

University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

Dissertations, Theses, and Student Research
Papers in Mathematics

Mathematics, Department of

8-2013

Closure and homological properties of (auto)stackable groups

Ashley Johnson

University of Nebraska - Lincoln, s-aweathe4@math.unl.edu

Follow this and additional works at: <https://digitalcommons.unl.edu/mathstudent>



Part of the [Algebra Commons](#), and the [Geometry and Topology Commons](#)

Johnson, Ashley, "Closure and homological properties of (auto)stackable groups" (2013). *Dissertations, Theses, and Student Research Papers in Mathematics*. 44.
<https://digitalcommons.unl.edu/mathstudent/44>

This Article is brought to you for free and open access by the Mathematics, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in Dissertations, Theses, and Student Research Papers in Mathematics by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

CLOSURE AND HOMOLOGICAL PROPERTIES OF (AUTO)STACKABLE
GROUPS

by

Ashley Johnson

A DISSERTATION

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Doctor of Philosophy

Major: Mathematics

Under the Supervision of Professors Mark Brittenham and Susan Hermiller

Lincoln, Nebraska

August, 2013

CLOSURE AND HOMOLOGICAL PROPERTIES OF (AUTO)STACKABLE GROUPS

Ashley Johnson, Ph. D.

University of Nebraska, 2013

Advisers: Mark Brittenham and Susan Hermiller

Let G be a finitely presented group with Cayley graph Γ . Roughly, G is a *stackable group* if there is a maximal tree T in Γ and a function ϕ , defined on the edges in Γ , for which there is a natural ‘flow’ on the edges in $\Gamma \setminus T$ towards the identity. Additionally, if $\text{graph}(\phi)$, which consists of pairs $(e, \phi(e))$ for e an edge in Γ , forms a regular language, then G is *autostackable*. In 2011, Brittenham and Hermiller introduced stackable groups in [4], in part, as a means to gain traction on the word problem for 3-manifold groups. They showed that if $\text{graph}(\phi)$ is (at least) decidable, as a language, then there is an effective algorithm which solves the word problem; furthermore, they show that stackable groups have an inductive procedure for building van Kampen diagrams, which helps provide insight into the complexity of the word problem.

As one part of my thesis research, I consider group constructions under which the (auto)stackable property is preserved. In this thesis, I show positive results in the case of graph products (a generalization of direct and free products), group extensions and finite index supergroups, and in the case of free products with amalgamation of free abelian groups over an infinite cyclic group. Using closure under group extensions, I also show that polycyclic groups are autostackable, and that there exists an autostackable group with unsolvable conjugacy problem.

Autostackable groups generalize the structures of automatic groups and groups

with finite complete rewriting systems, both of which are known to be of type FP_∞ . However, in this paper, I show that there exists an autostackable group that is not of type FP_3 .

DEDICATION

To Robert, for your unwavering support and belief in me.

ACKNOWLEDGMENTS

First of all, I owe a huge thank you to my advisors Mark Brittenham and Susan Hermiller. Your patience and constant support, from studying for comprehensive exams through these final edits of my dissertation, has been amazing. I truly believe that I could not have picked a better pair of advisors under whom to study. I look forward to continuing to work with you in the future! I would also like to thank Jamie Radcliffe and John Meakin for being readers of this dissertation, and Brian Harbourne for being an outstanding teaching mentor. I am also grateful to all of the UNL math faculty members for making my experience at UNL such a great one, but especially Judy Walker for all the help during the job search and Dave Skoug for talking baseball with me on demand. Also, thanks to Marilyn for being my go-to person for an ear to rant to, a shoulder to cry on and, of course, chocolate.

My fellow graduate students contributed a lot to what made my time in graduate school so wonderful. Melanie and Katie, thank you for distracting me when I needed distracting (you will always be my crossword buddies!), encouraging me when I got discouraged and slapping some sense into me when I threatened to quit. Melanie, I owe you an extra thank you for always dropping everything to help me when I thought I broke my dissertation. Amanda, thank you for being my mathematical role model; I have learned so much from you over the years. I will miss my shopping and yoga buddy, but there is always FaceTime! Courtney, thank you for making me laugh when I needed it most, and, of course, for being my seminar buddy. It was really great to get to know you these last few years, and I look forward to being a NExT fellow with you.

Thank you to my mathematical siblings, Dave McCune, Anisah, Nathan and Melanie. You all have been a great resource to bounce ideas (sometimes bad ones)

off of and I am incredibly appreciative for that. Also, a big thanks to Derrick Stolee for answering my numerous questions about language theory. Ben, thank you for cheering me up when I needed it (and even when I didn't!). I would also like to thank Sara, Lauren, Becky, Nora and Christina for being great friends and Mathlete teammates. While I am on the topic, I want to thank everyone who participated in or cheered for our Mathlete sports program. Our sporting events always gave me something to look forward to, and grad school would have not been the same without all of you! Riemann, Hilbert, Nakayama! To my Grace Chapel softball teammates, thank you so much for inviting me into your family. I hope I am lucky enough to find a new team half as awesome as you guys in Alabama.

I am also very grateful to many people that helped me get to graduate school in the first place. I want to thank Samantha Herrington for going above and beyond the duties of a teacher and helping me to find scholarship money to take my first college math course. Jackie Jensen, Angie Brown, Brian Loft and Mietek Dabkowski have been great role models and mentors throughout college and graduate school. I should also credit the Nebraska Conference for Undergraduate Women in Mathematics for further selling me on the idea of graduate work and on the University of Nebraska in particular.

Finally, thank you to my family. You have supported and encouraged me from day one and I followed my dreams because of you. My older brother Erik gets a special shout out for getting me interested in mathematics in the first place. And last but certainly not least, thank you to my husband, Robert. You have sacrificed a lot for me throughout graduate school and I would not have made it through without your support.

Contents

Contents	vii
1 Introduction	1
1.1 Background	1
1.2 Formal Language Theory	9
2 Closure Properties	14
2.1 Graph Products	14
2.2 Group Extensions	24
2.3 Finite Index Supergroups	34
3 Homological Properties	41
4 Examples	51
Bibliography	59

Chapter 1

Introduction

1.1 Background

In [4], Brittenham and Hermiller define *stackable* groups as a generalization of the algorithmic structures of almost convex groups and groups with finite complete rewriting systems. They show in [4] that almost convex groups and groups with finite complete rewriting systems are (algorithmically) stackable. They refine stackability into autostackability in [5]. The theory of autostackable groups is partly motivated by a desire to gain traction on the word problem, specifically in the class of fundamental groups of closed 3-manifolds. Automatic groups were defined by Epstein, Cannon, Holt, Levy, Paterson and Thurston in [12] with the same motivation in mind, but the class of automatic groups fails to include fundamental groups of closed 3-manifolds with Nil or Sol geometries. Later on, we discuss some differences between automatic groups and autostackable groups, but first we require some definitions and notation.

Let G be a group with finite inverse-closed generating set A and let Γ be the associated Cayley graph of G . A set of *normal forms* for G over A is a collection of canonical representations of the group elements in G as words in A^* . Throughout

this paper, we label the normal form for an element $g \in G$ as the word y_g . If two words u and v represent the same group element, then we write $u =_G v$. If they are identically the same word, then we write $u = v$. Given a word w , we will use \bar{w} to represent the group element g such that $g =_G w$. That is, if $G = \langle A \rangle$, then $\bar{\cdot} : A^* \rightarrow G$ is the quotient map that takes words to group elements. By slight abuse of notation, we will likewise use y_w to represent the normal form for the group element \bar{w} .

Let $\vec{E}(\Gamma) = \vec{E}$ be the set of directed edges in the Cayley graph Γ . Let $e_{g,a}$ represent the directed edge with initial vertex g and terminal vertex ga labeled by a . Again by slight abuse of notation, we will write $e_{w,a}$ to represent the directed edge labeled by a with the group element \bar{w} as the initial vertex. We call an edge $e_{g,a}$ *degenerate* if either $y_g a = y_{ga}$ or $y_{ga} a^{-1} = y_g$. The collection of degenerate edges is labeled the set \vec{E}_d . If an edge is not degenerate, then it is *recursive* and we define $\vec{E}_r = \vec{E} \setminus \vec{E}_d$ to be the collection of all recursive edges.

An intuitive description of stackability is as follows. Let $G = \langle A \rangle$ be a group with A a finite, inverse-closed generating set, and let Γ be its corresponding Cayley graph with directed edge set \vec{E} . Define a function $\phi : \vec{E} \rightarrow \{\text{paths in } \Gamma \text{ of length } \leq k\}$ for some $k \in \mathbb{N}$. A stackable structure for G is the function ϕ together with a choice of maximal tree T in Γ such that the function ϕ acts as a ‘flow’ of the edges in $\vec{E} \setminus T$ towards the identity. If, in addition, the graph of ϕ is a regular language, then G has an autostackable structure.

For example, consider the group $G = \mathbb{Z}^2$ generated by two elements a and b with a single commuting relation $ab = ba$. The Cayley graph for this presentation is the 2-dimensional lattice, and $\mathcal{N} = \{a^i b^j \mid i, j \in \mathbb{Z}\}$ is a set of normal forms for G . Since the set \mathcal{N} dictates that the normal form for $a^i b^j \cdot a$ is $a^{i+1} b^j$, the edge labeled by a between $a^i b^j$ and $a^{i+1} b^j$ is recursive. In fact, all recursive edges in this example are of that form. Suppose e is an edge labeled by a in the upper half plane. Then $e = e_{a^i b^j, a}$

with $j > 0$. The flow from the edge e is downward, toward the horizontal axis, and the function indicating the flow is $\phi(e) = b^{-1}ab$. In the Cayley graph, this amounts to taking a “step” down from one recursive edge to the next below until you eventually end up on the horizontal axis which consists of all edges in the tree which are traced out by words in \mathcal{N} . Additionally, $\text{graph}(\phi)$ is a regular language, as $\phi(e)$ depends only on the edge label and whether the edge is above the a -axis ($j > 0$) or below the a -axis ($j < 0$).

Brittenham and Hermiller introduced stackable groups in [4], with a flow function labeled c defined only on the set \vec{E}_r of recursive edges. It was then extended to a function c' on all edges in the Cayley graph. There is a natural one to one correspondence between edges in the Cayley graph and the set $\mathcal{N} \times A$, which associates a pair (y, a) with the edge $e_{y,a}$, labeled by a , with initial vertex labeled by \bar{y} . We now have the following definition, which is introduced in [5].

Definition 1.1. *Let G be a group with finite inverse-closed generating set A . Then G is autostackable if there is a set of normal forms \mathcal{N} for G over A with $1 \in \mathcal{N}$, a constant k and a function $\phi : \mathcal{N} \times A \rightarrow \{\text{words in } A^* \text{ of length } \leq k\}$ such that:*

1. *The set $\text{graph}(\phi) := \{(y_g, a, \phi(y_g, a)) | g \in G, a \in A\}$, is a regular language when viewed as a set of words over a padded set $(A \cup \$)^3 \setminus \{(\$ \$ \$)\}$.*

2. *For each $e_{g,a} \in \vec{E}(\Gamma)$, we have $\phi(y_g, a) =_G a$ and*

(2d) *We have $\phi(y_g, a) = a$ if and only if $e_{g,a} \in \vec{E}_d$*

(2r) *The transitive closure $<_\phi$ of the relation $<$ on recursive edges \vec{E}_r , defined by $e' < e_{g,a}$ whenever $e_{g,a}, e' \in \vec{E}_r$ and e' is on the path $\phi(y_g, a)$ from the initial vertex of $e_{g,a}$, is a strict well-founded partial ordering.*

If (2) holds, but (1) does not, then we say that G is stackable. If (2) holds, and the language described in (1) is decidable but not regular, then we say that G is algorithmically stackable.

We refer to the function ϕ as either the “flow” function or the “stacking” function. Because the edges $e_{g,a}$ are in a one-to-one correspondence with the pairs (y_g, a) , we will sometimes use $\phi(e_{g,a})$ to mean $\phi(y_g, a)$. When a directed edge e' lies along the path $\phi(e)$, we call e' a *child* of e .

Recall that a *van Kampen diagram* Δ for a word $w =_G 1$ is a connected, simply connected, planar 2-complex such that the edges of the boundary of Δ , $\partial\Delta$, are labeled by w and for every 2-cell, σ of Δ , the edges of $\partial\sigma$ are labeled by a relator in R . For more information about van Kampen diagrams, see [23]. In [4], Brittenham and Hermiller show that having a stacking structure gives an inductive procedure to build van Kampen diagrams. In short, stackability gives a way to fill “icicles”, the van Kampen diagrams for words of the form $y_g a y_{ga}^{-1}$, and using the *seashell* method general van Kampen diagrams are filled using these icicles. For more detail, see [4].

In the same paper, Brittenham and Hermiller present a *stacking reduction procedure* which acts as a methodical way to put a word into its normal form. The stacking reduction procedure can be interpreted visually: trace out a word w in the Cayley graph. For each edge e along the path traced by w that is recursive, replace it with the path $\phi(e)$ in the Cayley graph described above. This procedure does not change the group element that the word represents, as the path begins and ends at the same vertices as e . Continue this process until all recursive edges have been rewritten; the result is a path consisting only of degenerate edges. The final step is to freely cancel, thus removing any edge traversed multiple times. The final path labels the normal form word y_w . Since any algorithm which can find normal forms can find the normal

form for a word representing $1 \in G$, this stacking reduction procedure provides a solution to the word problem. With the added restriction that the language $\text{graph}(\phi)$ be (at least) decidable, this algorithm is effective. Since decidable languages contain the class of regular languages, autostackable groups also have this effective solution to the word problem.

Throughout this paper, we require a set of normal forms of a stackable group to be *prefix-closed*. That is, if $w \in \mathcal{N}$ decomposes as $w = uv$ for words $u, v \in A^*$, then $u \in \mathcal{N}$. While the definition does not strictly require prefix-closed normal forms, Brittenham and Hermiller show that stackable groups have prefix-closed normal forms as a consequence of the stacking reduction procedure [4, Lemma 1.5].

