order to prove that this quandle is an invariant, it must be invariant over the Reidemeister moves, which in turn means that any algebraic object created in this manner must satisfy a number of axioms, which just so happen to be the quandle axioms.

The following is an example of an involutory knot quandle.

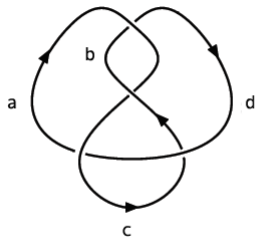

Figure 3: The figure eight knot

Example 14. First we label the arcs of a figure eight knot, as shown in Figure 3. Then the involutory quandle presentation is $\langle a, b, c, d | a * b = c, c * d = b, b * a = d, d * c = a \rangle$. Note since this is for an involutory quandle, / is defined to be $*$ and the handedness of each crossing does not matter.

Written as a Cayley table (a multiplication table), the quandle looks like:

$*$	a	b	c	d	$a * d$
a	a	c	d	$a * d$	b
b	d	b	$a * d$	c	a
c	$a * d$	a	c	b	d
d	b	$a * d$	a	d	c
$a * d$	c	d	b	a	$a * d$

For each row i and column j the entry at (i, j) is $i * j$.

Remember that a quandle is uniquely defined by its $*$ operation, so there is no need to give a table for its / operation.

Or viewed as a Cayley graph, which will be discussed in more detail later, it looks like Figure 4.

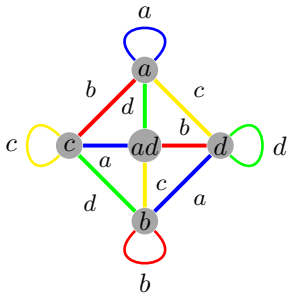

Figure 4: Involutory Cayley Graph for $\langle a, b, c, d | a * b = c, c * d = b, b * a = d, d * c = a \rangle$

But how do we know that those are the only elements of the quandle? How do we know that two of those elements aren't actually the same? Formally, we want to solve the word problem:

Definition 15. Let $Q =< G|R >$. The word problem over Q is the following decision problem:
Instance: A pair of words w_1, w_2 in $L(G)$.
Question: Is $w_1 = w_2$ in Q?

This is equivalent to asking if the relation $w_1 = w_2$ is in a derivation $\Delta(D)$ over $L(G)$ and the relations R.

We will need additional tools to be able to answer this question.

We will attempt to answer this question for a variety of quandle presentations.

1.2 Cayley Graphs

In order to better determine what a quandle looks like from its presentation, it will be helpful to depict what the generators do to the words.

Definition 16 (Winker)**.** The Cayley graph of a quandle $Q = \langle G|R\rangle$ is a directed, labeled graph associated with Q such that there is exactly one vertex labeled x for every x in Q and whenever $x * g = y$ for elements $x, y \in Q$ and generator $g \in G$, there is a directed edge from x (the vertex labeled x) to y labeled g.

Note we say that a word x labels a vertex if x in canonical form can be found at that vertex by following the appropriate edges. $x * g$ is found by following the directed edge from x labeled g. x/g is found by following the directed edge to x labeled g.

In the case of an involutory quandle, we can use undirected edges instead, because $x * g = y$ implies $y * g = x$:

Definition 17 (Winker)**.** The Cayley graph of an involutory quandle $Q = \langle G|R\rangle$ is an undirected, labeled graph associated with Q such that there is exactly one vertex labeled x for every x in Q and whenever $x * g = y$ for elements $x, y \in Q$ and generator $g \in G$, there is an edge between x (the vertex labeled x) and y labeled g.

The construction of these graphs can very difficult and may not always be possible. This directly relates to the difficulty of the word problem. When the word problem isn't solvable, it is not possible to create its Cayley graph, because if it were then you could use the graph to solve the word problem. Nevertheless, the following conditions hold in Cayley graphs:

1. For every $g \in G$, there is a vertex labeled g (with an edge labeled g that directs to itself.)
2. For every vertex q and $g \in G$, there is exactly one edge labeled g directed from q and exactly one edge labeled g directed to q. (Note in the case of involutory quandles this just means that there is exactly one edge labeled g at q.)

You can see that these conditions hold in the example given in Figure 5. Note it is easy to determine what element of the quandle any word over the generators is equal to by converting the word to normal form and then following the edges, starting at the first generator in the word.

It is not easy to tell, however, if this graph is in fact the Cayley graph for $\langle a, b|ababa = b, babab = a\rangle$. Namely, are there any elements of the quandles that appear at multiple vertices? Winker describes an algorithm that ensures the graph created is in fact the Cayley graph: no elements are duplicated in the graph [10].

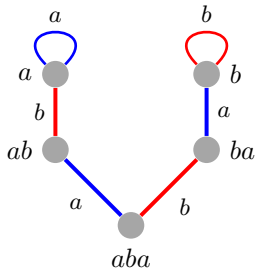

Figure 5: Involutory Cayley Graph for $\mathrm{IQ}(I_3) = \langle a, b | ababa = b, babab = a \rangle$

2 Constructing Cayley Graphs

We need to ensure that the relations R given in a presentation $\langle G|R\rangle$ are reflected in a graph. Unfortunately, there are other so-called secondary relations that can be derived from R and the quandle axioms that may not be reflected in the graph that the algorithm also must take into account. For example, in a quandle with the relation $(a * b) * a = b$, even if this relation is reflected in a graph by a path from a to b through $a * b$, other relations that can be derived from it are not reflected by this path, such as $b * ((a * b) * a) = b$ or $((a * b) * a) * (a * b) = b * (a * b)$. The question is then what relations you need to consider to make sure all possible relations derivable are reflected in the graph. It turns out that two kinds of other relations are sufficient:

Definition 18. Let $Q = \langle G|R\rangle$. Let $S(R)$, the secondary relations of R, be the set of relations of the form $x * s = x * t$ for all x in $C(G)$ and relations $s = t$ in R.

