A COMBINATORIAL APPROACH TO STALLINGS' ALGORITHM

HIKARU KIYO

Abstract. We reconstruct the Stallings Algorithm for finding the rank of a subgroup of a free group combinatorially, and investigate the complexity of the algorithm on different subgroups of free groups.

Contents

1. Introduction

Given a finitely geenrated subgroup of a free group, it is a nontrivial question to ask what the rank of the subgroup is. The reason the inquiry is nontrivial, is because there may be relations among the generators that are not so obvious. Hence to find the rank of the subgroup, we are intereted in finding the free basis of the subgroup. However, it is often not a simple task to determine whether a redundant generator exists. Consider the example $H = \langle a^2b^{-1}, ba^{-1}ba, aba^{-1}, a^6\rangle$. It's unclear whether H has a redundant generator simply by looking at the generators. It happens to be true that $a^2b^{-1} \cdot ba^{-1}ba \cdot (aba^{-1})^{-1} = a^2$, and hence this implies that $H = \langle a^2b^{-1}, ba^{-1}ba, aba^{-1}, a^6\rangle = \langle a^2b^{-1}, ba^{-1}ba, aba^{-1}\rangle$. But again, there may be another redundant generator among $a^2b^{-1}, ba^{-1}ba, aba^{-1}$, which may require a much more complicated computation to obtain.

To answer this question, John Stallings came up with an algorithm that not only eliminates but simplifies the generators of a subgroup of a freegroup. The algorithm uses graphs, and a function on graphs, which we will later call as folding. Stallings uses a topological aproach, using fundamental groups as the primary tool to prove his result. In this paper we reprove the Stalling algorithm with a more combinatorial construction. Afterwards, we will briefly explore the complexity of the algorithm, and give examples of subgroups that behave differently with respect to the algorithm.

2. Labled Oriented Graphs

We begin by defining labled oriented graphs, which will be our primary object of investigation in this paper.

Date: DEADLINE AUGUST 22, 2008.

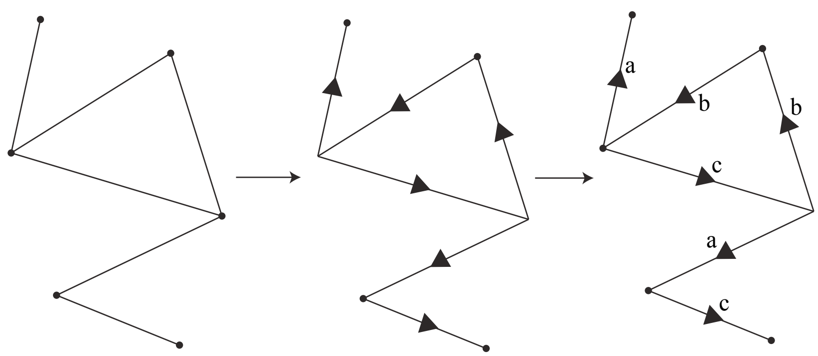

FIGURE 1. Constructing a labled oriented graph.

Definition 2.1. Let $A = \{a_1, \cdots, a_n\}$ be a finite set, and let $A^{-1} = \{a_1^{-1}, \cdots, a_n^{-1}$. We think of $A \cup A^{-1}$ as an *alphabet*, and each a_i, a_i^{-1} as *letters* of the alphabet. Using these letters, we can form *words*, which is just a finite string of letters and their inverses.

In case we have a subword of the form aa^{-1} or $a^{-1}a$, we can *freely reduce* the word by omitting the subword. For example, $abaa^{-1}b$ can freely reduced to ab^2. We say that a word is *freely reduced* if it doesnt contain any subwords of the form aa^{-1} or $a^{-1}a$ for any $a \in A$.

Example 2.2. Let $A = \{a, b, c\}$. Then

$$a^5b^{-2}a,\ c^4ab^{-3},\ \text{and } ab^7cabca$$

are all freely reduced words formed from the alphabet A.

Now we define graphs which we will later lable using alphabets.

Definition 2.3. A *graph* Γ consists of two sets E and V and two functions $f : E \to E$ and $\iota : E \to V$ such that f is bijective, $\overline{e} \neq e$, and $\overline{\overline{e}} = e$ where $\overline{e}$ is the image of e under f. Each $e \in E$ is a *directed edge* of Γ, and $\overline{e}$ is the *reverse* of e. V is the set of *vertices* of Γ. Finally, we define $\iota(e)$ as the *initial vertex* of e and $\tau(e) = \iota(\overline{e})$ as the *terminal vertex* of e.

Given a graph Δ, we'll denote its edge set by $\mathcal{E}(\Delta)$ and its vertex set by $\mathcal{V}(\Delta)$.

Definition 2.4. Let Γ be a graph. Then we can make an *oriented graph* $\hat{\Gamma}$ by choosing one edge for each pair $\{e, \overline{e}\}$.

Definition 2.5. Let A be an alphabet. Then an *A-labled oriented graph*, $\hat{\Gamma}$, is an oriented graph where each edge $e \in E$ is labled by a letter in A, which we denote by $l(a)$, with the condition that if $l(e) = a$, then $l(\overline{e}) = a^{-1}$.

Definition 2.6. Let $\hat{\Gamma}$ be a A-labled oriented graph. Then a *path* p in $\hat{\Gamma}$ is a sequence of edges $p = e_1, e_2, \cdots, e_k,\ e_i \in \mathcal{E}(\hat{\Gamma})$, such that $\tau(e_i) = \iota(e_{i+1})$ for each i. We define the *origin* of p as $\iota(p) = \iota(e_1)$, and the *end point* of p as

$\tau(p) = \tau(e_k)$. The *length* of p is k, and for each path there is a word that lables the path, $l(p) = l(e_1)l(e_2)\cdots l(e_k)$. Hence $l(p)$ is a word in the alphabet A. If $\iota(p) = \tau(p) = v$, then we say that p is a *loop based at* v .