We now have the language to compare autostackable groups with automatic groups. In [5], Brittenham and Hermiller show that all automatic groups with prefix-closed automatic structures are autostackable. However, the class of autostackable groups is strictly larger than the class of automatic groups with prefix-closed automatic structures. For example, all fundamental groups of closed 3-manifolds with uniform geometries are autostackable [5]. In [12], Epstein et. al. show that all automatic groups satisfy a quadratic Dehn function; the only known restriction on the Dehn function of autostackable groups is that it be computable. In fact, the iterated Baumslag Solitar groups presented by

$$\langle a_0, a_1, \dots, a_k | a_{i+1}^{-1} a_i a_{i+1} = a_i^2, 0 \leq i \leq k-1 \rangle$$

were shown by Gersten in [14] to have Dehn function given by a k -fold iterated exponential and by Brittenham and Hermiller in [5] to be autostackable. Therefore iterated exponential functions provide a lower bound on the upper bound for the Dehn function of an arbitrary autostackable group. For more information about Dehn functions,

see [23]. The question of whether or not all automatic groups are autostackable reduces to the question of whether or not every automatic group possesses an automatic structure that includes a set of prefix-closed normal forms.

As stated before, one goal of autostackable groups is to gain traction on algorithmic problems in the class of fundamental groups of closed 3-manifolds. As the fundamental group of a closed 3-manifold can be realized as the fundamental group of a graph of groups with edge groups 1 , \mathbb{Z} or \mathbb{Z}^2 and vertex groups G where G is the fundamental group of a closed 3-manifold group with uniform geometry, closure properties are of particular interest here.

In Chapter 2, we investigate closure properties of stackable, algorithmically stackable and autostackable groups. We have the following results.

Theorems 2.6 and 2.7. *Let groups G_1, G_2, \dots, G_n be stackable (respectively autostackable, algorithmically stackable) with finite inverse-closed generating sets A_1, A_2, \dots, A_n , and stacking structures $(\mathcal{N}_1, \phi_1), (\mathcal{N}_2, \phi_2), \dots, (\mathcal{N}_n, \phi_n)$ respectively. Then any graph product, G , of the groups G_1, \dots, G_n is also stackable (respectively autostackable, algorithmically stackable).*

Theorems 2.10 and 2.11. *Let $1 \rightarrow K \xrightarrow{\iota} G \xrightarrow{q} Q \rightarrow 1$ be an exact sequence for groups K, G and Q . If $K = \langle A | R \rangle$ and $Q = \langle B | S \rangle$ are stackable (respectively autostackable, algorithmically stackable) with A and B finite inverse-closed generating sets and stacking structures (\mathcal{N}_K, ϕ_K) and (\mathcal{N}_Q, ϕ_Q) respectively, then $G = \langle A \cup \hat{B} \rangle$ is stackable (respectively autostackable, algorithmically stackable).*

Theorems 2.16 and 2.17. *Let G be a stackable (respectively autostackable, algorithmically stackable) group with finite inverse-closed generating set A and stacking structure (\mathcal{N}_G, ϕ_G) . If G is a finite index subgroup of a group H , then H is also stackable (respectively autostackable, algorithmically stackable).*

From Theorem 2.11, we have the following two corollaries.

Corollary 2.12. *Let G be a polycyclic group. Then G is autostackable.*

Corollary 2.13. *There exists an autostackable group which does not have solvable conjugacy problem*

Closure properties in related classes of groups have been well-studied. In [12], Epstein et.al. show that automatic groups are closed under free and direct products, finite index subgroups, finite index supergroups and HNN extensions and free products with amalgamation over finite subgroups. In [5], Brittenham and Hermiller show that groups with finite complete rewriting systems are autostackable. In [18], Hermiller and Meier show that groups with finite complete rewriting systems are closed under graph products. In [17], Groves and Smith show closure under finite index subgroups and certain amalgamated products and HNN extensions. In [19], Hermiller and Meier provide a different proof of closure under finite index subgroups.

Since autostackable groups are closely related to automatic groups and groups with finite complete rewriting systems, it is natural to investigate which properties autostackability shares with these two families of groups. In Chapter 3, we examine the following.

Definition 1.2. *A group G is of type FP_n ($n \geq 1$) if the $\mathbb{Z}G$ -module \mathbb{Z} admits a projective resolution which is finitely generated in all dimensions $\leq n$.*

In the finitely presented case, this is equivalent to:

Definition 1.3. *A group G is of type F_n ($n \geq 1$) if there exists a $K(G, 1)$ with finite n -skeleton, where a $K(G, 1)$ is path connected space with contractible universal cover whose fundamental group is G .*

For more information about type FP_n and group cohomology, see Brown's text *Cohomology of groups* [6]. Both automatic groups and groups with finite complete rewriting systems are of type FP_∞ . These results were shown for automatic groups by Alonso in [1] and for groups with finite complete rewriting systems in Anick [2], Brown [7], Groves [16], Kobayashi [21], Farkas [13], and Lafont [22]. It should be noted that Squier in [26] shows that groups with finite complete rewriting systems are of type FP_3 ; this is one of the first results on homological properties of groups with finite complete rewriting systems. For more information, see the survey by Daniel E. Cohen [9]. However, we have the following result for autostackable groups.

Theorem 3.4. *There exists an autostackable group that does not have type FP_3 .*

In 1963, John Stallings provided the first example of a finitely presented group which was of type FP_2 but not of type FP_3 [27]. It is shown here that Stallings' group S is autostackable. While this is a somewhat surprising result, Stallings' group is the first group to show that the class of autostackable groups contains more than the union of the class of automatic groups with prefix-closed automatic structures and the class of groups with finite complete rewriting systems.

Some years after Stallings' original paper, Robert Bieri realized Stallings' group as the kernel of a particular map from $F_2 \times F_2 \times F_2 \rightarrow \mathbb{Z}$, and expanded this idea to create a family of groups, called the Bieri-Stallings groups, which are of type FP_n but not of type FP_{n+1} [3]. Stallings' original not FP_3 group, S , has been well-studied. In [10], Dison, Elder, Riley and Young show that S has quadratic Dehn function. In [11], Elder and Hermiller show that S is not minimally almost convex on the generating set used in [27].

The group \mathbb{Z} is an autostackable group. Let $\mathbb{Z} = \langle a \rangle$ with normal form set $\{a^i | i \in \mathbb{Z}\}$. Then there are no recursive edges to stack, and the function $\phi(e_{g,a}) = a$ is a

stacking function on \mathbb{Z} . Using Theorem 2.7, we have closure of the autostackable property under direct products, and so the group \mathbb{Z}^n is autostackable for any $n \in \mathbb{N}$. In Chapter 4, we show the following class of groups is autostackable.

Theorem 4.1. *Let $G = G_1 *_H G_2$ with $\alpha : H \hookrightarrow G_1$ and $\beta : H \hookrightarrow G_2$ any injective homomorphisms. Then if $G_1 = \mathbb{Z}^n$, $G_2 = \mathbb{Z}^m$ and $H = \mathbb{Z}$, then G is autostackable.*

1.2 Formal Language Theory

This section contains a review of necessary facts from formal language theory. Informally, the theory of formal languages looks at recognition of subsets of words in an alphabet under different models of computation. For example, a *regular* language is one for which a machine with bounded memory, called a finite state automaton, can be used to determine whether a given word is in the language. A language is *decidable* if there is a machine with unbounded memory, called a Turing machine, which can determine membership in the language. The book *Word Processing in Groups* by Epstein et. al. [12] provides a nice introduction to the theory of regular languages, and the book *Introduction to the Theory of Computation* by Sipser [25] provides a nice introduction to both regular languages and decidable languages. We include many of their definitions here.

A *string* or *word* over a set (alphabet) A is a concatenation of elements (letters) from A . For example, if $A = \{a, b\}$ then $aaaa$ and $ababaab$ are both words, over A . The word $aaaa$ is usually denoted by a^4 . Let ϵ denote the word with no letters. The collection of all strings over an alphabet A is denoted A^* . Note that the empty word ϵ is in the set A^* . A *language* over A is any subset (including the empty subset) of A^* . Given two languages K and L , over the same alphabet A , the *concatenation* $K \cdot L$, or KL , is the set of words w for which $w = w_1w_2$ in A^* with $w_1 \in K$ and

$w_2 \in L$. Define the union, \cup , and the intersection, \cap , of two languages L and K exactly as expected, as languages are sets. The *Kleene star*, sometimes called the Kleene closure, is defined by

$$K^* = \bigcup_{n \geq 0} K^n$$

where K^n is the concatenation of K with itself n times. When $n = 0$, then $K^0 = \{\epsilon\}$ contains only the empty word. Finally, the complement of a language L is the set $L^c := A^* \setminus L$.

When working with languages with many variables, we can think of the language as padded strings. To each alphabet, A_i , add an additional padding symbol $\$$. We use these extra symbols to take an n -tuple whose entries have multiple bits and write it as a product of n -tuples whose entries consist of a single letter. For example, suppose we have the 3-tuple $(abab, a, b^{-1}ab)$. Here all entries are over the same alphabet, so we use a single padded letter $\$$. We have the following equivalence.

$$(abab, a, b^{-1}ab) = (a, a, b^{-1}) \cdot (b, \$a) \cdot (a, \$, b) \cdot (b, \$, \$).$$

For more information about padded languages, see Section 1.4 in [12].

Definition 1.4. A language L over an alphabet A is regular if it can be built from finite subsets of A^* using the operations \cup , \cap , \cdot , c and $*$.

An expression using the above operations is called a *regular expression*. For example, let $A = \{a, b\}$. Then the language $\{a^i b^j | i, j \in \mathbb{N}_0\}$ is regular, as it can be written as the following regular expression:

$$\{a^i b^j | i, j \in \mathbb{N}_0\} = \{a\}^* \cdot \{b\}^*$$

We often think of regular languages as those languages which can be accepted by a machine with a limited amount of memory. With this idea in mind, it is not too hard to see that the language $\{a^i b^i \mid i \in \mathcal{N}\}$ is an example of a language which is *not* regular, as a machine would have to keep track of the (arbitrarily large) number i .

From the definition of a regular language, we can see that regular languages will be closed under the finite union, finite intersection, concatenation, complementation and Kleene star. Regular languages are closed under homomorphic images and inverse images. For a more complete discussion, see Chapter 1 of [12].

A larger class of languages are those which are *decidable*. A language L is decidable if it is the language accepted by a Turing Machine. For the purposes of this paper, we will not provide the full definition of a Turing Machine. We will instead say that a language L is decidable if there is an algorithm to decide membership in L . For a full introduction to Turing Machines, and more information on regular languages, consult Michael Sipser's *Introduction to the Theory of Computation* [25]. Decidable languages are also closed under finite union, finite intersection, concatenation, complementation, and Kleene star [25]. Decidable languages are also closed under inverse images of homomorphisms. Suppose $f : L \rightarrow L'$ be a homomorphism, and suppose that $N \subset L'$ is a decidable language. As N is decidable, given a $w \in L$, we can decide whether or not $f(w)$ is in N . If it is, then $w \in f^{-1}(N)$ and if it isn't, then $w \notin f^{-1}(N)$. Hence $f^{-1}(N)$ is decidable. In general, decidable languages are not necessarily closed under homomorphic images. Define the map $\rho_1 : ((A \cup \$)^3)^* \rightarrow A^*$ to be projection onto the first coordinate, with the symbol $\$$ mapping to 1. If the language $graph(\phi) = \{(y, a, \phi(y, a)) \mid y \in \mathcal{N}, a \in A\}$ is decidable, the image $\rho_1(graph(\phi))$ is also decidable. Let $w \in A^*$. Since there are finitely many pairs (a, x) for x a word in the image of ϕ , test each pair (w, a, x) for membership in $graph(\phi)$ for all possible pairs (a, x) . If the 3-tuple is in $graph(\phi)$, then $w \in \mathcal{N}$. If not, then $w \notin \mathcal{N}$. A similar argument works

in the case of the image under ρ_1 of sublanguages of $\text{graph}(\phi)$.

Lemma 1.5. *Let $G = \langle A | R \rangle$ be a stackable group with stacking structure (\mathcal{N}, ϕ) . The set $\text{graph}(\phi) = \{(y_g, a, \phi(y_g, a)) | g \in G, a \in A\}$ is regular (respectively decidable) if and only if the sets $S_{a,x} = \{y | (y, a, x) \in \text{graph}(\phi)\}$ are regular (respectively decidable) for each fixed pair of $a \in A, x \in \text{im}(\phi)$.*

Proof. Let $\rho_i : ((A \cup \{\$\})^3)^* \rightarrow A^*$ be the natural extension of the monoid homomorphism that projects onto the i th coordinate which maps the symbol $\$$ to 1. For a fixed $a \in A$ and $x \in \text{im}(\phi)$, define

$$T_{a,x} = \text{graph}(\phi) \cap \rho_2^{-1}(\{a\}) \cap \rho_3^{-1}(\{x\}).$$

Then $T_{a,x}$ is the collection of all padded 3-tuples in $\text{graph}(\phi)$ with a and x fixed as the second and third coordinates, respectively. Observe that $S_{a,x} = \rho_1(T_{a,x})$, where $T_{a,x}$ is a sublanguage of $\text{graph}(\phi)$. Then using the closure properties of regular and decidable languages, for a fixed $a \in A, x \in \text{im}(\phi)$, $S_{a,x}$ is regular (respectively decidable) provided that $\text{graph}(\phi)$ is regular (respectively decidable).

Conversely, for a fixed $a \in A$ and $x \in \text{im}(\phi)$, define

$$P_{a,x} := \rho_1^{-1}(S_{a,x}) \cap \rho_2^{-1}(\{a\}) \cap \rho_3^{-1}(\{x\}).$$

Observe that the languages $\{a\$^*\} = \{a\} \cdot \{\$\}^*$, and $\{x\$^*\} = \{x\} \cdot \{\$\}^*$ are both regular, and thus also decidable. Note also that $P_{a,x}$ consists of exactly those tuples in $\text{graph}(\phi)$ which have the second and third coordinates fixed as a, x respectively. Letting a, x range over all of A and $\text{im}(\phi)$ we get

$$\text{graph}(\phi) = \bigcup_{a \in A} \bigcup_{x \in \text{im}(\phi)} P_{a,x}.$$

Finally, assuming that $S_{a,x}$ is regular (respectively decidable), the closure properties of regular and decidable languages show that $graph(\phi)$ is also regular (respectively decidable). \square

Chapter 2

Closure Properties

This chapter discusses the question of closure of stackability, algorithmic stackability and autostackability under various group constructions. We show closure of all three properties under the graph product, group extension and finite index supergroup constructions.