Definition 19. Let $Q = \langle G|R\rangle$. Let I_G, the idempotency relations, be the set of relations of the form $g * g = g$ for all g in G.

Using these relations, we can create the Cayley graph of a quandle $\langle G|R\rangle$ by adding and merging vertices in a partially finished graph until there are no more relations to consider. The proof of its correctness will be given for the Cayley graphs of involutory quandles, but the general proof for non-involutory quandles follows in the same manner. The algorithm when the quandle is involutory is as follows:

Algorithm 20. Let $\langle G|R\rangle$ be a quandle presentation where $|G|, |R| < \infty$. If $|R| = 0$, i.e. if $\langle G|R\rangle$ is the free quandle, then to simplify the algorithm and its proof we set R to be $\langle G|g = g\rangle$ for some g in G. (The proof that all elements of the quandle are represented in the quandle relies on the existence of at least one relation, and this added relation does not change the quandle.)

Start with the partial graph consisting of a vertex labeled g for every g in G. Then trace the relations $R \cup I_G$. To trace a relation is to do the following:

For a given relation $a_1a_2 \dots a_i = b_1b_2 \dots b_j$, If $a_1a_2 \dots a_i$ or $b_1b_2 \dots b_j$ are not in the graph, create the necessary edges and vertices. That is, create the vertex a_1a_2 and the edge labeled a_2 from a_1 to a_1a_2, etc.

Then merge the two vertices $a_1a_2 \dots a_i$ and $b_1b_2 \dots b_j$ by removing one of them and setting all the edges that used to go from that vertex to the other.

Then repeat the following until there are no more vertices to merge: For any newly merged vertices p and q and for all g in G, merge the vertices $p * g$ and $q * g$ if both vertices exist in the partial graph.

After $R \cup I_G$ has been traced, we will trace a subset, which we will call T, of $S(R)$ in the following manner: For each relation $p = q$ in R, trace $x * p = x * q$ if x is a label of a vertex in the graph or $x = y * g$ for some y which is a label of a vertex and all g in G. (Note that it is not necessary to trace $x * p = x * q$ if it has already been traced.) After these relations have been traced, there may be new vertices in the graph. Repeat tracing this set of secondary relations until there are no new vertices in the graph.

Note that this algorithm will not terminate if the quandle is infinite. However, the algorithm will nevertheless be helpful in showing what the structure of these infinite quandles is.

Now, in the general case when the quandle isn't necessarily involutory, we modify this algorithm in the following ways: The algorithm instead creates directed edges. When two vertices labeled p and q are merged, now not only $p * g$ and $q * g$ need to be merged for each g in G, but p/g and q/g also need to be merged. Similarly, the secondary relations $x * p = x * q$ and $x/p = x/q$ need to be traced for each primary relation $p = q$ where x is a label of a vertex or $x = y * g$ for some y which is a label of a vertex and all g in G.

Example 21. As an example, we show how the algorithm runs on the involutory quandle presentation $\langle a, b | a * b = a \rangle$. First, we create a vertex for every generator, as shown in Figure 6.

Figure 6: Partial Cayley Graph for $\langle a, b | a * b = a \rangle$ with the generators

Then the relations I_G are traced, as shown in Figure 7.

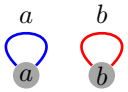

Figure 7: Partial Cayley Graph for $\langle a, b | a * b = a \rangle$ with the generators and I_G

After, the primary relation is traced, as shown in Figure 8.

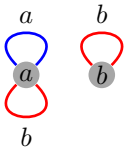

Figure 8: Partial Cayley Graph for $\langle a, b | a * b = a \rangle$ with the generators, I_G, and the primary relations

Then, the secondary relations are traced. There are two vertices, a and b, and there is one word, $b * a$, where $b * a = x * g$ where x is either a or b and g is a generator, but $x * g$ is not already in the graph. Then the secondary relations that are traced are $a * (a * b) = a * a$, $b * (a * b) = b * a$,

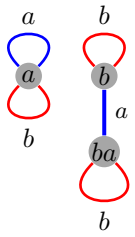

Figure 9: Cayley Graph for $\langle a, b | a * b = a \rangle$

and $(b * a) * (a * b) = (b * a) * a$. Converting these to canonical form, we get $abab = a$, $bbab = ba$, and $babab = b$. These are then traced, resulting in Figure 9.

Since the secondary relations have already been traced for every vertex in the graph and for every $x * g$, where x is a label of a vertex and g is a generator, $x * g$ is already labeled by a vertex in the graph, the algorithm terminates. Then Figure 9 is the Cayley graph for $\langle a, b | a * b = a \rangle$.

We prove that the algorithm works in the involutory case. In order to prove this algorithm works, we will need to show that any relation that can be derived from the presentation is reflected in the graph and that there is one and only one vertex for every distinct element in the quandle. We start with the first of these.

To do this, we show which relations are automatically implied by tracing some given set of relations:

Proposition 22. *Let* Π *be the partial graph given by using the above algorithm to trace each relation in some set* R *over the generators* G*. Two words* v *and* w *label the same vertex if and only if the relation* $v = w$ *is in* $D_r(R)$ *over the language* $C(G)$*, where* D_r *is the following set of rules:*

Reflexivity: For any word w *in* $C(G)$*, derive* $w = w$*.*

Symmetry: Given a relation $p = q$ *derivable from* R*, derive* $q = p$*.*

Transitivity: Given relations $p = q$ *and* $q = r$ *derivable from* R*, derive* $p = r$*.*

Right Multiplication of generators: Given a relation $p = q$ *derivable from* R*, and for any generator* g *in* G*, derive* $(p * g)' = (q * g)'$*.*

Proof. First assume $v = w$ is derivable over D_r but v and w do not label the same vertex. If $v = w$ is derivable, then there is a series of steps resulting in $v = w$, each of which results in a derived relation. Let the relation $x = y$ be the first relation in these steps where x and y don't label the same vertex. Now there will be a contradiction no matter what kind of step derived $x = y$:

If $x = y$ was a relation in R, then x and y label the same vertex, as dictated by the construction of Π, a contradiction.