A path from v_1 to v_2 is exactly what one thinks it is, its a sequence of connected edges that takes you from v_1 to v_2. Note that you can have many distinct paths from v_1 to v_2. To read the lable of the path, just read the lable of the sequence of edges, except when you go through an edge in the opposite direction than its orientation, you read the inverse of the lable.

Note that in our definition of a path, we can have a path that contains a subpath of the form $e\bar{e}$ or $\bar{e}e$ for any edge e. We will define paths that have no subpaths of such form to be *path reduced* (sometimes we will just say a *reduced* path). Furthermore, we will say that a path is *freely reduced* if the word it represents is freely reduced.

From now on, unless otherwise stated, we will assume that our graphs are *connected*, which implies that given any pair of vertices $\{v_1, v_2\}$, there exists a path p such that $\iota(p) = v_1$ and $\tau(p) = v_2$.

Definition 2.7. Let $\hat{\Gamma}$ be a A-labled oriented graph. Let v be a vertex of $\hat{\Gamma}$, and suppose these exists e_1, e_2, edges of $\hat{\Gamma}$ such that $\iota(e_1) = \iota(e_2)$ and $l(e_1) = l(e_2)$. Then we construct a new A-labled oriented graph, $\hat{\Delta}$, by identifying $\tau(e_1)$ with $\tau(e_2)$, and e_1 with e_2. More precisely,

(1) Let $\mathcal{V}(\hat{\Delta}) = (\mathcal{V}(\hat{\Delta})\backslash\{\tau(e_1), \tau(e_2)\}) \cup \{v_{12}\}$. We obtain v_{12} from identifying $\tau(e_1)$ and $\tau(e_2)$.
(2) Similarly, let $\mathcal{E}(\hat{\Delta}) = (\mathcal{E}(\hat{\Delta})\backslash\{e_1, e_2\}) \cup \{e_{12}\}$. e_{12} is added to replace e_1 and e_2 that have been identified.
(3) We let $\tau_{\hat{\Delta}}(e) = \tau_{\hat{\Gamma}}(e)$ for any $e \in \mathcal{E}(\hat{\Gamma})$ such that $e \neq e_1, e_2$, and $\tau_{\hat{\Delta}}(e_{12}) = v_{12}$. Similarly, $\iota_{\hat{\Delta}}(e) = \iota_{\hat{\Gamma}}(e)$ for any $e \in \mathcal{E}(\hat{\Gamma})$ such that $e \neq e_1, e_2$, and $\iota_{\hat{\Delta}}(e_{12}) = \iota_{\hat{\Gamma}}(e_1) = \iota_{\hat{\Gamma}}(e_2)$.
(4) The lables of $\hat{\Delta}$ are defined as $l_{\hat{\Delta}}(e) = l_{\hat{\Gamma}}(e)$ if $e \neq e_1, e_2$, and $l_{\hat{\Delta}}(e_{12}) = l_{\hat{\Gamma}}(e_1) = l_{\hat{\Gamma}}(e_2)$.

Thus we get a A-labled oriented graph $\hat{\Delta}$. We can construct an analogous construction in the case $\tau(e_1) = \tau(e_2)$ and $l(e_1) = l(e_2)$, by identifying the two end points and the two edges. We call this process *folding*, and we say that we obtained $\hat{\Delta}$ by folding $\hat{\Gamma}$ along e_1 and e_2.

The reader is encouraged to experiment some examples of folding. It can be somewhat complicated when there are many edges coming out of the vertices that are to be identified with each other. Also, its worth trying examples where folding results in creating a loop, as illustrated in figure 2.

Definition 2.8. Let $\hat{\Gamma}$ be a A-labled oriented graph. Then we say that $\hat{\Gamma}$ is *folded* if for every $v \in \mathcal{V}(\hat{\Gamma})$ and letter $a \in A$, there is at most one edge $e_1 \in \mathcal{E}(\hat{\Gamma})$ such that $l(e_1) = a$ and $\iota(e_1) = v$, and there is at most one edge $e_2 \in \mathcal{E}(\hat{\Gamma})$ such that $l(e_2) = a$ and $\tau(e_2) = v$.

In other words, when a graph is folded, we are no longer able to fold the graph. In figure 2, both examples are folded once the fold is made (in other words, the graphs on the left is unfolded, and the graphs on the right are folded). Note that folding a connected graph results in another connected graph. Also, if we have a

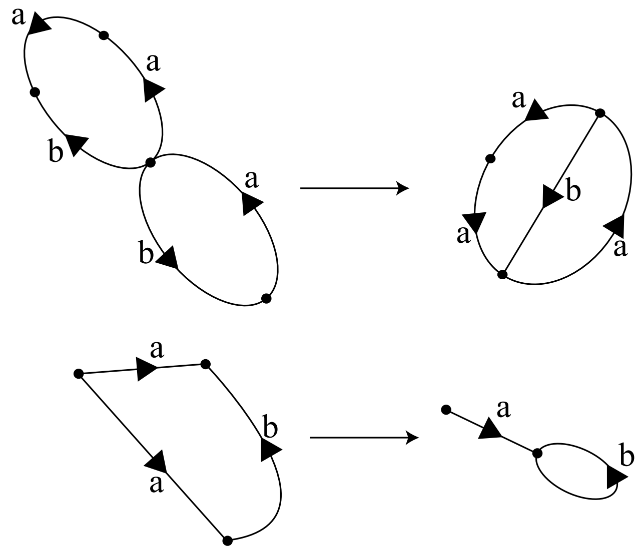

FIGURE 2. Two exaples of a fold. The bottom example creates a loop.

path p with origin v_0 and end point v_1, and the image of v_0, v_1 under folding is v_0', v_1' respectively, then the image of p under folding is a path p', with origin v_0' and end point v_1', where $l(p) = l(p')$. In other words, the lable of a path is preserved under folding.

Folding is essentially all we need to construct Stallings algorithm. The algorithm consists of folding a labled oriented graph until we get a folded graph. We'll come back to the actual construction later.