2.1 Graph Products

We begin with a description of the graph product construction. Recall that a simple graph is an undirected graph with no loops or multiple edges.

Definition 2.1. *Let Λ be a finite simple graph with vertex set $V(\Lambda) = \{v_1, \dots, v_n\}$ and edge set $E(\Lambda) = \{e_1, \dots, e_k\}$. To each vertex v_i , associate a group $G_i = \langle A_i | R_i \rangle$. The graph product of the groups $\{G_i\}_{i=1}^n$ with respect to Λ is the group generated by the vertex groups with the added relations that G_i commutes with G_j if vertices v_i and v_j are connected by an edge in Λ .*

Graph products were introduced by Elisabeth Green in [15] in 1990 as a generalization of direct products and free products. A graph Λ with no edges defines a

free product of the vertex groups and a graph Λ' which is complete (includes edges between every pair of vertices) defines a direct product of the vertex groups. Right angled Artin groups are graph products in which the defining graph has all vertex groups labeled \mathbb{Z} and right angled Coxeter groups are graph products in which the defining graph has all vertex groups labeled $\mathbb{Z}/2\mathbb{Z}$.

Before we discuss normal forms for a graph product, we need to set up some notation. We follow the terminology of [18]. Given a word w in the graph product, a subword w' of w is a *local word* if it is a longest possible non-empty subword written with generators from a single vertex group. The *type* of a local word is the index of its associated vertex group. Define a monoid homomorphism τ from all words A^* into the quotient monoid $\langle 1, \dots, n \mid i^2 = i \text{ for } i = 1, \dots, n \rangle$ where n is the number of vertices in the graph Λ defining the graph product. Given a generator $a \in A_i$, we define $\tau(a) = i$. Extend the definition to all words in A^* in the natural way. The image $\tau(w)$ is exactly the string of word types which make up the word w .

Example 2.2. Let G be the graph product defined by Figure 2.1

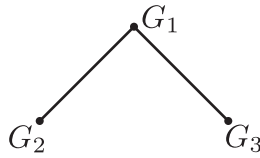


Figure 2.1: The graph Λ defining G

for $G_1 = \langle a \mid \rangle$, $G_2 = \langle b \mid \rangle$, and $G_3 = \langle c \mid \rangle$. Then $G = \langle a, b, c \mid ab = ba, ac = ca \rangle$. Consider the word $w = a^3 b^{-2} a^2 c^5$. Then $w' = b^{-2}$ is a local word, as are a^3, a^2 and c^5 . The types of these local words are 2, 1, 1 and 3, respectively. The value of $\tau(w)$ then, is 2113.

To get normal forms for a graph product, we need an ordering on the vertex groups. Given groups G_1, \dots, G_n , let $<_G$ be a total ordering on this finite set of groups. Let $A = \bigcup_{i=1}^n A_i$. The following set \mathcal{N} of words in A^* is shown to be a set of normal forms for G by Hermiller and Meier in [18, Prop 3.2]; a topological proof of these normal forms is provided by Hsu and Wise in [20]

Definition 2.3. *Given a set of normal forms, \mathcal{N}_i , for each vertex group, and a total ordering $<_G$ on the vertex groups, a word w in A^* is in the set \mathcal{N} if*

1. *each local word of w is a normal form in its respective vertex group*
2. *if $w = \dots w_i \dots w_j \dots$ with w_i, w_j local words of type i, j respectively, with $G_j <_G G_i$ or $i = j$, then there is a local word w_k of type k such that $w = \dots w_i \dots w_k \dots w_j \dots$ with vertices v_j and v_k nonadjacent.*

We can refer to the word w_k described in (2) as a *barrier*. In Example 2.2, the word w is not in normal form because a^3 and a^2 are separated by a local word b^{-2} , with which a^2 commutes. Using the ordering $G_1 <_G G_2 <_G G_3$, the normal form for the word w is $y_w = a^5 b^{-2} c^5$.

We now develop the concept of a *banned string*. The set of banned strings is the collection of all subword types which do not result in a normal form. Define the set $I_j := \{k | v_k \text{ and } v_j \text{ are connected by an edge in } \Lambda\}$ and the set $P_j := \{k | k \in I_j \text{ and } G_j <_G G_k\}$. Finally, define the set of *banned strings*, \mathbb{B} to be the set

$$\mathbb{B} = \bigcup_{j=1}^{|V(\Lambda)|} \left(\bigcup_{k \in P_j} k \cdot \{i | i \in I_j\}^* \cdot j \right)$$

A complementary idea, termed admissible strings, were developed by Hermiller and Meier in [18]. The collection of admissible strings are those strings of word types which do not contain a banned string.

As an example, let G be the graph product of groups G_1 , G_2 and G_3 defined in Figure 2.1 with the ordering $G_1 <_G G_2 <_G G_3$. We have that $I_1 = P_1 = \{2, 3\}$ and both P_2 and P_3 are empty. From these sets we can build \mathbb{B} as

$$\mathbb{B} = 2\{2, 3\}^*1 \cup 3\{2, 3\}^*1$$

Any word whose word type contains a string in \mathbb{B} is not in normal form. Note that excluding banned strings is not enough to check that a word is in normal form. It also must be checked that each local word is also in its normal form.

Notation 2.4. For a given word $w \in A^*$, we represent the maximal suffix of w in the letters A_i as $\text{suf}_i(w)$. That is, if $v = \text{suf}_i(w)$, then either $v = 1$ or v is the local word of type i at the end of w .

Now that we have established a set of normal forms, we determine the set of recursive edges.

Lemma 2.5. Let groups G_1, G_2, \dots, G_n be generated by finite inverse-closed sets A_1, A_2, \dots, A_n with normal form sets $\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_n$ respectively. Let G be any graph product of the groups G_1, \dots, G_n with normal form set described in Definition 2.3.

Define the following sets:

$$\begin{aligned} \mathcal{A} &= \{e_{g,a} \mid y_g = wv, \text{ where } \tau(v) = i, a \in A_i \text{ and } va, y_v a^{-1} \notin \mathcal{N}_i\} \\ \mathcal{B} &= \{e_{g,a} \mid y_g = wv \text{ with } \tau(v) = j, a \in A_i, \text{ and } i \neq j, \text{ where } \tau(y_g)i \\ &\quad \text{does not contain a banned string and } a \notin \mathcal{N}_i\} \end{aligned} \quad (\dagger)$$

$$\mathcal{C} = \{e_{g,a} \mid a \in A_i \text{ and } \tau(y_g)i \text{ contains a banned string}\}$$

Then the set of recursive edges is exactly $\vec{E}_r = \mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$.

Proof. (Of Lemma 2.5) Let \mathcal{A} , \mathcal{B} and \mathcal{C} be as defined in (\dagger) . First, we show $\mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$ is contained in \vec{E}_r . Let $e_{g,a} \in \mathcal{A}$ for $a \in A_i$. Then we can write y_g as $y_g = wv$ for

v a local word of type i . The circumstances of set \mathcal{A} tell us that va and $y_{va}a^{-1}$ are not in \mathcal{N}_i and so by definition of \mathcal{N} , we know that neither y_ga nor y_ga^{-1} are in \mathcal{N} . Thus $e_{g,a} \in \vec{E}_r$. Let $e_{g,a} \in \mathcal{B}$ for $a \in A_i$ with $y_g = wv$ for v a local word of type j with $i \neq j$. The assumptions on this set have that $\tau(y_g)i$ does not contain a banned string. However, because the word a is not a normal form, we have $y_ga \neq y_{ga}$. Moreover, since $i \neq j$, the word y_g does not end with a^{-1} . Thus $e_{g,a} \in \vec{E}_r$. Lastly, let $e_{g,a} \in \mathcal{C}$ for $a \in A_i$. Then by the definition of set \mathcal{C} , the string of word types of $\tau(y_g)i$ contains a banned string, and therefore $y_ga \neq y_{ga}$. Lastly, as $y_g \in \mathcal{N}$, we know that $\tau(y_g)$ does not contain a banned string. However, since $\tau(y_g)i$ does contain a banned string, the word y_g cannot end with the letter a^{-1} . This shows that $e_{g,a} \in \vec{E}_r$.

Let $e_{g,a}$ be a recursive edge with $a \in A_i$. Then by definition, y_ga is not in normal form, nor does y_g end with the letter a^{-1} . However, we can see from the normal form set that either the $\tau(y_g)i$ contains a banned string, or it does not and the letter a either joins or creates a local word at the end of y_g . The former case is the situation of set \mathcal{C} , while the latter cases are the situations of sets \mathcal{A} and \mathcal{B} , respectively. Thus $\vec{E}_r = \mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$. \square

Now that we have a set of normal forms and a set of recursive edges, we show the following.

Theorem 2.6. *Let groups G_1, G_2, \dots, G_n be stackable with finite inverse-closed generating sets A_1, A_2, \dots, A_n and stacking structures $(\mathcal{N}_1, \phi_1), (\mathcal{N}_2, \phi_2), \dots, (\mathcal{N}_n, \phi_n)$ respectively. Then any graph product of the groups G_1, \dots, G_n is also stackable.*

Proof. Let $G_i = \langle A_i \rangle$ and $<_G$ be any total ordering on the finite set of groups $\{G_i\}$. Let G be a graph product of the groups $\{G_i\}$ with generating set $A := \bigcup_i A_i$. The set of normal forms, \mathcal{N} , for the graph product is introduced in Proposition 2.3 and the recursive edges are detailed in Proposition 2.5. For each $i \in \{1, \dots, n\}$, let $<_i$ be the

induced ordering on the recursive edges in the Cayley graph of G_i , as described in Definition 1.1.

Define a function $\phi : \mathcal{N} \times A \rightarrow A^*$ to be

$$\phi(y_g, a) = \begin{cases} b^{-1}ab & \text{if } a \in A_i \text{ and } \tau(y_g)i \text{ contains a banned string,} \\ & \text{where } b \text{ is the final letter of } y_g \\ \phi_{G_i}(y_v, a) & \text{otherwise, where } a \in A_i, \text{ and } \text{suf}_i(y_g) = v \end{cases}$$

From the definition of the function ϕ , we can see that ϕ satisfies $\text{im}(\phi)$ is a finite set. Also, if an edge $e_{g,a}$ with $a \in A_i$ is degenerate then the string of word types $\tau(y_g)i$ does not contain a banned string, and either $\text{suf}_i(y_g)a \in \mathcal{N}_i$ or $\text{suf}_i(y_g)$ ends with a^{-1} . In this case, the edge $e_{\text{suf}_i(y_g),a}$ is degenerate, and so $\phi(e_{g,a}) = a$. Conversely, suppose $e_{y_g,a}$ is an edge such that $\phi(y_g, a) = a$ for $a \in A_i$. Then $a = \phi_{G_i}(y_v, a)$ where $v = \text{suf}_i(y_g)$. Since ϕ_{G_i} is a stacking function, the edge $e_{v,a}$ is a degenerate edge in the Cayley graph of G_i and thus either $y_v a = y_{va}$ or $y_v a^{-1} = y_v$. But this implies that either $y_g a = y_{ga}$ or $y_g a^{-1} = y_g$ and so $e_{y_g,a}$ is a degenerate edge in the Cayley graph of G . Therefore $\phi(y_g, a) = a$ if and only if $e_{g,a} \in \vec{E}_d$.

Let \vec{E}_i be the set of all edges in the Cayley graph of G_i with respect to A_i . As $<_i$ is a strict well-founded partial ordering on the recursive edges in \vec{E}_i , we can define a map $\psi_i : \vec{E}_i \rightarrow \mathbb{N}$ as

$$\psi_i(e_{g,a}) = \text{maximum length of a decending chain } e_{g,a} >_i e' >_i e'' \dots$$

for $g \in G_i$ and $a \in A_i$. Using the maps ψ_i for each G_i we can define a map $\psi : \vec{E}_r \rightarrow \mathbb{N}^2$ by

$$\psi(e_{g,a}) = \begin{cases} (\ell(y_g), 0) & \text{if } a \in G_i \text{ and } \tau(y_g)i \text{ contains a banned string} \\ (0, \psi_i(e_{v,a})) & \text{else, where } \text{suf}_i(y_g) = v \end{cases}$$

Let $<_{\mathbb{N}^2}$ be the ordering on \mathbb{N}^2 defined by $(a, b) <_{\mathbb{N}^2} (c, d)$ whenever $a < c$ or $a = c$ and $b < d$. We define an ordering $<_{\psi}$ on \vec{E}_r by $e' <_{\psi} e$ if and only if $\psi(e') <_{\mathbb{N}^2} \psi(e)$. As $<_{\mathbb{N}^2}$ is a well-founded strict partial order, so is $<_{\psi}$.

To see that $<_{\phi}$ is a well-founded strict partial order, it suffices to show that for e' a recursive child of e , we have that $e' <_{\psi} e$. We consider the edges of \vec{E}_r in two cases. Case 1: Suppose the edge $e_{g,a}$ has associated stacking function value $\phi(y_g, a) = \phi_{G_i}(v, a)$ for $a \in A_i$ and $v = \text{su}f_i(y_g)$.

Decompose y_g as $y_g = wv$ for $w \in \mathcal{N}$ and let $\phi_{G_i}(v, a) = \alpha_1 \cdots \alpha_k$ for $\alpha_j \in A_i$; define e_j to be the child labeled by the letter α_j . By the definition of the normal forms of G , we can decompose the normal form of the group element labeling the initial vertex of e_j as wv_j for $v_j \in \mathcal{N}_i$ and w the same word appearing in the decomposition of y_g . Since e_{v_j, α_j} is a child of $e_{v, a}$, we have that $\psi_i(e_{v_j, \alpha_j}) <_{\mathbb{N}} \psi_i(e_{v, a})$ and thus $\psi(e_{wv_j, \alpha_j}) <_{\mathbb{N}^2} \psi(e_{wv, a})$. Therefore, $e_j <_{\psi} e_{g, a}$.

Case 2: Suppose the edge $e_{g,a}$ has associated stacking function value $\phi(y_g, a) = b^{-1}ab$ for $y_g \in \mathcal{N}$, $a \in A_i$ and $b \in A_j$ the final letter of y_g .