Reflexivity: The relation is in the form $x = x$. But x must label the same vertex as x, a contradiction.

Symmetry: The relation $x = y$ is derived from $y = x$. But y and x label the same vertex, as this relation was the first relation which doesn't label the same vertex, a contradiction.

Transitivity: The relation $x = y$ is derived from $x = z$ and $z = y$. But then x labels the same vertex as z and z labels the same vertex as y, so x labels the same vertex as y, a contradiction.

Right multiplication of generators: The relation is in the form $p * g = q * g$ for some g in G, derived from some relation $p = q$. But by assumption p labels the same vertex as q, and by the construction of Π $p * g$ must label the same vertex as $q * g$, a contradiction.

For the other direction, assume that v and w label the same vertex. Then at some point in the construction of Π, two vertices labeled v and w were merged. In this case, either it was a relation of R, in which case it is certainly derivable, or $v = p * g$ and $w = q * g$ for some two vertices labeled p and q that were merged. Inducting over the length of the word, $p = q$ was derivable, and so $p * g = q * g$ is also derivable. □

Unfortunately, this means that the Cayley graph doesn't reflect all of the rules that can be used to derive new relations. To reflect left multiplication, we need to trace additional relations. These are the secondary relations $S(R)$:

Proposition 23. *Let $x = y$ be in $D(R)$ for some set of relations R over $L(G)$ for some set of generators G. Then $x' = y'$ is in $D_r(R \cup S(R))$ over $C(G)$.*

Proof. Since $x = y$ is in $D(R)$, there is some derivation $\Delta(D)$ to $x = y$. For each relation in $\Delta(D)$, replace it with its canonicalized form. Let D' be D with canonicalization. That is, D' is the following set of rules:

Reflexivity: For any word w in $L(G)$, derive $w' = w'$.

Symmetry: Given a relation $p = q$ derivable from R, derive $q' = p'$.

Transitivity: Given relations $p = q$ and $q = r$ derivable from R, derive $p' = r'$.

Right Multiplication: Given a relation $p = q$ derivable from R, and for any word w in $L(G)$, derive $(p * w)' = (q * w)'$.

Left Multiplication: Given a relation $p = q$ derivable from R, and for any word w in $L(G)$, derive $(w * p)' = (w * q)'$.

This new sequence will be a valid sequence in $\Delta(D')$, so $x' = y'$ is in $D'(R)$. We now need to recast this derivation to be a derivation $\Delta(D_r)$. However, as it stands, it may contain right multiplications by an arbitrary word w in $C(G)$: $p = q$ implies $(p * w)' = (q * w)'$. But $(p * w)' = pg_1g_2 \dots g_i$ and similarly $(q * w)' = qg_1g_2 \dots g_i$, so this derivation may be expanded to a series of right multiplications by generators, one of the rules of D_r.

It also may contain left multiplications by an arbitrary word w in $C(G)$. We show that when there are no left multiplications before a given one that derives the nth relation of the derivation, the first n relations can be recast to a sequence of relations that do not use left multiplications but instead may use relations from $S(R)$. This suffices, as for any subsequent left multiplications this process can then be repeated until there aren't any left multiplications left in the recast sequence.

So assume that there are no left multiplications before the given one that derives the nth relation, $x * p = x * q$, of the derivation, where $p = q$ was the relation used to derive this one. This will be shown by induction on the number of relations in the derivation. In other words, either the left multiplication will be moved earlier in the derivation and we repeat, or it will be removed. In general, the way we do this is through the involutory quandle axiom III: We recast relations derived from left multiplications by first using a secondary relation and then deriving what we would have derived from the left multiplications by right multiplications by generators and then taking advantage of the involutory quandle axiom III.

The base case is when $n = 1$. The claim is vacuously true, as any left multiplication requires using a previous relation in the derivation, which is a contradiction when there is only one relation. Now suppose the claim is true for some $n - 1$. It now suffices to show that we can recast this sequence so the left multiplication occurs no later than deriving the $n - 1$th step. We do this with a case-by-case analysis on the $n - 1$th step, which is without loss of generality one of the relations used to derive the nth relation:

If the $n-1$th relation is $p = q$ in R, then the nth relation can remain $x * p = x * q$, as it is a relation in $S(R)$.

If the $n-1$th relation is $y * p = y * q$ in $S(R)$, then the following sequence of relations may be substituted:

$(n-1)$ $x * p = x * q$ (relation in $S(R)$)

(n) $x * p * y = x * q * y$ (right multiplication)

$(n+1)$ $x * p * y * p = x * q * y * p$ (right multiplication)

$(n+2)$ $(x * q * y) * p = (x * q * y) * q$ (relation in $S(R)$)

$(n+3)$ $x * p * y * p = x * q * y * q$ (transitivity)

This last relation is the desired one, $x * (y * p) = x * (y * q)$, in canonicalized form.

If the $n-1$th relation is derived from the symmetry rule, i.e. this relation is $p = q$, was derived from some earlier relation $q = p$, and the nth relation is $x * p = x * q$. Then we can substitute the following relations:

$(n-1)$ $x * q = x * p$ (left multiplication)

(n) $p = q$ (symmetry)

$(n+1)$ $x * p = x * q$ (symmetry)

Now apply the induction hypothesis to recast the first $n-1$ relations.

If the $n-1$th relation is derived from transitivity, i.e. the $n-1$th relation is $p = q$ and was derived from earlier relations $p = r$ and $r = q$. Now we need to create two different derivations which has been changed only in the $n-1$th step. The new $n-1$th steps are $x * p = x * r$ and $x * r = x * q$. We can apply the induction hypothesis to both of these new derivations to recast them, and then derive $x * p = x * q$ from those derivations, as desired.