Definition 2.9. Let $\hat{\Gamma}$ be a A-labled oriented graph, and $v \in \mathcal{V}(\hat{\Gamma})$. Then we define the *dictionary of* $\hat{\Gamma}$ *based at* v to be the set

$$\{l(p) \mid p \text{ is a reduced loop based at } v \text{ in } \hat{\Gamma}\},$$

which we will denote as $\mathcal{D}_v(\hat{\Gamma})$.

It's clear that under the operation of concatenation and path-reduction, $\mathcal{D}_v(\hat{\Gamma})$ is a group. We have the trivial word since it's the lable of the trivial loop, and the inverse of a word in the dictionary can be obtained by considering the reverse loop. As for closure, we just concatenate and path reduce the two reduced loops.

Note that even though the dictionary is composed of reduced loops, there may be words that are not freely reduced in the dictionary. However, when one considers folded graphs, this is no longer the case.

Lemma 2.10. *If* $\hat{\Gamma}$ *is a folded A-labled oriented graph, then for any* $v \in \mathcal{V}(\hat{\Gamma})$, $\mathcal{D}_v(\hat{\Gamma})$ *only contains words that are freely reduced.*

Proof. Let v be an arbitrary vertex of $\hat{\Gamma}$. In order for some $w \in \mathcal{D}_v(\hat{\Gamma})$ to be not freely reduced, there must be at least 2 edges e_1 and e_2 such that $l(e_1) = l(e_2)$ and $\tau(e_1) = \tau(e_2) = v$ or $\iota(e_1) = \iota(e_2) = v$. But since $\hat{\Gamma}$ is folded, this is impossible. Hence our claim holds. $\square$

3. Free Groups and Labled Graphs

Definition 3.1. Let $A \cup A^{-1}$ be an alphabet. Then we define the *free group on* A as the collection of all freely reduced words in A, with the group operation defined as

$$w_1 \cdot w_2 = \underline{w_1 w_2},$$

where w_1 and w_2 are words in A and $\underline{w_1 w_2}$ is the freely reduced word we obtain from the concatenated word $w_1 w_2$.

Definition 3.2. Let $\hat{\Gamma}$ be a A-labled oriented graph, and $v \in \mathcal{V}(\hat{\Gamma})$. Then we define the *reduced dictionary at* v to be the set

$$\{\underline{w} \mid w \in \mathcal{D}_v(\hat{\Gamma})\},$$

which we will denote as $\underline{\mathcal{D}}_v(\hat{\Gamma})$.

Now we show that dictionaries of an A-labled oriented graph are subgroups of $F(A)$.

Lemma 3.3. *Let $\hat{\Gamma}$ be an A-labled oriented graph, and let $v \in \mathcal{V}(\hat{\Gamma})$. Then $\underline{\mathcal{D}}_v(\hat{\Gamma})$ is a subgroup of $F(A)$.*

Proof. Let $x_1, x_2 \in \underline{\mathcal{D}}_v(\hat{\Gamma})$. Then this implies that there exists two loops p_1, p_2 based at v such that $l(p_i) = w_i$ and $\underline{w_i} = x_i$ for $i = 1, 2$. Consider the concatenated loop $p_1 p_2$, and let p_3 be the reduced loop we obtain from $p_1 p_2$. Then since the lable of p_3 can be obtained by freely reducing $w_1 w_2$, we get $\underline{l(p)} = \underline{w_1 w_2} = w_1 \cdot w_2 \in F(A)$. Now, since p is a reduced loop based at v, $\underline{l(p)} \in \underline{\mathcal{D}}_v(\hat{\Gamma})$. So, $\underline{\mathcal{D}}_v(\hat{\Gamma})$ is closed under multiplication. Now, its clear that x_1^{-1} can be obtained from the reverse path p_1^{-1}, and hence we have $x_1^{-1} = \underline{l(p_1^{-1})}$. Finally, we also have $1 \in \underline{\mathcal{D}}_v(\hat{\Gamma})$. Hence $\underline{\mathcal{D}}_v(\hat{\Gamma})$ is a subgroup of $F(A)$. □

Note that if $\hat{\Gamma}$ is folded, by lemma 2.10 its dictionaries only contain freely reduced words, and hence $\mathcal{D}_v(\hat{\Gamma}) = \underline{\mathcal{D}}_v(\hat{\Gamma})$ is a subgroup of $F(A)$ for each $v \in \mathcal{V}(\hat{\Gamma})$.

Now we prove that the freely reduced dictionary is preserved under folding.

Lemma 3.4. *Let $\hat{\Gamma}_0$ be a A-labled oriented graph, and let $v_0 \in \mathcal{V}(\hat{\Gamma}_0)$. If $\hat{\Gamma}_1$ and v_1 are the images of $\hat{\Gamma}_0$ and v_0 respectively under a single folding, then we have*

$$\underline{\mathcal{D}}_{v_0}(\hat{\Gamma}_0) = \underline{\mathcal{D}}_{v_1}(\hat{\Gamma}_1).$$