In this case, the ψ -function value of the edge $e_{g,a}$ is $\psi(e_{g,a}) = (\ell(y_g), 0)$ with $\ell(y_g) > 0$, by assumption. The edge $e_{g,a}$ has three children: $e_{g,b^{-1}}$, $e_{gb^{-1},a}$ and $e_{gb^{-1}a,b}$. Since y_g has decomposition $y_g = wb$ and our normal forms are prefix-closed, we know that $y_{wb} = y_wb$ and thus the first child is a degenerate edge. If the second child is not degenerate, then either $\tau(y_{gb^{-1}})i$ still contains a banned string, or it does not. In the former case, we have that the ψ -function value of this child is $\psi(e_{y_gb^{-1},a}) = (\ell(y_gb^{-1}), 0)$. As $\ell(y_gb^{-1}) = \ell(y_g) - 1$, we have that $\psi(e_{y_gb^{-1},a}) <_{\mathbb{N}^2} \psi(e_{g,a})$ and therefore $e_{y_gb^{-1},a} <_{\psi} e_{g,a}$. In the latter case, the string of word types $\tau(y_{gb^{-1}})i$ does not contain a banned string. As this edge is recursive, we have that for $v = \text{su}f_i(y_{gb^{-1}})$, the edge $e_{v,a}$ is a recursive edge in G_i . Then $\psi_i(e_{v,a}) = m$ for m the maximal length of a decending chain of edges from $e_{v,a}$ and so $\psi(e_{y_gb^{-1},a}) = (0, m)$. Thus $\psi(e_{y_gb^{-1},a}) <_{\mathbb{N}^2} \psi(e_{g,a})$ and

so $e_{gb^{-1},a} <_{\psi} e_{g,a}$.

Finally, let $w_j = \text{su}f_j(y_g)$ and decompose y_g as $y_g = uw_j$ for $u \in \mathcal{N}$. Recall that $a \in A_i$ and $b \in A_j$. We show that the normal form of the word $y_gb^{-1}a$ is $y_{ua}y_{w_jb^{-1}}$ and if $w_jb^{-1} = 1$, then the string of word types $\tau(y_{ua})j$ does not contain a banned string. Suppose first that in the word $y_{gb^{-1}}$ there is no local word w_i of type i which commutes with every local word to the right of it. That is, the word $y_{gb^{-1}a}$ contains the local word y_a . In this case, the word $y_{gb^{-1}a}$ is the word $y_{gb^{-1}}$ with the local word y_a inserted into the appropriate place in the word. In particular, the local words of the word $y_{gb^{-1}}$ remain in the same relative order in $y_{gb^{-1}a}$. Therefore the word w_jb^{-1} is still the final local word. If $w_jb^{-1} = 1$, then the string of word types $\tau(y_{gb^{-1}a})j$ does not contain a banned string, as $\tau(y_{gb^{-1}})j$ does not, and by assumption $i < j$.

Suppose instead that there exists a local word w_i of type i in the word $y_{gb^{-1}}$ such that $y_{gb^{-1}}$ decomposes as $y_{gb^{-1}} = \cdots w_i w_{k_1} \cdots w_{k_m} w_j b^{-1}$ where there is an edge between G_i and G_z in the graph defining the graph product for each $z = k_1, \dots, k_m$. If $y_{w_i a} \neq 1$, then the normal form of the word $y_{gb^{-1}a}$ is the normal form for $y_{gb^{-1}}$ with w_i replaced by $y_{w_i a}$. If $y_{w_i a} = 1$, then the normal form for $y_{gb^{-1}a}$ is the normal form for $y_{gb^{-1}}$ with the word w_i removed. Indeed, suppose there exist two local words v_k and v_l of types k and l , respectively, such that $y_{gb^{-1}} = \cdots v_k \cdots v_l \cdots$ but $G_l \leq_G G_k$. If the decomposition of $y_{gb^{-1}}$ has either $\cdots w_i \cdots v_k \cdots v_l \cdots$ or $\cdots v_k \cdots v_l \cdots w_i \cdots$, then the cancellation of the word w_i in the word $y_{gb^{-1}a}$ has no effect on the words $v_k \cdots v_l$. Assume then that the decomposition of the word $y_{gb^{-1}}$ is of the form $\cdots v_k \cdots w_i \cdots v_l \cdots$. For the letter a to combine with the local word w_i , we must have that w_i commutes with the local word v_l , as well as every local word between them. However this implies that there must be some other barrier between the words v_k and v_l (as described in Definition 2.3) as the barrier w_i can be commuted past v_l and thus would no longer be a barrier. Therefore removal of the local word w_i does not change the remaining

word $y_{gb^{-1}}$. Thus the edge $e_{gb^{-1}a,b}$ is a degenerate edge.

Therefore (\mathcal{N}, ϕ) satisfies stacking property 2r of Definition 1.1 and thus is a stacking structure for G . \square

Theorem 2.7. *Let groups G_1, G_2, \dots, G_n be autostackable (respectively algorithmically stackable) with finite inverse-closed generating sets A_1, A_2, \dots, A_n , and stacking structures $(\mathcal{N}_1, \phi_1), (\mathcal{N}_2, \phi_2), \dots, (\mathcal{N}_n, \phi_n)$ respectively. Then any graph product, G , of the groups G_1, \dots, G_n is also autostackable (respectively algorithmically stackable).*

Proof. First, we show that the set of normal forms \mathcal{N} is regular (decidable) and then proceed using Lemma 1.5. We can view the normal form set for the graph product as the intersection of normal forms for a free product intersected with all words whose string of word types does not contain a banned string. Recall that provided $\text{graph}(\phi_i)$ is regular (respectively decidable), the set of normal forms \mathcal{N}_i is also regular (respectively decidable). To show that the normal forms for a free product are regular (respectively decidable), we follow the method of [8]. For each $i = 1, \dots, |V(\Gamma)|$, define a monoid homomorphism $p_i : A^* \rightarrow (A_i \cup \$)^*$ for $\$$ a letter not already in the alphabet A , by

$$p_i(a) := \begin{cases} a & \text{if } a \in A_i \\ \$ & \text{if } a \in A \setminus A_i \end{cases}$$

Then the set $\mathcal{N}_{f.p.} = \bigcap_{i=1}^n p_i^{-1}(\mathcal{N}_i(\$ \mathcal{N}_i)^*)$ is a set of normal forms for the free product of the groups G_1, \dots, G_n . Indeed, each inverse image $p_i^{-1}(\mathcal{N}_i(\$ \mathcal{N}_i)^*)$ is the collection of all words where local words of type i are in normal form. The intersection over all $i \in \{1, \dots, n\}$ yields all words in normal form in the free product. Therefore the set $\mathcal{N}_{f.p.}$ is regular (respectively decidable), when each \mathcal{N}_i is.

Recall the set I_j is defined as $I_j = \{k | v_k \text{ and } v_j \text{ are connected by an edge}\}$ and the set P_j is defined as $P_j = \{k | k \in I_j \text{ and } G_j <_G G_k\}$. To see that the set of normal forms \mathcal{N} for the graph product are regular (respectively decidable), define $L_{j,k}$ to be the set $L_{j,k} := A_k \left(\bigcup_{i \in I_j} A_i \right)^* A_j$. The set $L_{j,k}$ is similar to the set of banned strings for fixed j and k , except where banned strings use indices, the set $L_{j,k}$ uses words whose types correspond to those indices. Consider the following expression:

$$\mathcal{N} = \mathcal{N}_{f.p.} \cap \left(A^* \setminus A^* \left(\bigcup_{j=1}^n \bigcup_{k \in P_j} L_{j,k} \right) A^* \right)$$

This expression is regular (respectively decidable), as it is written using only \cap , \cup , $*$, \setminus and \cdot with regular (respectively decidable) sets. Now we split into two cases.

Case 1: Fix $a \in A_i$ and $x \in im(\phi_{G_i})$.

The set of words y for which $(y, a, x) \in graph(\phi)$ are those $y \in \mathcal{N}$ such that $(suf_i(y), a, x) \in graph(\phi_{G_i})$ and such that $\tau(y)i$ does not contain a banned string. Alter the definition of the set $L_{j,k}$ to define $M_{j,k} := A_k \left(\bigcup_{i \in I_j} A_i \right)^*$. That is, $M_{j,k}$ is the collection of prefixes of banned strings ending in a word of type j and beginning with a word of type k . Unioning over each $k \in P_j$ we define the set $M_j := \bigcup_{k \in P_j} M_{j,k}$ of all prefixes of banned strings ending with a word of type j . Finally, we can write the set $S_{a,x}$ as

$$[\mathcal{N} \setminus (A^* M_i)] \cap \left[\left(A^* \setminus (A^* a_i | a_i \in A_i) \right) \cdot \left(\rho_1(graph(\phi_{G_i}) \cap \rho_2^{-1}(\{a\}) \cap \rho_3^{-1}(\{x\})) \right) \right]$$

for ρ_i as defined in the proof of Lemma 1.5.

Case 2: Fix $a \in A_j$ and $x = b^{-1}ab$ in the image of ϕ .

The set of words $y \in \mathcal{N}$ such that $(y, a, b^{-1}ab) \in graph(\phi)$ are those in normal form which end with the letter b and have $\tau(y)j$ containing a banned string. We can

express the set $S_{a,b^{-1}ab}$ as

$$S_{a,b^{-1}ab} = \mathcal{N} \cap A^*b \cap A^*M_j$$

We have written the sets $S_{a,x}$ and the normal form set \mathcal{N} in terms of the maps ρ_i , inverse images, finite intersections and unions and concatenations of the languages $\text{graph}(\phi_{G_i})$ and a handful of regular languages. Therefore we have that $\text{graph}(\phi)$ is regular (respectively decidable) whenever each of the sets $\text{graph}(\phi_{G_i})$ are regular (respectively decidable). \square

In [18], Hermiller and Meier present another proof that \mathcal{N} is a regular set of normal forms for a graph product using finite state automata.

2.2 Group Extensions

We continue the investigation into closure properties with the extension of a group K by a group Q . Let

$$1 \rightarrow K \xrightarrow{\iota} G \xrightarrow{q} Q \rightarrow 1$$

be a short exact sequence with injection map $\iota : K \hookrightarrow G$ and projection map $q : G \twoheadrightarrow Q$. Then G is the extension of the group K by the group Q . As the sequence is exact, we have that K is a normal subgroup of G and the group Q is isomorphic to the quotient group G/K .

Let $K = \langle A|R \rangle$ and $Q = \langle B|S \rangle$ be presentations for the groups K and Q , respectively, with A and B finite inverse-closed generating sets. For each $b \in B$, let \hat{b} be an element of G such that $q(\hat{b}) = b$; let $\hat{B} = \{\hat{b} | b \in B\}$. We can choose \hat{B} to be inverse closed. Let C be the set $A \cup \hat{B}$. Then C is an inverse-closed generating set

for G . For each word $w = b_1 \cdots b_n \in B^*$, we define $\hat{w} := \hat{b}_1 \cdots \hat{b}_n$. For a set S , define $\widehat{S} := \{\hat{w} | w \in S\}$.

Lemma 2.8. *Let $K = \langle A | R \rangle$ and $Q = \langle B | S \rangle$ be groups with A and B finite inverse-closed generating sets, and normal form sets \mathcal{N}_K and \mathcal{N}_Q , respectively. Then $\mathcal{N} = \{u_g t_g | u_g \in \mathcal{N}_K \text{ and } t_g \in \widehat{\mathcal{N}_Q}\}$ is a set of normal forms for G*

Proof. First, since Q is isomorphic to G/K , $\widehat{\mathcal{N}_Q}$ is a set of coset representatives for G/K . That is $\widehat{\mathcal{N}_Q}$ is a transversal for G/K . By definition of a transversal, every element $g \in G$ has a unique representation as $u_g t_g$ with $u_g \in \mathcal{N}_K$ and $t_g \in \widehat{\mathcal{N}_Q}$. \square

We can present the extension G as

$$G = \langle C | R \cup \{\hat{s} = u_{\hat{s}} | \hat{s} \in \widehat{S}\} \cup \{\hat{b}a = u_{\hat{b},a} \hat{b} | a \in A, \hat{b} \in \widehat{B}\} \rangle \quad (2.1)$$

The words $u_{\hat{s}}$ and $u_{\hat{b},a}$ are in \mathcal{N}_K . Relations of the form $\hat{b}a = u_{\hat{b},a} \hat{b}$ are found as follows: the word $\hat{b}a$ is an element of G , and so applying Lemma 2.8 we can write it in normal form as $u_{\hat{b},a} t_{\hat{b},a}$ for $u_{\hat{b},a}$ a word in \mathcal{N}_K depending on \hat{b} and a , and $t_{\hat{b},a}$ a word in $\widehat{\mathcal{N}_Q}$. But K is a normal subgroup of G , and so the right cosets equal the left cosets. Therefore $t_{\hat{b},a} = \hat{b}$. In the case of a split extension, we also have that $u_{\hat{s}} = 1$. Recall that both A , B , R and S are all finite sets, and thus the presentation in (2.1) is a finite presentation of G .

Using these normal forms we can determine the set \vec{E}_r of recursive edges.

Lemma 2.9. *Let G be an extension of the group $K = \langle A | R \rangle$ by the group $Q = \langle B | S \rangle$ with A and B finite inverse-closed generating sets with presentation described on line (2.1). Let \mathcal{N}_K and \mathcal{N}_Q be normal forms for the groups K and Q , respectively, and let \mathcal{N} as described in Lemma 2.8 be a set of normal forms for G . Define the sets \mathcal{A} , \mathcal{B}*

and \mathcal{C} as

$$\begin{aligned}\mathcal{A} &= \{e_{g,a} | y_g = u_g \in \mathcal{N}_K, \text{ with } a \in A, \text{ and neither } u_g a \in \mathcal{N}_K \text{ nor } y_{u_g a} a^{-1} \in \mathcal{N}_K\} \\ \mathcal{B} &= \{e_{g,a} | y_g = u_g t_g \text{ with } u_g \in \mathcal{N}_K, t_g \in \widehat{\mathcal{N}}_Q \setminus \{1\}, \text{ and } a \in A\} \\ \mathcal{C} &= \{e_{g,\hat{b}} | y_g = u_g t_g \text{ with } u_g \in \mathcal{N}_K, t_g \in \widehat{\mathcal{N}}_Q, \text{ and } \hat{b} \in \hat{B}, \text{ and neither } t_g \hat{b} \in \mathcal{N}_K \\ &\quad \text{nor } y_{t_g \hat{b}} \hat{b}^{-1} \in \mathcal{N}_K\}\end{aligned}$$

Then the set of recursive edges \vec{E}_r is $\vec{E}_r = \mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$.