If the $n-1$th relation is derived from right multiplication of a generator, i.e. the $n-1$th relation is $p * g = q * g$ and was derived from an earlier relation $p = q$. Then we can substitute the following relations:

$(n-1)$ $(x * g) * p = (x * g) * q$ (left multiplication)

(n) $p * g = q * g$ (right multiplication)

$(n+1)$ $x * g * p * g = x * g * q * g$ (right multiplication)

This last relation is the desired one, $x * (p * g) = x * (q * g)$, in canonicalized form. Now apply the induction hypothesis to recast the first $n-1$ relations.

Now note that we may assume reflexivity is not used, as the only relations that can be derived from it is the relation $w = w$, which is a derivation already in the desired form.

This is all the possible cases, and so we have successfully moved up left multiplications until they could be replaced with relations from $S(R)$. Hence this new derivation to $x' = y'$ is a derivation $\Delta(D_r)$ using the relations $R \cup S(R)$. Thus $x' = y'$ is in $D_r(R \cup S(R))$. □

Now we can prove that two words are equal if and only if they can be derived in the manner appropriate for the Cayley graph using only the sets of relations R, $S(R)$, and I_G.

Corollary 24. *$p = q$ is in $D(R \cup A_G)$ if and only if $p' = q'$ is in $D_r(R \cup S(R) \cup I_G)$.*

Proof. Assume $p = q$ is in $D(R \cup A_G)$. By Proposition 23, $p' = q'$ is in $D_r(R \cup A_G \cup S(R \cup A_G))$. However in D_r derivations all relations are canonicalized, in which case relations from A_G correlating to axioms II and III are unnecessary, as they are trivialized when canonicalized to $x = x$, which is already a rule in D_r. Then $D_r(R \cup A_G \cup S(R \cup A_G)) = D_r(R \cup S(R) \cup I_G)$.

For the other direction, since there is some derivation $\Delta(D_r)$ to $p' = q'$, $\Delta(D_r)$ can be recast by removing all canonicalization and by changing all instances of axioms from $S(R)$ to uses of the left multiplication rule. Also, $I_G \subset A_G$, so this is now a derivation $\Delta(D)$ to $p = q$ using only the relations $R \cup A_G$. Hence $p = q$ is in $D(R \cup A_G)$, as desired. □

Since R, $S(R)$, and I_G are the relations traced in the algorithm and $D(R \cup A_G)$ are the relations that can be derived from the presentation, this shows that any relations that can be derived from the presentation is reflected in the Cayley graph.

Now we show that there is one and only one vertex for every distinct element in the quandle.

Proposition 25. *Let Γ be the graph constructed as by the above algorithm for a given quandle presentation $Q = \langle G|R\rangle$. For every x in $C(G)$, x labels some vertex in Γ.*

That is, all of the necessary vertices are in the graph.

Proof. Let $x = g_1 g_2 \dots g_n$ in $C(G)$. If $x = z$ is a relation in $R \cup I_G$ for some z in $C(G)$, then by the definition of the algorithm x would have been traced, and x indeed labels some vertex. Otherwise, we prove using induction over n that $x * p = x * q$ for $p = q$ in R gets traced. g_1 is a generator and therefore $g_1 * p = g_1 * q$ is traced. Now assume that $g_1 g_2 \dots g_i * p = g_1 g_2 \dots g_i * q$ is traced. Then $g_1 g_2 \dots g_i$ is a vertex in Γ. Thus either $g_1 g_2 \dots g_i g_{i+1}$ is already a vertex in Γ, or it isn't, but then g_{i+1} is a generator and $g_1 g_2 \dots g_i$ is a vertex, so $g_1 g_2 \dots g_i g_{i+1} * p = g_1 g_2 \dots g_i g_{i+1} * q$ subsequently gets traced. Since by definition of the algorithm, $|R| > 0$, $g_1 g_2 \dots g_i g_n$ labels some vertex. □

Finally, we prove that there is only one vertex for every unique element of the quandle.

Corollary 26. *Let Γ be the graph constructed as by the above algorithm for a given quandle presentation $Q = \langle G|R\rangle$. Then p and q are the same element of the quandle if and only if p' and q' label the same vertex of Γ.*

Proof. By definition, p and q are the same element of the quandle if and only if $p = q$ is a relation in $D(R \cup A_G)$. By Corollary 24, $p = q$ is a relation in $D(R \cup A_G)$ if and only if $p' = q'$ is in $D_r(R \cup S(R) \cup I_G)$. And by Proposition 22, $p' = q'$ is in $D_r(R \cup T \cup I_G)$ if and only if p' and q' label the same vertex of Γ, where T is the subset of $S(R)$ traced in the algorithm. Recall T is composed of one secondary relation for each vertex in Γ. It remains to show that $p' = q'$ is in $D_r(R \cup S(R) \cup I_G)$ if and only if $p' = q'$ is in $D_r(R \cup T \cup I_G)$. First of all the backwards direction follows from T being a subset of $S(R)$. Now let $\Delta(D_r)$ be a derivation of the relation $p' = q'$ in $D_r(R \cup S(R) \cup I_G)$. For each secondary relation $x * a = x * b$ in $\Delta(D_r)$, let $y * a = y * b$ be the relation in T such that x and y label the same vertex in Γ. From the definition of the algorithm, there is at least one relation in T $z * a = z * b$ such that z labels a given vertex. Also, by Proposition 25, the vertex at z exists, so this relation exists. x and y label the same vertex in Γ, so $x = y$ is in $D_r(R \cup T \cup I_G)$. Let $\delta(D_r)$ be the derivation of $x = y$. Then we may replace the secondary relation $x * a = x * b$ with:

1. $x = y$ $(\delta(D_r))$

2. $x * a = y * a$ (sequence of right multiplications by generators)

3. $x * b = y * b$ (sequence of right multiplications by generators)

4. $y * a = y * b$ (relation in T)

5. $x * a = x * b$ (transitivity)

This derivation now uses only relations in $D_r(R \cup T \cup I_G)$, so $p' = q'$ is in $D_r(R \cup T \cup I_G)$, as desired. □

We now put together the above proofs to show that the construction for the Cayley graph is correct:

Theorem 27. *Let Γ be the graph constructed as by the above algorithm for a given quandle presentation $Q = \langle G|R\rangle$. Then Γ is the Cayley graph for Q.*

We must show that the above algorithm satisfies Definition 16.