Proof. Suppose $\hat{\Gamma}_1$ is obtained from $\hat{\Gamma}_0$ by identifying $e_1, e_2 \in \mathcal{E}(\hat{\Gamma}_0)$, where $\iota(e_1) = \iota(e_2) = x_0$, $x_0 \in \mathcal{V}(\hat{\Gamma}_0)$ and $l(e_1) = l(e_2) = a$, $a \in A$. Let e be the edge that replaces e_1, e_2, where $l(e) = a$, and let x_1 be the image of x_0 in $\hat{\Gamma}_1$. Let $w \in \underline{\mathcal{D}}_{v_0}(\hat{\Gamma}_0)$, so that there exists a path-reduced loop p_0 based at v_0 such that $\underline{l(p_0)} = w$. Suppose that p_0 has no sub-path of the form $\overline{e_1} e_2$ or $\overline{e_2} e_1$. Then if p_1 is the image of p_0, p_1 is a path-reduced loop based at v_1, where $l(p_1) = l(p_0) \Rightarrow \underline{l(p_1)} = \underline{l(p_0)}$. Hence we have $w = \underline{l(p_1)} \in \underline{\mathcal{D}}_{v_1}(\hat{\Gamma}_1)$. If p_0 contains sub-paths of the form $\overline{e_1} e_2$ or $\overline{e_2} e_1$, p_1 will not be path reduced. Let p_1' be the path-reduced loop we obtain from p_1. Then we have $\underline{l(p_1')} = \underline{l(p_1)} = \underline{l(p_0)}$, which also implies that $w \in \underline{\mathcal{D}}_{v_1}(\hat{\Gamma}_1)$. Hence we have $\underline{\mathcal{D}}_{v_0}(\hat{\Gamma}_0) \subseteq \underline{\mathcal{D}}_{v_1}(\hat{\Gamma}_1)$. Now we show the other inclusion. Let $w' \in \underline{\mathcal{D}}_{v_1}(\hat{\Gamma}_1)$. Then there exists q', a path-reduced loop based at v_1 in $\hat{\Gamma}_1$, such that $l(q') = w'$. Let q be the preimage of q'. If q' doesn't contain the edge e or $\overline{e}$, we clearly have $\underline{l(q)} = w'$, and hence $w' \in \underline{\mathcal{D}}_{v_0}(\hat{\Gamma}_0)$. Suppose q' contains the edge e or $\overline{e}$. Then we can partition q' in the following manner: $q' = q_1 y_1 q_2 y_2 \cdots q_n y_n q_{n+1}$ where $y_i = e$ or $\overline{e}$, and q_i is a reduced path that does not contain e or $\overline{e}$. Note that since q' is path-reduced, $q_i \neq 1$ for $i = 2, \cdots, n$. Now, since each q_i does not contain e or $\overline{e}$, their preimage in $\hat{\Gamma}_0$ is the same path. To construct a reduced path in $\hat{\Gamma}_0$ with lable w', for each i, we need to find an edge z_i that will make $q_i z_i q_{i+1}$ into a path in $\hat{\Gamma}_0$. Since by folding we've identified e_1 and e_2 into e, for each y_i, we just choose either e_1 or e_2 if $y_i = e$, and $\overline{e_1}$ or $\overline{e_2}$ if $y_i = \overline{e}$. Then we will have a reduced path $q_i z_i q_{i+1}$, which has the same lable as $q_i y_i q_{i+1}$, since $l(e) = l(e_1) = l(e_2)$. Hence by repeating this process for each i, we obtain a reduced path $q = q_1 z_1 q_2 z_2 \cdots q_n z_n q_{n+1}$ in $\hat{\Gamma}_0$, where $l(q) = l(q')$. Thus $l(q) \in \mathcal{D}_{v_0}(\hat{\Gamma}_0) \Rightarrow \underline{l(q)} \in \underline{\mathcal{D}}_{v_0}(\hat{\Gamma}_0)$. So, we have $\underline{\mathcal{D}}_{v_1}(\hat{\Gamma}_1) \subseteq \underline{\mathcal{D}}_{v_0}(\hat{\Gamma}_0)$, and hence we have $\underline{\mathcal{D}}_{v_1}(\hat{\Gamma}_1) = \underline{\mathcal{D}}_{v_0}(\hat{\Gamma}_0)$. □

Now that we've shown that the freely reduced dictionary is preserved under folding, we go on to prove that for any subgroup of $F(A)$, there exists a labled oriented graph whose dictionary at a certain vertex is precisely the given supgroup.

Proposition 3.5. *Let $H = \langle h_1, \cdots, h_k \rangle$ be a finitely generated subgroup of $F(A)$. Then there exists a connected, folded A-labled oriented graph $\hat{\Gamma}$ and a vertex $v \in \mathcal{V}(\hat{\Gamma})$ such that $\mathcal{D}_v(\hat{\Gamma}) = H$.*

Proof. We will construct $\hat{\Gamma}$ in the following manner. Let $\hat{\Gamma}_1$ be a connected A-labled oriented graph constructed by wedging k A-labled oriented circles together at a vertex v_1. Each circle is to have the lable that corresponds to a generator of H. So, the ith circle is made up of $|h_i|$ edges such that when read from v_1 to v_1, the lable is h_i. With this construction, its easy to see that the lable of every freely reduced path in $\hat{\Gamma}_1$ is a freely reduced word in H, and vice versa. Hence we have $\underline{\mathcal{D}}_{v_1}(\hat{\Gamma}_1) = H$. Now, we construct a sequence of graphs by folding graphs at each stage. At each i, if $\hat{\Gamma}_i$ is folded, we stop the sequence. If not, we construct $\hat{\Gamma}_{i+1}$ by folding. Since each time we fold we decrease the number of edges by 1, and since we began with a finite numebr of edges, this sequence terminates at $\hat{\Gamma}_n$, for some $n \in \mathbb{N}$. Now, since folding preserves connectedness, $\hat{\Gamma}_i$ is connected for each i. Also, by the previous theorem, we have $\underline{\mathcal{D}}_{v_1}(\hat{\Gamma}_1) = \underline{\mathcal{D}}_{v_n}(\hat{\Gamma}_n)$ where v_n is the image of v_1 in $\hat{\Gamma}_n$. Now, since $\hat{\Gamma}_n$ is folded, by lemma 1.11 we have $\underline{\mathcal{D}}_{v_n}(\hat{\Gamma}_n) = \mathcal{D}_{v_n}(\hat{\Gamma}_n)$. Hence we get the chain of equalities

$$H = \underline{\mathcal{D}}_{v_1}(\hat{\Gamma}_1) = \underline{\mathcal{D}}_{v_n}(\hat{\Gamma}_n) = \mathcal{D}_{v_n}(\hat{\Gamma}_n).$$

So, $\hat{\Gamma}_n$ is a connected, folded A-labled oriented graph with vertex v_n such that $\mathcal{D}_{v_n}(\hat{\Gamma}_n) = H$. □

Now we prove that the folded graph that we obtain is unique up to isomorphism. First we define morphisms between graphs.