Proof. First, we show that each of \mathcal{A}, \mathcal{B} and \mathcal{C} is contained in \vec{E}_r . Let $e_{g,a} \in \mathcal{A}$. Then $y_g = u_g$ with $u_g \in \mathcal{N}_K$, but neither $u_g a \in \mathcal{N}_K$ nor $y_{u_g a} a^{-1} \in \mathcal{N}_K$. As y_g has no component from \hat{B}^* , $y_g a \in \mathcal{N}$ only when $u_g a \in \mathcal{N}_K$. But by assumption $u_g a \notin \mathcal{N}_K$ and so $y_g a \notin \mathcal{N}$. Similarly, as $y_{u_g a} a^{-1} \notin \mathcal{N}_K$ we know that $y_{u_g a} a^{-1}$ is not in normal form. Therefore $e_{g,a} \in \vec{E}_r$.

Let $e_{g,a} \in \mathcal{B}$ with $a \in A$. We can write y_g as $y_g = u_g t_g$ with $u_g \in \mathcal{N}_K$ and $t_g \in \widehat{\mathcal{N}}_Q \setminus \{1\}$. As concatenating y_g with a letter $a \in A$ leaves the word t_g unchanged, we know that $y_g a \neq y_{ga}$. However, we also have that y_g ends with a letter $\hat{b} \in \hat{B}$ and so $y_g a^{-1}$ cannot equal y_g . Therefore $e_{g,a} \in \vec{E}_r$.

Let $e_{g,\hat{b}} \in \mathcal{C}$. Then $y_g = u_g t_g$ with $u_g \in \mathcal{N}_K$ and $t_g \in \widehat{\mathcal{N}}_Q$ and neither $t_g \hat{b}$, nor $y_{t_g \hat{b}} \hat{b}^{-1}$ in $\widehat{\mathcal{N}}_Q$. As multiplying y_g by a generator $\hat{b} \in \hat{B}$ leaves the word u_g unchanged, we can repeat the argument in the case of set \mathcal{A} to get $e_{g,\hat{b}} \in \vec{E}_r$. Therefore $\mathcal{A} \cup \mathcal{B} \cup \mathcal{C} \subseteq \vec{E}_r$.

To see that $\vec{E}_r \subseteq \mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$, let $e_{g,x} \in \vec{E}_r$ for $x \in C$ and $g \in G$. By the definition of a recursive edge, we have that neither $y_g x = y_{gx}$ nor $y_{gx} x^{-1} = y_g$. Let $y_g = u_g t_g$ for $u_g \in \mathcal{N}_K$ and $t_g \in \widehat{\mathcal{N}}_Q$. We will consider these edges in three cases.

Suppose $t_g = 1$ and, $x \in A$. Then neither $u_g x = y_{u_g x}$ nor $y_{u_g x} x^{-1} = u_g$ by assumption. As normal forms are unique, this tells us that neither $u_g x$ nor $y_{u_g x} x^{-1}$ is in normal form, and so $e_{g,x} \in \mathcal{A}$. Next, suppose $t_g \neq 1$ and $x \in A$. Then this edge

is in the situation of set \mathcal{B} . Finally, if $x \in \hat{B}$, then $u_g t_g x \notin \mathcal{N}$ and $y_{u_g t_g x} x^{-1} \notin \mathcal{N}$. By Lemma 2.8, the normal form of the word $u_g t_g x$ is $u_g y_{t_g x}$, and so the word $u_g t_g x \in \mathcal{N}$ if and only if the word $t_g x \in \widehat{\mathcal{N}}_Q$. Similarly, we have that $y_{u_g t_g x} x^{-1} \in \mathcal{N}$ if and only if $y_{t_g x} x^{-1} \in \widehat{\mathcal{N}}_Q$. Therefore $e_{g,x} \in \mathcal{C}$ and so $\vec{E}_r = \mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$. \square

With the two previous lemmas, we have everything we need to show closure under group extensions.

Theorem 2.10. *Let $1 \rightarrow K \xrightarrow{\iota} G \xrightarrow{q} Q \rightarrow 1$ be an exact sequence for groups K , G and Q . If $K = \langle A | R \rangle$ and $Q = \langle B | S \rangle$ are stackable groups with finite inverse-closed generating sets A and B and stacking structures (\mathcal{N}_K, ϕ_K) and (\mathcal{N}_Q, ϕ_Q) respectively, then G is a stackable group.*

Proof. Let $C = A \cup \hat{B}$ where $\hat{B} = \{\hat{b} | q(\hat{b}) = b \text{ for } b \in B\}$ be a generating set for G and let G have the set of normal forms defined in Lemma 2.8. Define the stacking function, $\phi : \mathcal{N} \times A \rightarrow A^*$, to be

$$\phi(y_g, x) = \begin{cases} \phi_K(u_g, x) & \text{where } y_g = u_g \in \mathcal{N}_K, \text{ and } x \in A \\ v_{x,g} \phi_Q(\widehat{q(t_g)}, q(x)) & \text{where } y_g = u_g t_g, \text{ for } u_g \in \mathcal{N}_K, t_g \in \widehat{\mathcal{N}}_Q, \text{ with} \\ & x \in \hat{B} \\ \hat{b}^{-1} u_{\hat{b},x} \hat{b} & \text{where } y_g = u_g t_g, x \in A, u_g \in \mathcal{N}_K, t_g \in \widehat{\mathcal{N}}_Q \setminus \{1\}, \\ & \hat{b} \in \hat{B} \text{ the last letter of } t_g \end{cases}$$

where $v_{x,g}$ is defined to be the unique word $v_{x,g} := \hat{x}(\phi_Q(\widehat{q(t_g)}, q(\hat{x})))^{-1}$ in \mathcal{N}_K . Note that the three cases in the definition of ϕ are disjoint, and thus ϕ is a well-defined function. As the images of the words \hat{x} and $\phi_Q(\widehat{q(t_g)}, q(\hat{x}))$ are equal in the quotient group Q , we know such a word is in fact in the subgroup K . Since the image of ϕ_Q is finite, the collection of the words $v_{x,g}$ is finite and so the image of ϕ is finite. Figures

2.2 and 2.3 show an example of a stacking in the Cayley graph of the group Q and an example of a stacking in the Cayley graph of the group G .

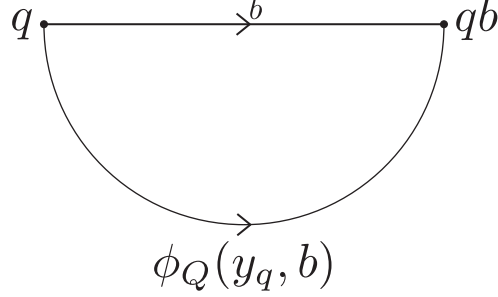


Figure 2.2: The stacking of an edge $e_{q,b}$ in the Cayley graph of Q

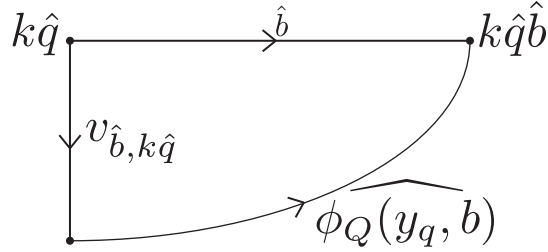


Figure 2.3: The stacking of an edge $e_{k\hat{q},\hat{b}}$ in the Cayley graph of G

Suppose $\phi(y_g, a) = a$ for $a \in A$. By the definition of the stacking function we can decompose y_g as $y_g = u_g$ for $u_g \in \mathcal{N}_K$. We can rewrite the equation $\phi(y_g, a) = a$ as $\phi_K(u_g, a) = a$. This shows that $e_{g,a}$ is a degenerate edge, when thought of as an edge in the Cayley graph of K , and so either $y_{u_g}a = y_{u_g}a$ or $y_{u_g}a^{-1} = y_{u_g}$. In either case, this also shows that $e_{g,a}$ is a degenerate edge when thought of as an edge in G . If $\phi(y_g, \hat{b}) = \hat{b}$, then we can decompose y_g as $y_g = u_g t_g$ where $u_g \in \mathcal{N}_K$ and $t_g \in \widehat{\mathcal{N}}_Q$. As the stacking function value of an edge labeled by \hat{b} is $\phi(y_g, \hat{b}) = v_{x,g} \widehat{\phi_Q(t_g, x)}$, we have that $v_{x,g} = 1$ and $\widehat{\phi_Q(q(t_g), q(\hat{b}))} = \hat{b}$. Mapping into the quotient Q , we see that

$\phi_Q(q(t_g), b) = b$ and thus the edge $e_{q(t_g), b}$ is a degenerate edge in the Cayley graph of Q . As $e_{q(t_g), b}$ is a degenerate edge in the Cayley graph of Q , we know that either $y_{q(t_g)b} = y_{q(t_g)}b$ or $y_{q(t_g)b}b^{-1} = y_{q(t_g)}$. However, the normal form for $u_g t_g b$ is $u_g y_{t_g b}$, so either $y_{u_g t_g \hat{b}} = y_{u_g t_g} \hat{b}$ or $y_{u_g t_g \hat{b}} \hat{b}^{-1} = y_{u_g t_g}$. Thus the edge $e_{g, \hat{b}}$ is a degenerate edge.

On the other hand, let $e_{g, x}$ be a degenerate edge. Then we can either decompose y_g as $y_g = u_g$ with $u_g \in \mathcal{N}_K$ and $x \in A$, or $y_g = u_g t_g$ with $u_g \in \mathcal{N}_K$, $t_g \in \widehat{\mathcal{N}}_Q$ and $x \in \hat{B}$. In the first case, we have that $\phi(y_g, x) = \phi_K(y_g, x) = x$ as $e_{g, x}$ is a degenerate edge in the Cayley graph of K . In the second case, the stacking function value for the edge $e_{g, a}$ is $\phi(y_g, x) = v_{x, g} \phi_Q(\widehat{q(t_g)}, q(\hat{b}))$ with $v_{x, g}$ in \mathcal{N}_K . As $e_{u_g t_g, x}$ is degenerate, we know that either $y_{u_g t_g} x = y_{u_g t_g} x$ or $y_{u_g t_g} x x^{-1} = y_{u_g t_g}$. Since concatenating the word $u_g t_g \in \mathcal{N}$ by a letter $x \in \hat{B}$ leaves u_g unchanged, we get that $e_{q(t_g), q(x)}$ is a degenerate edge in the Cayley graph of Q . Thus $\phi_Q(q(t_g), q(x)) = q(x)$, and so $\phi(u_g t_g, x) = v_{x, g} x$. Finally, recall that $v_{x, g} := x \phi_Q(\widehat{q(t_g)}, q(x))^{-1}$. Thus in this case, $v_{x, g} = 1$ and so $\phi(y_g x) = x$. Therefore $\phi(y_g, x) = x$ if and only if the edge $e_{g, x}$ is degenerate.

As K is a stackable group, we can define a function ψ_K on the recursive edges of the Cayley graph of K by

$$\psi_K(e_{k, a}) = \text{maximal length of a decending chain } e_{k, a} >_{\phi_K} e' >_{\phi_K} e'' \dots$$

for an edge $e_{k, a}$ with initial vertex labeled by $k \in K$ and edge labeled $a \in A$. We can define the function ψ_Q on the recursive edges in the Cayley graph of Q in an analogous fashion.

Define a function $\psi : \vec{E}_r \rightarrow \mathbb{N}^2$ on recursive edges in the Cayley graph of G by

$$\psi(e_{g,x}) = \begin{cases} (1, \psi_K(e_{g,x})) & \text{when } g \in K \text{ and } x \in A \\ (2, \ell(t_g)) & \text{where } y_g = u_g t_g \text{ for } u_g \in \mathcal{N}_K, t_g \in \widehat{\mathcal{N}}_Q \setminus \{1\} \\ & \text{and } x \in A \\ (3, \psi_Q(e_{q(t_g), q(x)})) & \text{where } y_g = u_g t_g \text{ for } u_g \in \mathcal{N}_K, t_g \in \widehat{\mathcal{N}}_Q \\ & \text{and } x \in \hat{B} \end{cases}$$

where $\ell(t_g)$ is the number of letters in the normal form word t_g . Define an order $<_{\mathbb{N}^2}$ on \mathbb{N}^2 by $(a, b, c) < (a', b', c')$ if $a < a'$ or $a = a'$ and $b < b'$ or $a = a'$, $b = b'$ and $c < c'$. Define an order $<_{\psi}$ on recursive edges by $e' <_{\psi} e$ if and only if $\psi(e') <_{\mathbb{N}^2} \psi(e)$. As $<_{\mathbb{N}^2}$ is a strict well-founded partial order on \mathbb{N}^2 , the ordering $<_{\psi}$ is a well-founded strict partial order on the set of recursive edges of G . We show that for e an edge in the Cayley graph of G if e' is a recursive child of e , then $e' <_{\psi} e$.

Case 1: Let $e_{g,x}$ be a recursive edge with $y_g = u_g$ for $u_g \in \mathcal{N}_K$ and $x \in A$.

The stacking function value for this edge is $\phi_K(u_g, x) = a_1 \cdots a_j$ for $a_i \in A$. Each child of $e_{g,x}$ is of the form $e_i := e_{ga_1 \cdots a_{i-1}, a_i}$. However, $y_{ga_1 \cdots a_{i-1}}$ is a word in A^* , and so the ψ function value for each child e_i is $\psi(e_i) = (1, \psi_K(e_{ga_1 \cdots a_{i-1}}, a_i))$. But each child e_i is a child of the edge $e_{g,x}$ as an edge in the Cayley graph of K and so $\psi_K(e') < \psi_K(e)$. Therefore $\psi(e') <_{\mathbb{N}^2} \psi(e)$ and so $e' <_{\psi} e$.