Proof. First we need to show that if $x * g = y$ for x, y in Q and g in G, there is a directed edge from x to y labeled g. But if $x * g = y$ in Q, then by Corollary 26, $x * g$ and y label the same vertex, i.e. there is an edge from x to y labeled g.

Now we must show that for every q in Q, there is a vertex in Γ labeled q. Write q in canonicalized form. Then this follows directly from Proposition 25.

Finally, we must show uniqueness of the vertices, i.e. if $p = q$ in Q then p and q label the same vertex. This is implied by Corollary 26, so Γ is indeed the Cayley graph for Q. □

We now show that not only is the algorithm correct, but that it works quickly enough to be useful. Namely, we want the algorithm to take finite time when the quandle is finite.

Theorem 28. *Let $Q = \langle G|R\rangle$ where $|G|$ and $|R|$ are finite. If $|Q|$ is finite, then Algorithm 20 takes finite time.*

Proof. It suffices to show that $|R \cup T \cup I_G|$ is finite. Since $|G|$ and $|R|$ are finite, $|R \cup I_G|$ is finite. Now in the tracing of T, in each iteration, $|R|$ times the number of new vertices in the graph are traced. The number of new vertices is bounded by the total number of vertices at the end of the algorithm, which must be finite if Q is finite. Then $|T|$ is finite, as desired. □

3 Using Cayley Graphs

Using Cayley graphs allows us to simplify argumentation about quandles quite a bit. In order to show the desirability of this simplification, we show how to bound the size of a quandle with and without quandles. The quandles we will do this for are quandles that come from Coxeter groups. See [1] for more on Coxeter groups. We introduce a method to create quandle presentations from Coxeter groups:

Definition 29. For a Coxeter group H given by presentation $\langle G|R\rangle$, we define $Q(G, R)$ to be the quandle given by the presentation $\langle G|R'\rangle$, where

$$R' = \{(r_1 * r_2) * \ldots * r_{n-1} = r_n | r_1 r_2 \ldots r_n \in R\}.$$

Since for all Coxeter groups, for every generator r, $r^2 = e$, it is safe to assume that there are no inverses in any word in R. Furthermore, in this translation, the associated relation in the quandle to the relation $r^2 = e$ is $r = r$, a trivial relation. Then there is no need to include these trivial relations in the set of relations for the quandle presentation.

Similarly,

Definition 30. For a group H given by presentation $\langle G|R\rangle$, we define $\mathrm{IQ}(G, R)$ to be the involutory quandle given by the presentation $\langle G|R'\rangle$, where

$$R' = \{(r_1 * r_2) * \ldots * r_{n-1} = r_n | r_1 r_2 \ldots r_n \in R\}.$$

These may not be well-defined operations for a given group. That is, two different presentations for the same group may yield different quandles. However, when it is clear which presentation we are using, we employ the notation $\mathrm{IQ}(Q)$ for $\mathrm{IQ}(G, R)$.

We will study the quandles $\mathrm{IQ}(I_k)$, where I_k is the Coxeter group given by the Coxeter matrix $\left(\begin{smallmatrix} 1 & k \\ k & 1 \end{smallmatrix}\right)$. In other words, $I_k = \langle a, b | a^2, b^2, (ab)^k, (ba)^k\rangle$. Then

$$\mathrm{IQ}(I_k) = \langle a, b | \underbrace{abab\ldots a}_{2k-1} = b, \underbrace{baba\ldots b}_{2k-1} = a\rangle.$$

First, we show the following bound on the size of $\mathrm{IQ}(I_k)$ without the use of Cayley graphs.

Lemma 31. $|\mathrm{IQ}(I_k)| \leq 2k - 1$.

Here we use an inductive argument instead.

Proof. Let W be

$$\{a, ab, aba, \ldots, \underbrace{aba\ldots}_{k}, b, ba, bab, \ldots, \underbrace{baba\ldots}_{k}\}.$$

(Note not all of these words are necessarily unique.)

We will show that $\mathrm{IQ}(I_k) = W$. Specifically, we will show that every word in $\mathrm{IQ}(I_k)$, in normal form, has length less than k. Recall that the normal form of a word is when the word has been left-associated, and instances of axioms I and II have been used on it to shorten it as much as possible (see Definition 6). Let w be a word in $\mathrm{IQ}(I_k)$ be written in normal form. Then $w = bab\ldots$ or $w = aba\ldots$. We will induct on $|w|$. When $|w| < k$, since w is in normal form, w must be in W, as desired. Else $|w| \geq k$. Without loss of generality, $w = aba\ldots$. Since $|w| \geq k$, we can use the relation $\underbrace{abab\ldots a}_{2k-1} = b$. Substituting, we get $w = (b)bab\ldots = bab\ldots$. Repeat until the length of w is no more than k, in which case it must be in W, proving the claim.

$|W| = 2k$. Furthermore, $\underbrace{abab\ldots a}_{2k-1} = b$ so $\underbrace{aba\ldots b}_{k} = \underbrace{baba\ldots b}_{k}$. Then

$$|\mathrm{IQ}(I_k)| \leq |W| - 1 = 2k - 1$$

. □

However, a Cayley graph for $\mathrm{IQ}(I_k)$ with $2k-1$ vertices will demonstrate the same thing, without having to go through an inductive argument.