Definition 3.6. Let Γ and Δ be graphs. Then a map $f : \Gamma \to \Delta$ is a *morphism* if $f(\mathcal{E}(\Gamma)) = \mathcal{E}(\Delta)$, $f(\mathcal{V}(\Gamma)) = \mathcal{V}(\Delta)$, and for every $e \in \mathcal{E}(\Gamma)$, $\iota(f(e)) = f(\iota(e))$ and $\tau(f(e)) = f(\tau(e))$.

Before we get to the actual theorem, we prove two lemmas. The lemmas will actually do most of the work, as we will see later.

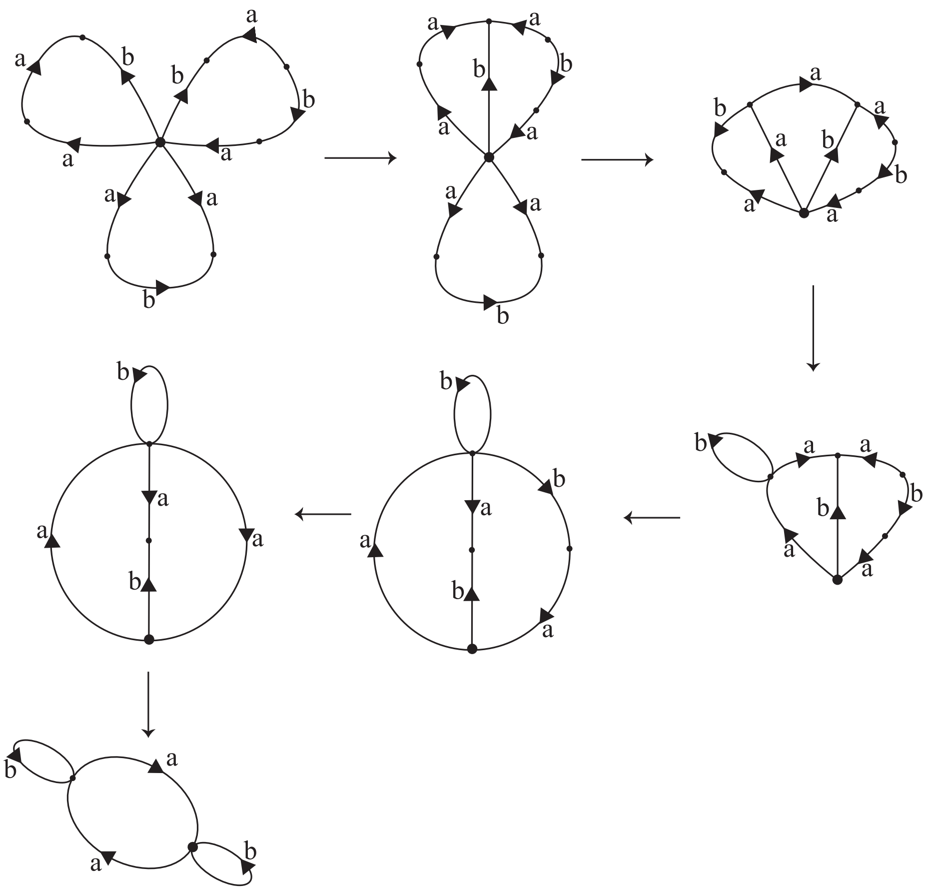

FIGURE 3. Finding the graph for $H = \langle a^2b^{-1}, ba^{-1}ba, ab^{-1}a\rangle$.

Lemma 3.7. *Let $\hat{\Gamma}$, $\hat{\Delta}$ be connected A-labled oriented graphs, with v a vertex of $\hat{\Gamma}$ and x a vertex of $\hat{\Delta}$. Suppose that $\hat{\Delta}$ is folded. Then there exists at most one morphism, $f : \hat{\Gamma} \to \hat{\Delta}$ such that $f(v) = x$.*

Proof. Suppose f, g are morphisms from $\hat{\Gamma}$ to $\hat{\Delta}$ such that $f(v) = g(v) = x$. Let y be an arbitrary vertex in $\hat{\Gamma}$, and let p be a path in $\hat{\Gamma}$ such that $\iota(p) = v$ and $\tau(p) = y$. Now, since $\hat{\Delta}$ is foled, this implies that there cannot be two paths p_1, p_2 such that $\iota(p_1) = \iota(p_2)$, and $l(p_1) = l(p_2)$, since if such two paths existed, we should be able to fold the graph. So, there can be at most one path p' in $\hat{\Delta}$ such that $l(p') = l(p)$. Now, since $l(f(p)) = l(g(p)) = l(p)$, and $\iota(f(p)) = \iota(g(p)) = x$, this implies that $f(p) = g(p)$, and hence $f(y) = g(y)$. Now, we can prove analogously that $f(e) = g(e)$ for any edge $e \in \hat{\Gamma}$. So we have $f = g$, and hence our claim holds. □

The next lemma constructs the morphism that we want. The uniqueness will follow from the previous lemma.