Case 2: Let $e_{g,x}$ be a recursive edge with $x \in A$ and $y_g = u_g t_g$ where $u_g \in \mathcal{N}_K$ and $1 \neq t_g \in \widehat{\mathcal{N}}_Q$.

The stacking function value for the pair (y_g, x) is the word $\hat{b}^{-1} u_{\hat{b},x} \hat{b}$, where the word t_g ends with the letter \hat{b} , and the relation $\hat{b}x = u_{\hat{b},x} \hat{b}$ is in the relation set of the presentation of G . Let $u_{\hat{b},x} = a_1 \cdots a_j$ for $a_i \in A$. The first child $e_{g,\hat{b}^{-1}}$ is degenerate, as the word y_g ends in \hat{b} . Let $e_i := e_{ga_1 \cdots a_{i-1}, a_i}$ be a child of $e_{g,x}$ labeled by $a_i \in A$.

By the definition of the normal forms for G , the normal form of the group element labeling the initial vertex of the child e_i has a decomposition into the form $u'_g t'_g$ where t'_g has length $\ell(t_g) - 1$. Recall that in this case, $\psi(e_{g,x}) = (2, \ell(t_g))$. If $\ell(t_g) = 1$, then the group elements labeling the initial vertices of e_1, \dots, e_j lie in the group K and thus $\psi(e_i) = (1, \psi_K(e_i))$ and therefore $\psi(e_i) <_{\mathbb{N}^2} \psi(e_{g,x}) = (2, \ell(t_g))$. Otherwise, the ψ function value for e_i is $\psi(e_i)$ and thus we also have $\psi(e_i) <_{\mathbb{N}^2} \psi(e_{g,x})$ for each child e_1, \dots, e_j of $e_{g,x}$. Therefore $e_i <_{\psi} e_{g,x}$ for each $i = 1, \dots, j$. Finally, as $y_{t_g \hat{b}^{-1} \hat{b}} = t_g$, we know that the child of $e_{g,x}$ labeled by \hat{b} is degenerate.

Case 3: Let $e_{g,x}$ be a recursive edge with $x \in \hat{B}$ and $y_g = u_g t_g$ where $u_g \in \mathcal{N}_K$ and $t_g \in \widehat{\mathcal{N}}_Q$.

The stacking function value of the pair (y_g, x) is $\phi(y_g, x) = v_{x,g} \phi_Q(\widehat{q(t_g)}, q(x))$ and the ψ -function value for the edge $e_{g,x}$ is $\psi(e_{g,x}) = (3, \psi_Q(e_{q(t_g), q(x)}))$. The children of the edge $e_{g,x}$ fall into two categories: those with labels of $a_i \in A$ and those with labels of $\hat{b}_i \in \hat{B}$. Let m be the length of the word $\phi(y_g, x)$. Then let e_1, \dots, e_j be the children with labels in A , and let e_{j+1}, \dots, e_m be the children with edge labels in \hat{B} . For children e_1 up to e_j , the ψ -function value for these edges are of the form $(1, m)$ where $m \in \mathbb{N}$. Thus $\psi(e_i) <_{\mathbb{N}^2} \psi(e_{g,x})$ for $i = 1, \dots, j$. For $k > j$, let the edge e_k have edge label $\hat{b}_k \in \hat{B}$ for $k = j+1, \dots, m$. By the definition of the normal forms for G , the normal form of the group element labeling the initial vertex of the edge e_{j+1} is $u'_g t_g$ for $u'_g \in \mathcal{N}_K$. However, the edge $e_{q(t_g), q(\hat{b}_{j+1})}$ is a child of $e_{q(t_g), q(x)}$ as an edge in the Cayley graph of Q by definition of the stacking function. But this means that $\psi_Q(e_{q(t_g), q(\hat{b}_{j+1})}) < \psi_Q(e_{q(t_g), q(x)})$ and so $\psi(e_{j+1}) <_{\mathbb{N}^2} \psi(e_{g,x})$. Similarly, we have $\psi(e_k) <_{\mathbb{N}^2} \psi(e_{g,x})$. Therefore if e' is a child of an edge $e_{g,x}$ we have that $e' <_{\psi} e_{g,x}$. \square

Notice that the language of normal forms for the group G can be written as

$$\mathcal{N} = \rho_1(\text{graph}(\phi_K)) \cdot \rho_1(\widehat{\text{graph}(\phi_Q)})$$

where ρ_1 is the map which projects onto the first coordinate and maps the symbol $\$$ to 1, and $\hat{\cdot} : B^* \rightarrow C^*$ is the homomorphism defined by $\hat{w} = \hat{b}_1 \cdots \hat{b}_n$ where $w = b_1 \cdots b_n$ is a word in B^* . Then by previous arguments, \mathcal{N} is a regular (respectively decidable) language, provided that the languages $graph(\phi_K)$ and $graph(\phi_Q)$ are regular (respectively decidable) languages.

Theorem 2.11. *Let $1 \rightarrow K \xrightarrow{\iota} G \xrightarrow{q} Q \rightarrow 1$ be an exact sequence for groups K , G and Q . If $K = \langle A|R \rangle$ and $Q = \langle B|S \rangle$ are autostackable (respectively algorithmically stackable) with A and B finite inverse-closed generating sets and stacking structures (\mathcal{N}_K, ϕ_K) and (\mathcal{N}_Q, ϕ_Q) respectively, then $G = \langle A \cup \hat{B} \rangle$ is autostackable (respectively algorithmically stackable).*

Proof. Since G was shown already to be stackable, we need only to prove that $graph(\phi)$ is a regular (respectively decidable) language. To show this, we will use Lemma 1.5 and break into the following cases:

1. $S_{a,x}$ where $a \in A$ and $x \in im(\phi_K)$
2. $S_{\hat{b},z}$ where $\hat{b} \in \hat{B}$ and $z = v_{x,g} \phi_Q(\widehat{q(t_g), q(\hat{b})})$,
3. $S_{a,w}$ where $a \in A$ and $w = \hat{b}^{-1} k_{\hat{b},a} \hat{b}$.

Case 1: Let $S_{a,x} = \{y | (y, a, x) \in graph(\phi)\}$ with $a \in A$ and $x \in im(\phi_K)$.

According to the stacking function ϕ the only $y \in \mathcal{N}$ which belong in the set $S_{a,x}$ are those $y \in \mathcal{N}_K$ for which $\phi_K(y, a) = x$. We can express this set as

$$S_{a,x} = \rho_1 \left(graph(\phi_K) \cap \rho_2^{-1}(\{a\}) \cap \rho_3^{-1}(\{x\}) \right).$$

for ρ_i as defined in the proof of Lemma 1.5. That is, $S_{a,x}$ is the collection of all $y \in \mathcal{N}_K$ which have $(y, a, x) \in graph(\phi_K)$.

Case 2: Let $S_{\hat{b},z} = \{y | (y, \hat{b}, z) \in \text{graph}(\phi)\}$ with $\hat{b} \in \hat{B}$ and $z = v_{\hat{b},g} \widehat{\phi_Q(q(t_g), b)}$

Let $P_{b,\phi_Q(q(t_g),b)} = \{y | (y, b, \phi_Q(q(t_g), b)) \in \text{graph}(\phi_Q)\}$. As in case 1, we can write this set as

$$P_{b,\phi_Q(q(t_g),b)} = \rho_1 \left(\text{graph}(\phi_Q) \cap \rho_2^{-1}(\{b\}) \cap \rho_3^{-1}(\{\phi_Q(q(t_g), b)\}) \right)$$

The collection of words in $S_{\hat{b},z}$ is a concatenation of normal forms in \mathcal{N}_K with the image of the words in $P_{b,\phi_Q(q(t_g),b)}$ under the map $\hat{\cdot} : B^* \rightarrow \hat{B}^*$. That is,

$$S_{\hat{b},z} = \mathcal{N}_K \cdot \widehat{P_{b,\phi_Q(q(t_g),b)}}.$$

Case 3: Let $S_{a,w} = \{y | (y, a, w) \in \text{graph}(\phi)\}$ with $a \in A$ and $w = \hat{b}^{-1} u_{\hat{b},a} \hat{b}$, for $\hat{b} \in \hat{B}$.

The collection of words in the language $S_{a,w}$ are those words in normal form in \mathcal{N} which end with the letter \hat{b} . We can express this language as

$$S_{a,w} = \mathcal{N} \cap (C^* \hat{b}).$$

Finally, in each of the previous three cases, we have a set S written in terms of $\text{graph}(\phi_K)$ and $\text{graph}(\phi_Q)$ using the maps ρ_i , inverse images, concatenations, finite unions and finite intersections, in addition to the use of a handful of sets which are written as regular expressions themselves. Therefore we have that G is autostackable (respectively algorithmically stackable) provided that K and Q are. \square

Recall that a polycyclic group is a group which has a series of subgroups

$$1 = G_0 \triangleleft G_1 \triangleleft \cdots \triangleleft G_n = G$$

such that each quotient G_{i+1}/G_i is cyclic. It is straightforward to show that all cyclic

groups are autostackable, and combining this with the previous result, we get the following corollary.

Corollary 2.12. *Let G be a polycyclic group. Then G is autostackable.*

In 1971, C.F. Miller III showed that there exists a split short exact sequence $1 \rightarrow F \xrightarrow{\iota} G \xrightarrow{q} T \rightarrow 1$ where F and T are finitely generated free groups such that G does not have solvable conjugacy problem ([24, Chapter III, Theorem 9]). Combining this result with Theorem 2.11, we get the following corollary.

Corollary 2.13. *There exists an autostackable group which does not have solvable conjugacy problem.*

2.3 Finite Index Supergroups

This section addresses closure of the stackable, algorithmically stackable and autostackable properties under finite index supergroups. That is, if a group G is stackable (respectively autostackable, algorithmically stackable) and G is a finite index subgroup of a group H , is H stackable (respectively autostackable, algorithmically stackable)? In contrast to the general group extension defined by a short exact sequence in Section 2.2, the construction here requires only a group G be a finite index subgroup of a group H , but not necessarily a normal subgroup. In the notation of Section 2.2, we are requiring the quotient G/K to be finite, but not requiring the group K to be normal in G .

The following lemma defines a set of normal forms for a finite index supergroup H given normal forms for the subgroup G .

Lemma 2.14. *Let $G = \langle A | R \rangle$ for A a finite inverse-closed generating set. Then if H is a finite index supergroup of G with finite transversal T representing a set of right*

coset representatives, the following is a set of normal forms for H :

$$\mathcal{N} = \{wt | w \in \mathcal{N}_G, t \in T\}$$

Proof. The proof of this fact follows from T being a set of right coset representatives for H/G . \square

Let $G = \langle A | R \rangle$ be a group with finite inverse-closed generating set A and let H be a finite index supergroup of G with finite right transversal T , with $1 \in T$. Let $V = T \setminus \{1\}$ and let $U = V \cup V^{-1}$. Then

$$H = \langle A, U | R \cup \{x = w_x \tilde{t}_x \text{ for all } x \in A \cup U\} \cup \{xy = w_{x,y} \tilde{t}_{x,y} \text{ for all } x, y \in A \cup U\} \rangle \quad (2.2)$$

is a presentation of H . The relations of the form $xy = w_{x,y} \tilde{t}_{x,y}$ and $x = w_x \tilde{t}_x$ for $x, y \in A \cup U$ represent the unique way to write the elements xy and x in normal form in H , respectively. Some relations in this presentation are redundant. For example, if t is an element of the transversal T , then the relation $t = w_t \tilde{t}_t$ reads $t = 1 \cdot t$, as the word t is already in its normal form. Regardless of redundant relators, this presentation is finite whenever G is finitely presented.

From this presentation, and the normal forms described in Lemma 2.14, we can describe the set of recursive edges in H .

Lemma 2.15. *Let H be a finite index supergroup of a group $G = \langle A | R \rangle$ with finite right transversal T and with generating set as described in Equation 2.2. Let \mathcal{N} be the set of normal forms for H defined in Lemma 2.14. Define the sets \mathcal{A} , \mathcal{B} and \mathcal{C} as*

$$\mathcal{A} = \{e_{g,a} | g \in G \text{ and } a \in A, \text{ where } y_g a, y_g a^{-1} \notin \mathcal{N}_G\}$$

$$\mathcal{B} = \{e_{g,s} | g \in G \text{ and } s \in U \setminus T\}$$

$$\mathcal{C} = \{e_{gt,x} | g \in G, t \in V \text{ and } x \in A \cup U \text{ where } x \neq t^{-1}\}$$

Then the set of recursive edges in the Cayley graph of H is $\vec{E}_r = \mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$.

Proof. We first show $\mathcal{A} \cup \mathcal{B} \cup \mathcal{C} \subset \vec{E}_r$. Let $e_{g,a} \in \mathcal{A}$ for $g \in G$ and $a \in A$. Then by assumption neither $y_g a \in \mathcal{N}_G$ nor is $y_g a^{-1} \in \mathcal{N}_G$. However, by the definition of \mathcal{N} , for $y \in A^*$, we have the word $y \in \mathcal{N}_G$ if and only if $y \in \mathcal{N}$. Thus $e_{g,a} \in \vec{E}_r$. Let $e_{g,s} \in \mathcal{B}$ for $g \in G$ and $s \in U \setminus T$. Since any word containing a letter $s \in U \setminus T$ is never in normal form, we have that $y_g s \notin \mathcal{N}$ and as $y_g \in A^*$, the word y_g does not end with the letter s^{-1} . Therefore $e_{g,s} \in \vec{E}_r$. Finally, let $e_{gt,x} \in \mathcal{C}$ for $g \in G$, $t \in V$ and $x \in A \cup U$. By the definition of the normal forms for H , an edge of the form $e_{gt,x}$ with $t \neq 1$ is recursive, so long as $x \neq t^{-1}$. But this is exactly the assumption for an edge in the set \mathcal{C} and so $e_{gt,x}$ is in \vec{E}_r . Thus $\mathcal{A} \cup \mathcal{B} \cup \mathcal{C} \subset \vec{E}_r$.