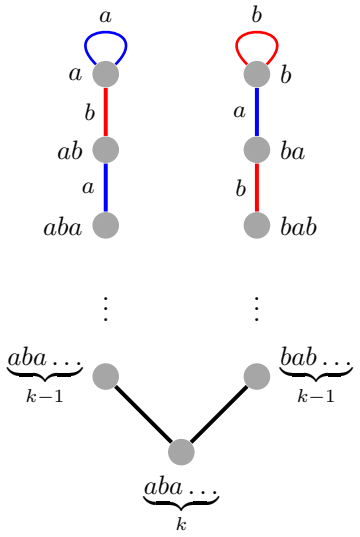

Figure 10: Involutory Cayley Graph for IQ(I_k)

This graph, shown in Figure 10, will provide enough information to determine exactly what IQ(I_k) is. But first, we need to be able to translate the graph into a quandle. The first step is to define a homomorphism between quandles that needs only to be defined on the generators that can be then extended to the entire quandle:

Proposition 32 (Winker)**.** *Let $\langle G|R\rangle$ be a presentation of a quandle Q. Let $h : G \to Q'$ be a map from the generators to a quandle Q'. Let $h' : Q \to Q'$ be an extension of h defined by $h'(t(g_1, g_2, \ldots, g_n)) = t(h(g_1), h(g_2), \ldots, h(g_n))$ where $t(g_1, g_2, \ldots, g_n)$ is a term over the generators in G (i.e. any element of Q). h' is a well-defined homomorphism if h' preserves every relation of Q. That is, if $w_1 = w_2$ is a relation in R, then $h'(w_1) = h'(w_2)$.*

With this homomorphism h', we can now give conditions for when it is an isomorphism:

Corollary 33. *If such a well-defined $h' : Q \to Q'$ exists, h' is onto, and $|Q| \leq |Q'|$, then $Q \cong Q'$.*

If such an $h' : Q \to Q'$ exists, then $|Q| \geq |h(Q)|$. But if h' is onto, then $|Q'| = |h(Q)|$, in which case $|Q'| \leq |Q| \leq |Q'|$, i.e. $|Q'| = |Q|$. Thus h' is also injective, and hence is an isomorphism.

Finally, this allows us to translate the graph into a quandle by relating the two through h':

Corollary 34. *Let $\Gamma = (V, E)$ be the Cayley graph associated with a quandle Q. Let Q' be the quandle $(V, *, /)$ where $*$ is defined in the following manner: $x * y$ is the element of V given by tracing $x * y$ written in normal form. That is $x * y = (\ldots((g_1 * g_2) * g_3) * \ldots g_{k-1}) * g_k$ for some $g_1, g_2, \ldots, g_k$ in G. Starting with the vertex g_1, there exists a path given by taking edges g_2, g_3, etc. in succession. $x * y$ is then defined as the vertex at the end of this path.*

Then $Q \cong Q'$.

In other words, the Cayley graph of the quandle represents the quandle in an intuitive fashion. To prove it, we merely need to check that the conditions from Corollary 33 hold true.

Proof. First, it is straightforward to check that Q' is well-defined.

Let $h : G \to Q'$ be given by $h(g) = g'$, where g' is the vertex labeled g. This is a well-defined map by Definition 16. Let h' be given as in Proposition 32. h' preserves relations, as for any relation $x = y$ in R, $h'(x)$ and $h'(y)$ are vertices in Γ and must be the same vertex by Definition 16. Then by Proposition 32, h' is a well-defined homomorphism. The bijection between elements of Q and vertices also both implies h' is onto and the number of vertices is equal to the size of the quandle. Then $|Q| = |Q'|$. Thus we have satisfied the conditions for Corollary 33, as desired. □

Corollary 35. $|\operatorname{IQ}(I_k)| = 2k - 1$.

Instead of having to use an inductive argument, we can show that the size of $\operatorname{IQ}(I_k)$ is $2k - 1$ much more easily using a Cayley graph. To show that Figure 10 is the Cayley graph for $\operatorname{IQ}(I_k)$, it suffices to give the relations needed to trace the graph for a given k.

The primary relations are $\underbrace{abab\dots a}_{2k-1} = b$ and $\underbrace{baba\dots b}_{2k-1} = a$. The idempotence relations are $a * a = a$ and $b * b = b$. The secondary relations are:

$$b(\underbrace{abab\dots a}_{2k-1}) = b(b)$$
$$b(\underbrace{abab\dots a}_{2k-1}) = b(a)$$
$$\underbrace{aba\dots}_{\ell}(\underbrace{abab\dots a}_{2k-1}) = \underbrace{aba\dots}_{\ell}(b)$$
$$\underbrace{aba\dots}_{\ell}(\underbrace{abab\dots a}_{2k-1}) = \underbrace{aba\dots}_{\ell}(a)$$

for $1 \leq \ell < 2k - 1$. Tracing these relations result in Figure 10, which has $2k - 1$ vertices. Thus $|\operatorname{IQ}(I_k)| = 2k - 1$. Furthermore, the Cayley graph gives a way to find out what any word in the quandle is: convert the word to normal form and then trace it through the graph.

As an aside, it turns out that $\operatorname{IQ}(I_k)$ are a really nice class of quandles, because they are Latin.

Definition 36. A Latin quandle is a quandle Q where for all x, y, z in Q, if $x * y = x * z$, then $y = z$.

The name is due to the fact that the Cayley table of a finite Latin quandle is also a Latin square, which means there are no repeats in any row or column.

Proposition 37. $\operatorname{IQ}(I_k) = \langle a, b | \underbrace{abab\dots a}_{2k-1} = b, \underbrace{baba\dots b}_{2k-1} = a \rangle$ *is Latin.*

Proof. First, we show that for any finite involutory quandle Q, if for some x_0 in Q we have $x_0 * y = x_0 * z$ implies $y = z$ for all y, z in Q, then Q is Latin.

So suppose there exists an x_0 in Q such that we have $x_0 * y = x_0 * z$ implies $y = z$ for all y, z in Q. Now let x in Q and suppose $x * y = x * z$. Since Q is finite, there exists a q in Q such that $x_0 * q = x$. Then $(x_0 * q) * y = (x_0 * q) * z$. Multiplying on the right, $((x_0 * q) * y) * q = ((x_0 * q) * z) * q$.

This is equivalent to $x_0 * (y * q) = x_0 * (z * q)$. From our assumption about x_0, $y * q = z * q$. Hence $y = z$, so Q is Latin.