Lemma 3.8. *Let $F(A)$ be a finite rank free group with a finite basis A, and suppose $K \leq H \leq F(A)$ are subgroups of $F(A)$. Further, suppose that $\hat{\Gamma}$ and $\hat{\Delta}$ are connected folded A-labled oriented graphs such that $\mathcal{D}_v(\hat{\Gamma}) = K$ and $\mathcal{D}_x(\hat{\Delta}) = H$, where $v \in \mathcal{V}(\hat{\Gamma})$ and $x \in \mathcal{V}(\hat{\Delta})$. Then there exists a unique morphism $f : \hat{\Gamma} \to \hat{\Delta}$ such that $f(v) = x$.*

Proof. We will first show that f exists. Let $v' \in \mathcal{V}(\hat{\Gamma})$, and let $p_{v'}$ be a reduced path from v to v'. Then this implies that there exists $w \in \mathcal{D}_v(\hat{\Gamma})$ such that the initial subword of $w \in K$ is $l(p_{v'})$. Moreover, since $K \leq H = \mathcal{D}_x(\hat{\Delta})$, there is a unique path $q_{v'}$ in $\hat{\Delta}$ such that $\iota(q_{v'}) = x$ and $l(q_{v'}) = w$. We let $f(v') = \tau(q_{v'})$, and show that $f(v')$ does not depend on our choice of path $p_{v'}$. Let $p'_{v'}$ be another path in $\hat{\Gamma}$ from v to v', and $q'_{v'}$ the corresponding path in $\hat{\Delta}$ with $\iota(q_{v'}) = x$ and $l(q'_{v'}) = w'$. Since $p_{v'}(p'_{v'})^{-1}$ is a loop in $\hat{\Gamma}$, $\underline{w(w')^{-1}} = y \in K = \mathcal{D}_v(\hat{\Gamma})$. This implies that there exists s, a reduced loop based at v such that $l(s) = y$. Then $sp'_{v'}$ is a path from v to $\tau(p'_{v'})$, and the path reduced form of $sp'_{v'}$ has lable $k \cdot w' = w$. Now, since $\hat{\Gamma}$ is folded, this implies that there is only one path with lable w and origin v. Hence the path reduced form of $sp'_{v'}$ is p_v, and hence $\tau(p_{v'}) = \tau(p_v)$, which implies that out definition is well-defined. Now we deal with the edges. Let $e \in \mathcal{V}(\hat{\Gamma})$. Then since $\hat{\Gamma}$ is folded and connected, there exists b a reduced loop based at v such that $b = b_1 e b_2$ where b_1 and b_2 are paths. Since b is a reduced loop based at v, $l(b) \in \mathcal{D}_v(\hat{\Gamma}) = K$. Since $K \leq H$, we also have $l(b) \in H$. This implies that there exist c, a unique reduced loop based at x such that $l(c) = l(b) = l(b_1)l(e)l(b_2)$. Then we can write $c = c_1 e' c_2$, where $l(c_1) = l(b_1)$, $l(e') = e$ and $l(c_2) = l(b_2)$. By our previous vertex construction, we have $\tau(c_1) = f(\iota(e))$. So, we define $f(e) = e'$. Note that we can only have at most one edge $e' \in \mathcal{E}(\hat{\Delta})$ such that $\iota(e') = f(\iota(e))$ and $l(e') = l(e)$, since $\hat{\Delta}$ is folded. Hence our definition is well-defined. So, we have explicitly constructed the function f that we desired. Note that the uniqueness of f follows from the previous lemma. ∎

Now we prove that folded graphs with the same dictionaries at a certain vertex are unique up to isomorphism. The result follows almost trivially from the above two theorems.

Theorem 3.9. *Let $F(A)$ be a free group with finite basis A and let $H \leq F(A)$ be a finitely generated subgroup of $F(A)$. Suppose $\hat{\Gamma}$ and $\hat{\Delta}$ are A-labled oriented graphs such that $\mathcal{D}_v(\hat{\Gamma}) = \mathcal{D}_x(\hat{\Delta}) = H$, where $v \in \mathcal{V}(\hat{\Gamma})$ and $x \in \mathcal{V}(\hat{\Delta})$. Then there exists a unique isomorphism $f : \hat{\Gamma} \to \hat{\Delta}$ such that $f(v) = x$.*

Proof. Since $H \leq H$, by Lemma 3.7, there exists a morphism $f : \hat{\Gamma} \to \hat{\Delta}$ such that $f(v) = x$. Similarly, there exists a morphism $g : \hat{\Delta} \to \hat{\Gamma}$ such that $g(x) = v$. Then by composition we obtain a morphism $(g \circ f) : \hat{\Gamma} \to \hat{\Gamma}$ such that $(g \circ f)(v) = v$. Now, by Lemma 3.6, there is at most one such morphism, which implies that $(g \circ f)$ is the identity morphism. We can prove similarly that $(f \circ g)$ is the idendity morphism. Hence f is an isomorphism. ∎

Finally, we prove that once we create a folded graph with the desired dictionary, we can obtain the free basis of our subgroup and hence its rank. We begin with a lemma.

Lemma 3.10. *Let $\hat{\Gamma}$ be a connected A-labled oriented graph, and $v \in \mathcal{V}(\hat{\Gamma})$. For each $x \in \mathcal{V}(\hat{\Gamma})$ such that $x \neq v$, let p_x be a reduced path in $\hat{\Gamma}$ from v to x. In addition, for each $e \in \mathcal{E}(\hat{\Gamma})$, let $p_e = p_{\iota(e)}e(p_{\iota(e)})^{-1}$, so that p_e is a reduced loop based at v in $\hat{\Gamma}$. Then the subgroup $H = \underline{\mathcal{D}}_v(\hat{\Gamma})$ is generated by the set $\mathcal{X} = \{\underline{l(p_e)} \mid e$ is a positive edge in $\hat{\Gamma}\}$.*