For the other containment, let $e_{gt,x} \in \vec{E}_r$ for $g \in G$, $t \in T$ and $x \in A \cup U$. By the definition of \vec{E}_r , we know that neither $y_{gt}x$ nor $y_{gt}x^{-1}$ are in the normal form set \mathcal{N} . If $t = 1$, then $y_g x \notin \mathcal{N}$ implies that $x \in A \cup U \setminus T$, as $y_g t$ is in normal form for any $t \in T$. If $x \in A$, then $y_g x \notin \mathcal{N}$ is equivalent to $y_g x \notin \mathcal{N}_G$, which is (half) of the situation of the set \mathcal{A} . The other half follows by the same argument from the assumption that $y_{gx}x^{-1} \notin \mathcal{N}$. If $x \in U \setminus T$, then the edge $e_{g,x}$ is the situation of the set \mathcal{B} . Finally, if $t \neq 1$, then $y_{gt}x, y_{gt}x^{-1} \notin \mathcal{N}$ implies that $x \neq t^{-1}$, which is the situation of set \mathcal{C} . Thus $\vec{E}_r = \mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$. \square

Theorem 2.16. *Let G be a stackable group with finite inverse-closed generating set A and stacking structure (\mathcal{N}_G, ϕ_G) . If G is a finite index subgroup of a group H , then H is also stackable.*

Proof. Let $G = \langle A | R \rangle$ be a stackable group, and let G be a finite index subgroup in

H with finite right transversal T . Assume that $1 \in T$. We can define the stacking function $\phi : \mathcal{N} \times (A \cup U) \rightarrow (A \cup U)^*$ by

$$\phi(y_h, x) = \begin{cases} \phi_G(y_h, x) & \text{if } h \in G \text{ and } x \in A \\ w_x \tilde{t}_x & \text{if } h \in G \text{ and } x \in U \\ t^{-1} w_{t,x} \tilde{t}_{t,x} & \text{if } h = gt \text{ for } g \in G, t \in T \setminus \{1\} \text{ and } x \in A \cup U \end{cases}$$

From the presentation given for H , we can see that $\phi(y_h, x) =_H x$ for every pair (y_h, x) . Next we show that $\phi(y_h, x) = x$ only when $e_{h,x} \in \vec{E}_d$. Let $\phi(y_h, x) = x$. According to the stacking function, there are three situations in which this could arise: $h \in G$, and $x \in A$, $h \in G$, and $x \in U$ or $h = gt$ for $g \in G, t \in V$ and $x \in A \cup U$. In the first case, $\phi(y_h, x) = \phi_G(y_h, x)$ and as ϕ_G is a stacking function, we know that $\phi_G(y_h, x) = x$ if and only if $e_{h,x}$ is a degenerate edge in G . However, this would also give $e_{h,x}$ is a degenerate edge in H . In the second case, if $h \in G$ and $x \in U$, then $\phi(y_h, x) = x$ implies that $w_x \tilde{t}_x = x$. However, for $x \in U$, this is only the case when $x \in T$, in which case the edge $e_{y_h, x}$ is degenerate. Finally, let $y_h = gt$ for $g \in G, t \in V$ and $x \in A \cup U$. Then $\phi(y_h, x) = t^{-1} w_{t,x} \tilde{t}_{t,x} = x$. For this equality to hold, we must have $x = t^{-1}$ and $w_{t,x} \tilde{t}_{t,x} = 1$. Since $h = gt$, this implies the edge $e_{h, t^{-1}}$ is degenerate. Therefore if $\phi(y_h, x) = x$, the edge $e_{h,x}$ is degenerate.

Conversely, let $e_{h,x}$ be a degenerate edge. By definition, either $y_h x = y_{hx}$ or $y_{hx} x^{-1} = y_h$. By the makeup of the normal forms, for one of these to hold true, we have one of the following situations also holds true: $h \in G, x \in A$ and $\phi_G(y_h, x) = x$; $h \in G$ and $x \in T$; or $h = gt$, with $g \in G$ and $t \in V$ with $x = t^{-1}$. It is straightforward to check that $\phi(y_h, x) = x$ in all of these cases.

Finally, we can also observe, since A and U are both finite sets and G is stackable, that $\text{im}(\phi)$ is a finite set. It remains to prove, then, that the function $\phi(y_g, a)$ gives rise to a strict well-founded partial ordering on recursive edges. Define a function ψ_G

on the edges in the Cayley graph of G by

$$\psi_G(e_{g,a}) = \text{maximal length of a descending chain } e_{g,a} >_{\phi_G} e' >_{\phi_G} e'' \dots$$

for $g \in G$ and $a \in A$. As G is stackable, this function is finite valued. Define a function $\psi : \vec{E}_r \rightarrow \mathbb{N}^2$ on the edges in the Cayley graph of H by

$$\psi(e_{h,x}) = \begin{cases} (0, \psi_G(e_{g,x})) & \text{when } h \in G, \text{ and } x \in A \\ (1, 0) & \text{when } h \in G, \text{ and } x \in U \\ (1, 1) & \text{when } h = gt, \text{ for } g \in G, t \in V \text{ and } x \in A \cup U \end{cases}$$

Define an order on \mathbb{N}^2 by $(a, b) < (c, d)$ if $a < c$ or $a = c$ and $b < d$. Define an order $<_\psi$ on recursive edges by $e' <_\psi e$ if and only if $\psi(e') <_{\mathbb{N}^2} \psi(e)$. As $<_{\mathbb{N}^2}$ is a strict well-founded order on \mathbb{N}^2 , the order $<_\psi$ is a strict well-founded order on \vec{E}_r . We show that for an edge $e \in \vec{E}_r$, if e' is a recursive child of e , then $e' <_\psi e$.

Case 1: Let $e_{g,x}$ be a recursive edge with $g \in G$ and $x \in A$.

The stacking function value for the edge $e_{g,x}$ is $\phi(y_g, x) = \phi_G(y_h, x)$. By design, the children of $e_{g,x}$ are also children of $e_{g,x}$ when thought of as an edge in the Cayley graph of G . That is, for a child e' of the edge $e_{g,x}$, we have that $\psi_G(e') <_{\mathbb{N}^2} \psi_G(e_{g,x})$. Thus $e' <_\psi e_{g,x}$.

Case 2: Let $e_{g,x}$ be a recursive edge with $g \in G$ and $x \in U$.

As this edge is recursive, $x \in U \setminus V$. The stacking function value for $e_{g,x}$ is $\phi(y_g, x) = w_x \tilde{t}_x$ and the ψ -function value is $\psi(e_{g,x}) = (1, 0)$. Let $w_x = a_1 \cdots a_j$ for $a_i \in A$. Then $e_{g,x}$ has $j + 1$ children: $e_i := e_{ga_1 \cdots a_{i-1}, a_i}$ for $i = 1, \dots, j$ and $e_{j+1} := e_{ga_1 \cdots a_j, \tilde{t}_x}$. Since the element $ga_1 a_2 \cdots a_{i-1}$ remains in the group G , the ψ function value for e_i where $1 \leq i \leq j$ is $\psi(e_i) = (0, \psi_G(e_i))$. Finally, as the element $ga_1 \cdots a_j$ is in G , the edge e_{j+1} , labeled by \tilde{t}_x , is degenerate. Therefore for every

recursive child e_i of $e_{g,x}$, we have $\psi(e_i) <_{\mathbb{N}^2} \psi(e_{g,x})$ and so $e_i <_\psi e_{g,x}$.

Case 3: Let $e_{h,x}$ be a recursive edge with $h = gt$ for $g \in G$ and $t \in V$ and $x \in A \cup U$.

The stacking function value for the edge $e_{h,x}$ in this case is $\phi(y_h, x) = t^{-1}w_{t,x}\tilde{t}_{t,x}$, where the word y_h ends with the letter $t \in V$. This edge has three types of children: the first child $e_{h,t^{-1}}$, the last child $e_{gw_{t,x},\tilde{t}_{t,x}}$ and the middle children, each labeled by a generator $a_i \in A$. The first and last children are both degenerate. The remaining children have edge labels in the set A , and so these edges have ψ function values of the form $(0, k)$ for $k \in \mathbb{N}_0$. However, as the ψ function value for the edge $e_{h,x}$ is $\psi(e_{h,x}) = (1, 1)$, we have $e_{h,x}, e' <_\psi e_{h,x}$ for every recursive child of $e_{h,x}$.

Therefore the ordering $<_\phi$ on recursive edges in the Cayley graph of H is a strict well-founded partial order, and by Definition 1.1, H is stackable. \square

Theorem 2.17. *Let G be an autostackable (respectively algorithmically stackable) group with finite inverse-closed generating set A and stacking structure (\mathcal{N}_G, ϕ_G) . If G is a finite index subgroup of a group H , then H is also autostackable (respectively algorithmically stackable).*

Proof. Since H was already shown to be stackable, we need only show that the set $\text{graph}(\phi)$ is regular (respectively decidable) language. To show this, we use Lemma 1.5 and break into the following cases:

1. $S_{a,x}$ where $a \in A$ and $x \in \text{im}(\phi_G)$.
2. $S_{s,w}$ where $s \in U \setminus T$ and $w = w_s\tilde{t}_s$
3. $S_{x,z}$ where $x \in A \cup U$ and $z = t^{-1}w_{t,x}\tilde{t}_{t,x}$

Case 1: Let $a \in A$ and $x \in \text{im}(\phi_G)$ be fixed.

The collection of the words $y \in \mathcal{N}$ which have $(y, a, x) \in \text{graph}(\phi)$ are those which also have $(y, a, x) \in \text{graph}(\phi_G)$. Therefore we can write $S_{a,x}$ as

$$S_{a,x} = \rho_1 \left(\text{graph}(\phi_G) \cap \rho_2^{-1}(\{a\}) \cap \rho_3^{-1}(\{x\}) \right).$$

for ρ_i as defined in the proof of Lemma 1.5.

Case 2: Let $s \in U \setminus T$ and $w = w_s \tilde{t}_s$.

The set of words $y \in \mathcal{N}$ which have $(y, s, w) \in \text{graph}(\phi)$ are all words in normal form in \mathcal{N}_G . That is,

$$S_{s,w} = \rho_1 \left(\text{graph}(\phi_G) \right)$$

Case 3: Let $x \in A \cup U$ and $z = t^{-1} w_{t,x} \tilde{t}_{t,x}$.

The set of words $y \in \mathcal{N}$ which have $(y, x, z) \in \text{graph}(\phi)$ are those words in normal form which end in the letter t . We can write this set as

$$S_{x,z} = \rho_1 \left(\text{graph}(\phi_G) \right) \cdot \{t\}$$

In each case above, the desired set S is expressed as images of the maps ρ_i , inverse images, concatenations, finite unions and finite intersections of the set $\text{graph}(\phi_G)$ and a few regular languages. Therefore $\text{graph}(\phi)$ is a regular (respectively decidable) language whenever $\text{graph}(\phi_G)$ is. □

Chapter 3

Homological Properties

The purpose of this chapter is to investigate the homological properties of autostackable groups. Throughout this chapter, we will use the term *x-tail* to mean the string y at the end of a given word w , in the letters $x^{\pm 1}$. For example, the *cd*-tail of a word w refers to the word using letters from $\{c, c^{-1}, d, d^{-1}\}$ that is the largest such suffix of w . The word $a^3c^{-2}bc^2dc^{-1}$ has a *cd*-tail of c^2dc^{-1} .

Let S be the group in [27] that was shown to not be of type FP_3 . The group S has presentation

$$S = \langle a, b, c, d, s \mid [a, c] = [a, d] = [b, c] = [b, d] = 1, [s, ab^{-1}] = [s, ac^{-1}] = [s, ad^{-1}] = 1 \rangle.$$

Define $A = \{a^{\pm 1}, b^{\pm 1}, c^{\pm 1}, d^{\pm 1}, s^{\pm 1}\}$. Then $S = \langle A \rangle$. Let N be the finitely generated subgroup of S given by $N = \langle ab^{-1}, ac^{-1}, ad^{-1} \rangle$. In [11], it is shown that the group N consists of all elements $g \in S$ that can be represented by a word of zero exponent sum. It follows that N is normal in S as conjugation preserves exponent sum. They also show that S can be written as the HNN extension $S \cong (F_2 \times F_2) *_N$, where the associated HNN maps are both the inclusion maps $\varphi_1, \varphi_2 : N \hookrightarrow F_2 \times F_2$.

Let

$$G = F_2 \times F_2 = \langle a, b, c, d \mid [a, c] = [a, d] = [b, c] = [b, d] = 1 \rangle$$

and let \mathcal{N}_G be the set of normal forms for G introduced in Section 2.1. That is, $uv \in \mathcal{N}_G$ when u is a reduced word in $\{a^{\pm 1}, b^{\pm 1}\}^*$ and v is a reduced word in $\{c^{\pm 1}, d^{\pm 1}\}^*$. All relations among generators in G preserve exponent sums, and so in fact if $g \in G$ has any representation as a word of zero exponent sum, all its representations are words with zero exponent sum. Because of this, we will use a slight abuse of language and refer occasionally to the exponent sum of a group element.

Lemma 3.1. *The following is a set of normal forms for S :*

$$\mathcal{N} = \{ws^{\epsilon_1}a^{i_1}s^{\epsilon_2}a^{i_2}\dots s^{\epsilon_n}a^{i_n} \mid w \in \mathcal{N}_G, n \geq 0, \epsilon_j = \pm 1, i_j \in \mathbb{Z} \text{ and if there exists a } j \text{ with } i_j = 0, \text{ then } \epsilon_j = \epsilon_{j+1}\}$$

Proof. By the normal form theorem for HNN Extensions [23, Chapter IV, Theorem 2.1], the above set is a set of normal forms if the set $\{a^i \mid i \in \mathbb{Z}\}$ is a set of coset representatives for $N \backslash (F_2 \times F_2)$. As the maps defining the HNN extension S are both inclusion maps, we have that $\varphi_1(N) = \varphi_2(N)$ and thus we need not worry about the power ϵ of the letter s .

Observe that Na^i consists of all elements of $F_2 \times F_2$ with total exponent sum i . Indeed, if $g \in F_2 \times F_2$ has exponent sum i , then the word $y_g =_G y_g a^{-i} a^i$ has exponent sum i , but $y_g a^{-i}$ has exponent sum zero, and thus $(y_g a^{-i})a^i \in Na^i$. Conversely, if we have an element ga^i in Na^i , then this element has exponent sum i , as $g \in N$ has exponent sum 0.