Then it suffices to show that for all y, z in $\mathrm{IQ}(I_k)$, if $a * y = a * z$, then $y = z$. We do this by showing that for some y in Q, $a * y = q$ for all q in $\mathrm{IQ}(I_k)$. Since $\mathrm{IQ}(I_k)$ is finite, this implies that $a * y$ is unique for each y, which implies what we need to show.

Let y in Q. We know that $\mathrm{IQ}(I_k) = \{a, ab, aba, \ldots, \underbrace{aba\ldots}_{k}, b, ba, bab, \ldots, \underbrace{bab\ldots}_{k}\}$, so $y = \underbrace{aba\ldots}_{\ell}$ for some ℓ such that $1 \leq \ell \leq k$ or $y = \underbrace{bab\ldots}_{\ell}$ for some ℓ such that $1 \leq \ell < k$. First, suppose that $y = \underbrace{aba\ldots}_{\ell}$. If ℓ is odd, then

$$a * y = a * (\underbrace{aba\ldots a}_{\ell}) = a\,\underbrace{aba\ldots a}_{2\ell-1} = \underbrace{aba\ldots a}_{2\ell-1}.$$

If ℓ is even, then

$$a * y = a * (\underbrace{aba\ldots b}_{\ell}) = a\,\underbrace{bab\ldots b}_{2\ell-1} = \underbrace{aba\ldots b}_{2\ell}.$$

Now suppose that $y = \underbrace{bab\ldots}_{\ell}$. If ℓ is odd, then

$$a * y = a * (\underbrace{bab\ldots a}_{\ell}) = a\,\underbrace{bab\ldots b}_{2\ell-1} = \underbrace{aba\ldots b}_{2\ell}.$$

If ℓ is even, then

$$a * y = a * (\underbrace{bab\ldots a}_{\ell}) = a\,\underbrace{aba\ldots a}_{2\ell-1} = \underbrace{aba\ldots a}_{2\ell-1}.$$

Thus there exists a y in Q such that $a * y$ is in $\{a, ab, aba, \ldots, \underbrace{aba\ldots a}_{2k-1}\}$. But this set is $\mathrm{IQ}(I_k)$. So $a * y = q$ for all q in $\mathrm{IQ}(I_k)$, as desired.

□

3.1 One-Relator Quandles

Besides quandles created from Coxeter groups, we are also interested in one-relator quandles, which are quandles with a single primary relation. It turns out there is a useful intersection:

Theorem 38. *One-relator involutory quandles on two generators can be classified into* $\mathrm{IQ}(I_k)$ *and* J_k*, where* J_k *is defined as* $J_k = \langle a, b | \underbrace{aba\ldots b}_{2k} = a\rangle$*. Furthermore, all one-relator involutory quandles on two generators are finite.*

Proof. Given a relation $x = y$ over two generators a and b, we can assume without loss of generality that x and y are in normal form and that x starts with a. x and y must be of the form $abab\ldots$ and $baba\ldots$. Then we can move all but one generator over from y to the left-hand side, resulting in four possible relations: $aba\ldots b = b$, $aba\ldots a = a$, $aba\ldots a = b$, or $aba\ldots b = a$. For those first two possible relations, we then multiply both sides on the right by b or a respectively, giving

two possibilities for this relation: $aba\ldots b = a$ for an even number of generators on the left-hand side, or $aba\ldots a = b$ for an odd number of generators on the left-hand side. This yields $\mathrm{IQ}(I_k) = \langle a, b | \underbrace{aba\ldots a}_{2k-1} = b\rangle$, and $J_k = \langle a, b | \underbrace{aba\ldots b}_{2k} = a\rangle$.

It remains to give the Cayley graph for J_k. The Cayley graph is shown in Figure 11.

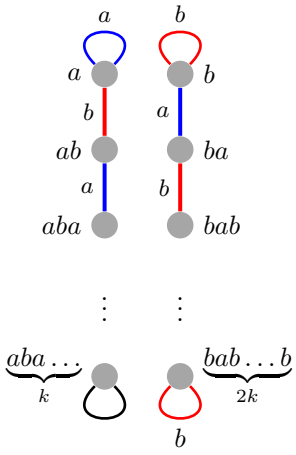

Figure 11: Involutory Cayley Graph for J_k

The relations necessary to trace this graph are the following: The primary relation is $\underbrace{aba\ldots b}_{2k} = a$, the idempotence relations are $a * a = a$ and $b * b = b$. The secondary relations are:

$$\underbrace{aba\ldots}_{\ell}(\underbrace{aba\ldots b}_{2k}) = \underbrace{aba\ldots}_{\ell}(a)$$
$$\underbrace{bab\ldots b}_{2\ell}(\underbrace{aba\ldots b}_{2k}) = \underbrace{bab\ldots b}_{2\ell}(a)$$

for $1 \leq \ell \leq k$. Tracing these relations results in Figure 11, so $|J_k| = 3k$. Hence all one-relator involutory quandles on two generators are finite.

□

Now that we know how Cayley graphs can be used to explore quandles like the one-relator involutory quandles, we return to the word problem:

Proposition 39. *Let $Q = \langle G|R\rangle$. If any finite portion of the Cayley graph for Q is constructible in finite time, then Q has a solvable word problem.*

Proof. Let $Q =< G|R >$ be given. Let w_1, w_2 in $L(G)$. Using Algorithm 20, construct a finite portion of Q that labels both w_1 and w_2. From Corollary 26, $w_1 = w_2$ if and only if w_1 and w_2 label the same vertex, solving the word problem. □

Since our algorithm constructs the entire Cayley graph in finite time for finite quandles, it immediately follows that finite quandles have a solvable word problem. This is a very natural result that should be expected to follow from this algorithm.

So far, all of the Cayley graphs we've seen have been finite. If the Cayley graph is infinite, however, the algorithm above to construct the graph will take infinite time. However, the algorithm nicely allows an alternative approach to construct the Cayley graph: We can induct over the number of iterations, merely showing that each iteration through the algorithm doesn't collapse vertices introduced in previous iterations. For example, the Cayley graph for $\langle a, b, c | a * b = c \rangle$ is shown in Figure 12.