Proof. Let e be a positve edge in $\hat{\Gamma}$. Then let p'_e be the reduced path we obtain from p_e. Then we have $l(p'_e) \in \mathcal{D}_v(\hat{\Gamma})$ and $l(p_e) = l(p'_e)$, and so we have $\underline{l(p_e)} = \underline{l(p'_e)} \in \underline{\mathcal{D}}_v(\hat{\Gamma}) = H$, which implies that $\langle \mathcal{X} \rangle \leq H$. Now, note that by our definition we have $p_{\overline{e}} = (p_e)^{-1}$, and hence $\underline{l(p_{\overline{e}})} = (\underline{l(p_e)})^{-1}$. We show that any element $h \in H$ can be written as a product of $\underline{l(p_e)}$. Since $h \in H = \underline{\mathcal{D}}_v(\hat{\Gamma})$, this means that there exists q, a loop based at v in $\hat{\Gamma}$ such that $\underline{l(q)} = h$. Now let $q = e_1, \cdots, e_k$, where $e_i \in \mathcal{E}(\hat{\Gamma})$. Now, consider the path $q' = p_{e_1} \cdots p_{e_k}$. Note that we can expand q' in the following manner,

$$q' = q_{v_1} e_1 (q_{v_2})^{-1} q_{v_2} e_2 (q_{v_3})^{-1} \cdots q_{v_k} e_k (q_{v_{k+1}})^{-1},$$

where $v_i = \iota(e_i)$, and p_{v_i} is a reduced path from v to v_i for each i. This implies that q' can be reduced to obtain q, and hence we have $\underline{l(q')} = \underline{l(q)} = h$. Also, we have $\underline{l(q')} = \underline{l(p_{e_1})} \cdots \underline{l(p_{e_k})} \in \langle \mathcal{X} \rangle$. This implies that $h \in \langle \mathcal{X} \rangle$. So, we have $H \leq \langle \mathcal{X} \rangle$, and hence we get $\langle \mathcal{X} \rangle = H$. □

All that remains to be proved is that our set is indeed a free basis, not just a generating set.

Theorem 3.11. *Let $F(A)$ be a free group, and $H \leq F(A)$ be a subgroup. Let $\hat{\Gamma}$ be the corresponding folded A-labled oriented graph with $v \in \mathcal{V}(\hat{\Gamma})$ and $\mathcal{D}_v(\hat{\Gamma}) = H$, and let T be a spanning tree of $\hat{\Gamma}$. For each $e \in \mathcal{E}(\hat{\Gamma})$, define $p_e = q_e e q_{e'}$, where q_e is a reduced path in $\hat{\Gamma}$ from v to $\iota(e)$ and $q_{e'}$ is a reduced path in $\hat{\Gamma}$ from $\tau(e)$ to v so that p_e is a reduced loop based at v in $\hat{\Gamma}$ and $l(p_e)$ is a freely reduced word in A. Let X be the set of positively oriented edges of $\hat{\Gamma}$ that are not in T. Then $\mathcal{X} = \{\underline{l(p_e)} \mid e \in X\}$ is a free basis for the subgroup H.*

Proof. Let e be a positive edge in T. Then $p_e = q_{e_1} e q_{e_2}$ can be reduced to the trivial path, and hence we have $\underline{l(p_e)} = 1$. This implies that the edges of $\hat{\Gamma}$ that are in T do not contribute to the group generated by $\{\underline{l(p_e)} \mid e$ is a positive edge in $\hat{\Gamma}\}$. Hence H is generated by $\mathcal{X}$, by the previous lemma. Now we show that $\mathcal{X}$ is indeed a free basis. Let h be a nontrivial freely reduced word in $\mathcal{X}$ where $h = \underline{l(p)}$ for some reduced loop based at v in $\hat{\Gamma}$. Then we can write $h = \underline{l(p_{e_1})} \cdots \underline{l(p_{e_k})}$, where $e_i \in \mathcal{E}(\hat{\Gamma} \backslash T)$, and $e_i \neq (e_{i+1})^{-1}$ for each i. Now, by defnition of p_{e_i}, we can rewrite p in the following form,

$$p = q_{e_1} e_1 q_{e'_1} q_{e_2} e_2 q_{e'_2} \cdots q_{e_k} e_k q_{e'_k}.$$

Note that given a, b, c vertices in a tree, a path from a to b concatenated with a path from b to c can be reduced to a path from a to c. So, since $q_{e'_{i-1}}, q_{e_i}$ are both paths in T, we can reduce the path into q_i, a reduced path in T with $\iota(q_i) = \iota(q'_{e_{i-1}})$ and $\tau(q_i) = \tau(q_{e_i})$. Hence we obtain p' by reducing p, where

$$p' = q_{e_1} e_1 q_2 e_2 q_3 \cdots q_k e_k q_{e'_k}.$$

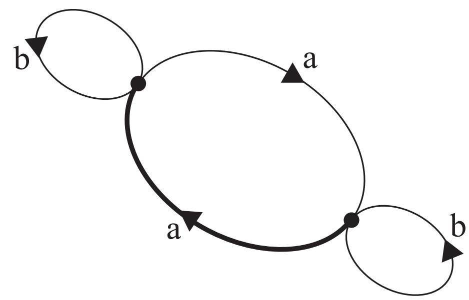

FIGURE 4. Finding the free basis of H from figure 3. We first take a spanning tree, which is bolded, then read loops based at the initial vertex where every edge but one is in the tree, until we cover every edge. Hence the free basis of H is $\langle aba^{-1}, a^2, b\rangle$.

Now, since q_{e_1}, q_i, and $q_{e'_k}$ are all paths in T and e_i are paths in $\hat{\Gamma}\backslash T$, q' is path reduced. Hence p' is a nontrivial loop based at v in $\hat{\Gamma}$. This implies that since $\hat{\Gamma}$ is folded, the lable of p' is a nontrivial word in A. Hence we get $h = \underline{l(p)} = \underline{l(p')} = l(p') \neq 1$. Hence a nontrivial word of $\mathcal{X}$ defines a nontrivial element of $F(A)$, and so $\mathcal{X}$ is a free basis for H. □

Figure 4 is an example of how to find the free basis of a subgroup once you obtain the corresponding folded graph. We essentially read a loop for each edge that were not included in the spanning tree. Note that depending on which spanning tree you take, you may get different bases. Also, note that the generators have become simpler. In our example H, we started with $H = \langle a^2b^{-1}, ba^{-1}ba, ab^{-1}a\rangle$, and found that actually $H = \langle aba^{-1}, a^2, b\rangle$. Hence, the rank of H is 3.