In order to show that this is the Cayley graph, we assume as the inductive hypothesis that the partial Cayley graph after iteration i has the following vertices: $\{a, c, \underbrace{caba\dots a}_{2k}, \underbrace{caba\dots b}_{2k+1}\}$ in the component containing c and $\{b, \underbrace{baba\dots a}_{2k}, \underbrace{baba\dots b}_{2k+1}\}$ in the component containing b for each $1 \leq k \leq i$. And the only vertices created in step i were the last four, i.e. $\underbrace{caba\dots a}_{2i}, \underbrace{caba\dots b}_{2i+1}, \underbrace{baba\dots a}_{2i}$, and $\underbrace{baba\dots b}_{2i+1}$.

Then in iteration $i+1$, the following relations are traced:

$$\begin{aligned}
\underbrace{caba\dots a}_{2i}(ab) &= \underbrace{caba\dots a}_{2i}(c) \\
\underbrace{caba\dots b}_{2i+1}(ab) &= \underbrace{caba\dots b}_{2i+1}(c) \\
\underbrace{baba\dots a}_{2i}(ab) &= \underbrace{baba\dots a}_{2i}(c) \\
\underbrace{baba\dots b}_{2i+1}(ab) &= \underbrace{baba\dots b}_{2i+1}(c)
\end{aligned}$$

Normalizing, these relations are:

$$\begin{aligned}
\underbrace{caba\dots bab}_{2i+3} &= \underbrace{caba\dots ac}_{2i+1} \\
\underbrace{caba\dots a}_{2i-2} &= \underbrace{caba\dots bc}_{2i+2} \\
\underbrace{baba\dots bab}_{2i+3} &= \underbrace{baba\dots ac}_{2i+1} \\
\underbrace{baba\dots a}_{2i-2} &= \underbrace{baba\dots bc}_{2i+2}
\end{aligned}$$

Tracing these relations, then, introduces four new vertices: $\underbrace{caba\dots a}_{2i+2}, \underbrace{caba\dots b}_{2i+3}, \underbrace{baba\dots a}_{2i+2}$, and $\underbrace{baba\dots b}_{2i+3}$, as desired. It is easy to check that tracing these relations yields the structure of the graph seen in Figure 12. Thus the induction is complete.

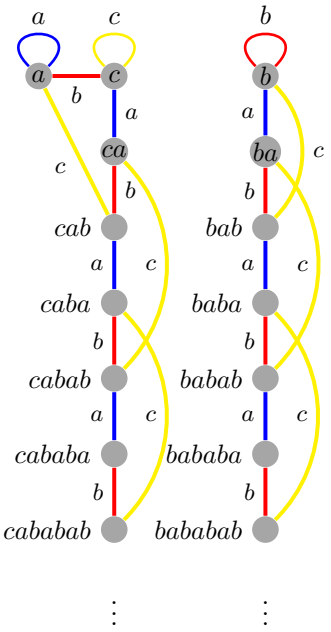

Figure 12: Involutory Cayley Graph for $\langle a, b, c | a * b = c \rangle$

4 Conclusion

In this paper, we have given an improved algorithm for solving the word problem in quandles. This allowed us to solve the word problem for all finite quandles, including one-relator involutory quandles on two generators. It also gives us a method for solving the word problem for non-finite quandles by taking advantage of the structure of the algorithm. This has given us a strong foothold into solving the word problem for many quandles.

There are many fronts for continuing work. First, there is work to be done on the algorithm itself. We have given a naive construction of the algorithm, but a clever one with a faster asymptotic runtime will establish good bounds on how long it takes to solve the word problem when it is indeed solvable.

Second, it is quite possible to prove that many quandles have a solvable word problem using this algorithm. To find out which quandles these are, we have written an implementation of the algorithm to speed up this process. One-relator quandles continue to be good candidates, as they seem to all have a solvable word problem so far. In addition, the algorithm takes time proportional to the number of relations, so these are faster to run. Other quandles that may prove to be interesting are quandles whose presentations come from presentations of groups. We give one way of translating presentations, but there needs to be more work on finding a well-defined translation that preserves the property that the word problem is solvable.

Finally, the existence and usefulness of Cayley graphs for quandles suggests that Cayley graphs may be possible for many types of algebras. Extending a notion of Cayley graphs to other algebras may then lead to insight on the word problem for other algebras.

While there is much more work to be done on the topic, this paper has made progress on solving

the word problem in quandles.

References

[1] M. Davis. *The Geometry And Topology of Coxeter Groups*. London Mathematical Society monographs: New Series. Princeton Univeristy Press, 2008.

[2] Trevor Evans. The word problem for abstract algebras. *Journal of the London Mathematical Society*, s1-26(1):64–71, 1951.

[3] Roger Fenn and Colin Rourke. Racks and links in codimension two. *Journal of Knot Theory and Its Ramifications*, 01(04):343–406, 1992.

[4] F. A. Garside. The braid group and other groups. *The Quarterly Journal of Mathematics*, 20(1):235–254, 1969.

[5] W. Magnus, A. Karrass, and D. Solitar. *Combinatorial Group Theory: Presentations Of Groups In Terms Of Generators And Relations*. Dover Books on Mathematics Series. Dover Publications, Incorporated, 2004.

[6] V.O. Manturov. *Knot Theory*. Chapman & Hall, 2004.

[7] P.S. Novikov. On the algorithmic unsolvability of the word problem in group theory. *Trudy Mat. Inst. Steklov.*, 44:3–143, 1955.

[8] J. Stillwell. The word problem and the isomorphism problem for groups. *Bulletin (New Series) of the American Mathematical Society*, 6(1):33–56, 1982.

[9] J. Tits. Le probleme des mots dans les groupes de Coxeter. *Sympos. Math. Rome 1967/68*.

[10] Steven K. Winker. *Quandles knot invariants and the N-fold branched cover*. PhD thesis, University of Illinois at Chicago, 1984.