Let us now go over the entire algorithm. Consider $H = \langle a^2b^{-1}, ba^{-1}ba, ab^{-1}a\rangle$, which has been used as our example. Our first goal is to obtain a folded graph that is associated with the subgroup. We did this in figure 3, where the bottom graph is the folded graph we wanted. Then we wanted to obtain a free basis of H by reading the folded graph, which is what we did in fire 4, obtaining $H = \langle aba^{-1}, a^2, b\rangle$. Hence we conclude that the rank of H is 3, which implies that we did not have a redundant element in our initial set of generators.

4. Complexity of the Algorithm

We are interested in giving a bound to the number of folds that we need to make in order to produce a folded graph. As mentioned before, when we fold a graph, the resulting graph has one less edge than the previous graph. Since we begin with a finite number of edges, the most trivial upper bound is the number of edges we begin with. On the other hand, the most trivial lower bound is 0, where we cannot fold at all. There is a better upper bound, but it is not much better, as the following proposition shows.

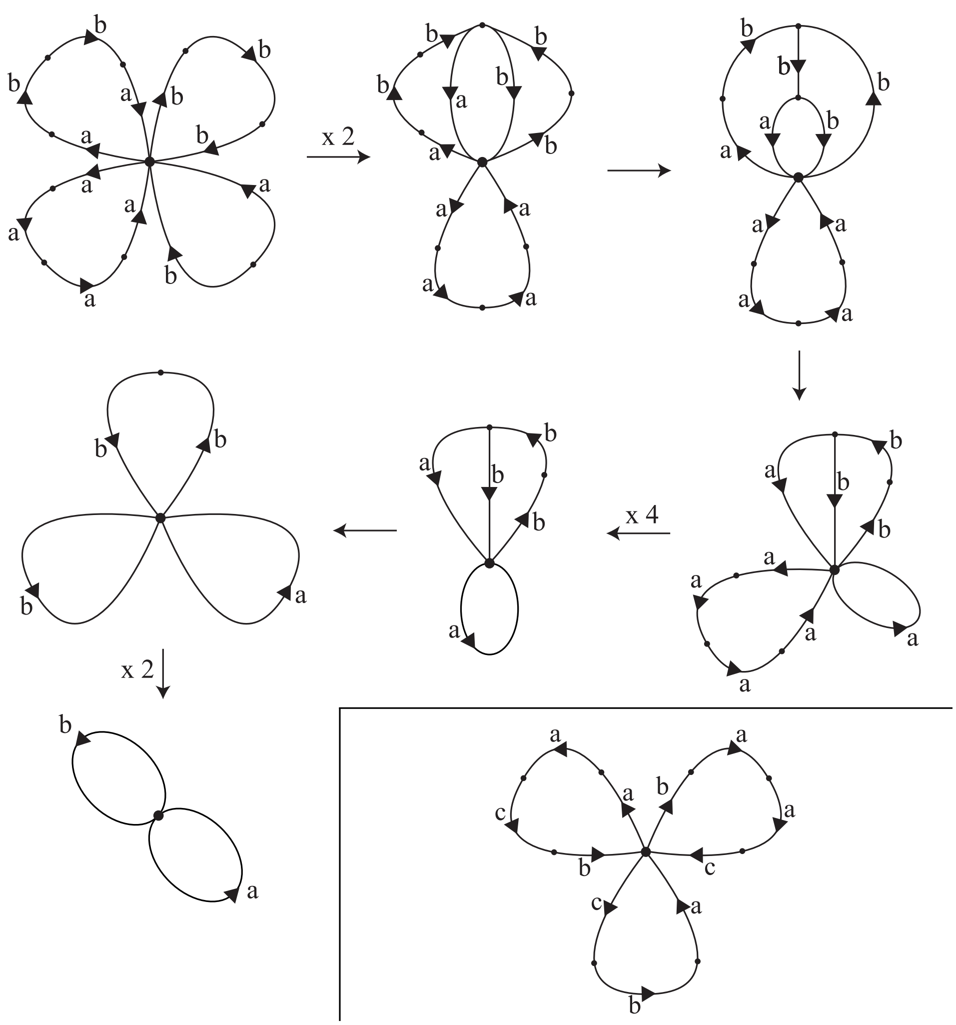

FIGURE 5. The sequence of graphs for $H_1 = \langle ab^2a, b^3, a^{-1}b, a^4\rangle$ and $H_2 = \langle a^2cb, ba^2c, cba\rangle$. H_1 folds until there is one edge per each letter, and H_2 doesn't fold at all. Note that $H_1 = F(\{a, b\})$.

Proposition 4.1. *Let $H = \langle h_1, \cdots, h_k\rangle$. Then if p is the number of folds needed to produce a folded graph by the construction in theorem 2.4, then*

$$0 \leq p \leq \sum_{i=1}^{k} |h_i| - |A|$$

where $|A|$ is the number of letters in A.

Proof. It is easy to see that there are subgroups whose initial wedge-graph cannot be folded. Let $A = \{a_1, \cdots, a_n\}$. Then just consider $\langle a_1a_2, a_2a_3, \cdots, a_{n-1}a_n, a_na_1\rangle$. If $\hat{\Gamma}$ is the graph obtained by wegding circles as shown in theorem 2.4, there are no edges that can be folded. Hence $p = 0$, and so there are no better lower bounds.

Now, for the upper bound, we know that since folding decreases the number of edges by 1, p can be at most the number of edges we begin with, which is $\sum_{i=1}^{k} |h_i|$. But we also know that since folding never eliminates a lable, for every letter in A, there is at least one edge whose lable is that letter. Hence p is at most $\sum_{i=1}^{k} |h_i| - |A|$. $\square$

In figure 5, the initial wedge-graph of H_1 has $|ab^2a| + |b^3| + |a^{-1}b| + |a^4| = 13$ edges, and is folded $13 - |\{a, b\}| = 11$ times until it becomes a folded graph. This is the maximum number of folds that a 13 edge graph can go through. On the other hand, the wedge-graph of H_2 cannot be folded and hence we cannot get a simpler set of generators for H_2. Hence one sees that we cannot do much better than the trivial bound for the number of folds required.