As every element has exponent sum i for some $i \in \mathbb{Z}$, each element of G is in at least one coset in $N \backslash G$. However, by an observation earlier, all words representing the same group element will have the same exponent sum and so the sets Na^i are

disjoint. Thus we have that $\{a^i | i \in \mathbb{Z}\}$ is a set of coset representatives for $N \setminus (F_2 \times F_2)$ and so \mathcal{N} is a set of normal forms for S . \square

Using these normal forms, we can determine which are the recursive edges in the Cayley graph of S . Any edge $e_{h,x}$ with initial vertex labeled by $h \in S$ such that $y_h = ws^{\epsilon_1}a^{i_1} \dots s^{\epsilon_n}a^{i_n}$ for $n \neq 0$ will be recursive if and only if $x \in \{b, b^{-1}, c, c^{-1}d, d^{-1}\}$. Indeed, edges of this type are recursive; the assumption $n \neq 0$ implies that y_{hx} ends with $a^{\pm 1}$ or $s^{\pm 1}$. Conversely, an edge $e_{h,x}$ with $x = a^{\pm 1}$ or $x = s^{\pm 1}$ is degenerate in the case where $n \neq 0$. The edges $e_{h,x}$, for $x = a^{\pm 1}$ have i_n either decreasing or increasing by 1, both of which result in a word in normal form, and thus a degenerate edge. If $x = s^{\pm 1}$ then multiplication by x either removes or adds an s^ϵ , which either results in a word in normal form, or peels off an $s^{-\epsilon}$ from the end of the word. Both result in a degenerate edge.

If $n = 0$, then we can use Lemma 2.5 detailing recursive edges of a graph product to find recursive edges in $F_2 \times F_2$. In the case that the edge is of the form $e_{g,x}$ for $g \in F_2 \times F_2$ and $x \in \{s, s^{-1}\}$, we have that $y_gx = y_{gx}$ and thus the edge $e_{g,x}$ is degenerate. Recall that for an element g in $F_2 \times F_2$, an edge of the form $e_{g,x}$ is degenerate if either $x \in \{c^{\pm 1}, d^{\pm 1}\}$ or if $x \in \{a^{\pm 1}, b^{\pm 1}\}$ with $y_g \in \{a^{\pm 1}, b^{\pm 1}\}^*$ and recursive otherwise. Together, these remarks prove the following lemma.

Lemma 3.2. *Let S be Stallings' not FP_3 group with normal form set described in Lemma 3.1. Define the sets*

$$\mathcal{A} = \{e_{h,x} | y_h = ws^{\epsilon_1}a^{i_1} \dots s^{\epsilon_n}a^{i_n}, w \in \mathcal{N}_G, n > 0, x \in \{b, b^{-1}, c, c^{-1}d, d^{-1}\}\}$$

$$\mathcal{B} = \{e_{g,x} | g \in G, x \in \{a, a^{-1}, b, b^{-1}\}, y_gx, y_gx^{-1} \notin \mathcal{N}_G\}$$

Then the set of recursive edges in the Cayley graph of S is $\vec{E}_r = \mathcal{A} \cup \mathcal{B}$.

Lemma 3.3. *Stallings' group S is stackable.*

Proof. From the normal forms defined in Lemma 3.1, we can define the following stacking function $\phi : \mathcal{N} \times A \rightarrow A^*$

$$\phi(y_h, x) = \begin{cases} z^{-1}xz & \text{for } h \in G \text{ and } x \in \{a^{\pm 1}, b^{\pm 1}\} \text{ where} \\ & z \in \{c^{\pm 1}, d^{\pm 1}\} \text{ is the last letter of } y_h \\ a^{-\eta}xa^\eta & \text{for } h \in S \text{ and } x \in \{c^{\pm 1}, d^{\pm 1}\} \text{ with} \\ & y_h = ws^{\epsilon_1}a^{i_1} \dots s^{\epsilon_n}a^{i_n} \text{ where } w \in \mathcal{N}_G, \\ & n > 0 \text{ and } i_n = \eta|i_n| \neq 0 \\ c^{-\eta}xc^\eta & \text{for } h \in S \text{ and } x \in \{b, b^{-1}\} \text{ with} \\ & y_h = ws^{\epsilon_1}a^{i_1} \dots s^{\epsilon_n}a^{i_n} \text{ where } w \in \mathcal{N}_G, \\ & n > 0 \text{ and } i_n = \eta|i_n| \neq 0 \\ s^{-\epsilon_n}xa^{-\eta}x^{\epsilon_n}a^\eta & \text{for } h \in S \text{ and } x \in \{b^\eta, c^\eta, d^\eta\} \text{ with} \\ & y_h = ws^{\epsilon_1}a^{i_1} \dots a^{i_{n-1}}s^{\epsilon_n} \text{ where } n > 0 \\ & \text{and } \eta \in \{1, -1\} \\ x & \text{for } h \in S \text{ and } x \in A \text{ where } e_{h,x} \in \vec{E}_d \end{cases}$$

It follows from the definition of ϕ above that ϕ has a finite image, $\phi(y_h, x) =_G x$ for all $h \in S$ and $x \in A$ and $\phi(y_h, x) = x$ when $e_{h,x}$ is a degenerate edge. To see that $<_\phi$ is a strict well-founded partial order on the set of recursive edges, consider the following function ψ defined on recursive edges:

$$\psi(e_{h,x}) = \begin{cases} (0, 0, \ell_{cd}(y_h)) & \text{when } h \in G \text{ and } x \in A \\ (n, |i_n|, 1) & \text{when } y_h = ws^{\epsilon_1}a^{i_1} \dots s^{\epsilon_n}a^{i_n} \text{ with } w \in \mathcal{N}_G, \\ & \epsilon_j \in \{-1, 1\} \text{ and } i_j \in \mathbb{Z} \text{ for each } j = 1, \dots, n, \\ & x \in \{b, b^{-1}\} \text{ and } n > 0 \\ (n, |i_n|, 0) & \text{when } y_h = ws^{\epsilon_1}a^{i_1} \dots s^{\epsilon_n}a^{i_n} \text{ with } w \in \mathcal{N}_G, \\ & \epsilon_j \in \{-1, 1\} \text{ and } i_j \in \mathbb{Z} \text{ for each } j = 1, \dots, n, \\ & x \in \{c^{\pm 1}, d^{\pm 1}\} \text{ and } n > 0 \end{cases}$$

where $\ell_{cd}(y_h)$ denotes the length of the cd -tail of the word y_h .

Define an ordering $<_\psi$ by $e' <_\psi e$ if and only if $\psi(e') <_{\mathbb{N}^3} \psi(e)$, where $<_{\mathbb{N}^3}$ is the lexicographical ordering on \mathbb{N}^3 . The ordering $<_\psi$ is a strict well-founded ordering on recursive edges in the Cayley graph of S ; if we show that $e' <_\phi e$ implies $e' <_\psi e$ for e' a child of the edge e , then we have that $<_\phi$ is also a strict well-founded partial order on recursive edges.

Case 1: Let $e_{h,x}$ be a recursive edge with $h \in G$ and $x \in \{a^{\pm 1}, b^{\pm 1}\}$.

In this case, $\psi(e_{h,x}) = (0, 0, \ell_{cd}(y_h))$. Suppose $z \in \{c^{\pm 1}, d^{\pm 1}\}$ is the last letter of y_h . Then the edge $e_{h,x}$ has three children: $e_{h,z^{-1}}$, $e_{hz^{-1},x}$ and $e_{hz^{-1}x,z}$. The first child is degenerate, as $y_{hz^{-1}} = y_h$. The second child, $e_{hz^{-1},x}$, has a ψ -function value of $\psi(e_{hz^{-1},x}) = (0, 0, \ell_{cd}(y_{hz^{-1}}))$. However, by design, $\ell_{cd}(y_{hz^{-1}}) = \ell_{cd}(y_h) - 1$, and so $e_{hz^{-1},x} <_\psi e_{h,x}$. The third child is of the form $e_{hz^{-1}x,z}$. Notice that for the decomposition $y_h = uv$ for $u \in \{a^{\pm 1}, b^{\pm 1}\}^*$ and $v \in \{c^{\pm 1}, d^{\pm 1}\}^*$, we have that $y_{hz^{-1}x} = y_{ux}y_{hz^{-1}}$. As $y_{uz^{-1}z} = u$, we have the third child is a degenerate edge.

Case 2: Let $e_{h,x}$ be a recursive edge for $h \in S$ and $x \in \{c^{\pm 1}, d^{\pm 1}\}$ with $y_h = ws^{\epsilon_1}a^{i_1} \dots s^{\epsilon_n}a^{i_n}$ where $w \in \mathcal{N}_G$, $n > 0$, $\epsilon_j \in \{-1, 1\}$, $i_j \in \mathbb{Z}$ for each $j = 1, \dots, n$ with $i_n \neq 0$.

The stacking function for this edge is $\phi(e_{h,x}) = z^{-1}xz$ for $z \in \{a, a^{-1}\}$ the final letter of y_h and the ψ -function value is $\psi(e_{h,x}) = (n, |i_n|, 0)$. Similar to case 1, the first child $e_{h,z^{-1}}$ is a degenerate edge. The second child, $e_{hz^{-1},x}$ has ψ function value of $\psi(e_{hz^{-1},x}) = (n, |i_n| - 1, 0)$. which is less than $\psi(e_{h,x}) = (n, |i_n|, 0)$. Thus $e_{hz^{-1},x} <_\psi e_{h,x}$. The third child is of the form $e_{hz^{-1}x,z}$. The normal form of the word $hz^{-1}x$ still contains n appearances of s and s^{-1} and therefore the edge $e_{hz^{-1}x,z}$, which is labeled by $z \in \{a, a^{-1}\}$, is a degenerate edge.

Case 3: Let $e_{h,x}$ be a recursive edge for a group element $h \in S$ and a generator $x \in \{b, b^{-1}\}$ with $y_h = ws^{\epsilon_1}a^{i_1} \dots s^{\epsilon_n}a^{i_n}$ where $w \in \mathcal{N}_G$, $n > 0$, $\epsilon_j \in \{-1, 1\}$, $i_j \in \mathbb{Z}$

for each $j = 1, \dots, n$ with $i_n \neq 0$.

The edge $e_{h,x}$ in this case has stacking function value $\phi(y_h, x) = c^{-\eta} x c^\eta$, where $i_n = \eta|i_n|$ and ψ -function value $\psi(e_{h,x}) = (n, |i_n|, 1)$. This edge has three children: $e_{h,c^{-\eta}}$, $e_{hc^{-\eta},x}$ and $e_{hc^{-\eta}x,c^\eta}$. Unlike the previous case, each of these children is a recursive edge. The first child, $e_{h,c^{-\eta}}$, has a ψ -function value of $\psi(e_{h,c^{-\eta}}) = (n, |i_n|, 0)$ which is less than $\psi(e_{h,x}) = (n, |i_n|, 1)$. Thus $e_{h,c^{-\eta}} <_\psi e_{h,x}$. Using Lemma 3.1, observe that $y_{hc^{-\eta}} = y_{wc^{-\eta}a^\eta} s^{\epsilon_1} a^{i_1} \dots s^{\epsilon_n} a^{i_n-\eta}$. Therefore the ψ -function value for the edge $e_{hc^{-\eta},x}$ is $\psi(e_{hc^{-\eta},x}) = (n, |i_n - \eta|, 1)$. But we have the inequality $|i_n - \eta| < |i_n|$ and so $\psi(e_{hc^{-\eta},x}) <_{\mathbb{N}^3} \psi(e_{h,x})$. Thus $e_{hc^{-\eta},x} <_\psi e_{h,x}$.

Finally, we consider the third child $e_{hc^{-\eta}x,c^\eta}$. The ψ -function value for this edge is dependent on several factors. Specifically, we get different values depending on whether $x = b$ or $x = b^{-1}$. and depending on whether $\eta = 1$ or $\eta = -1$. Let $x = b$. The case where $x = b^{-1}$ follows a similar argument. If $\eta = 1$, then the normal form of the word $hc^{-\eta}x$ is of the form $w' s^{\epsilon_1} a^{i'_1} \dots s^{\epsilon_n} a^{i_n}$ where $w' \in \mathcal{N}_G$ and $i'_j \in \mathbb{Z}$. It is possible that some $i'_j = 0$ which could cause s^{ϵ_j-1} to cancel with s^{ϵ_j} . However, in this case, we would have $\psi(e_{hc^{-\eta}x}) <_{\mathbb{N}^3} \psi(e_{h,x})$ based on a first coordinate comparison. If this is not the case, the ψ -function value of $e_{hc^{-\eta}x,c^\eta}$ is $\psi(e_{hc^{-\eta}x,c^\eta}) = (n, i_n, 0)$ which is still less than $\psi(e_{h,x}) = (n, i_n, 1)$ in the ordering $<_{\mathbb{N}^3}$. Suppose now that $\eta = -1$. Then the normal form of the word $hc^{-\eta}x$ is of the form $w'' s^{\epsilon_1} a^{i''_1} \dots s^{\epsilon_n} a^{i_n+2}$, where $w'' \in \mathcal{N}_G$ and $i''_j \in \mathbb{Z}$. Since $\eta = -1$, we know that $i_n < 0$ and so $|i_n + 2| < |i_n|$ and therefore $\psi(e_{hc^{-\eta}x,c^\eta}) <_{\mathbb{N}^3} \psi(e_{h,x})$. Thus in all situations, $e_{hc^{-\eta}x,c^\eta} <_\psi e_{h,x}$.

Case 4: Let $e_{h,x}$ be a recursive edge for $h \in S$ and $x \in \{b^\eta, c^\eta, d^\eta\}$ for $\eta \in \{1, -1\}$ with $y_h = w s^{\epsilon_1} a^{i_1} \dots s^{\epsilon_n}$ where $w \in \mathcal{N}_G$, $n > 0$, $\epsilon_j \in \{-1, 1\}$, $i_j \in \mathbb{Z}$ for each $j = 1, \dots, n-1$ and $i_n = \eta|i_n|$.

According to the stacking function ϕ , the edge $e_{h,x}$ has five children: $e_{h,s^{-\epsilon_n}}$, $e_{hs^{-\epsilon_n},x}$, $e_{hs^{-\epsilon_n}x,a^{-\eta}}$, $e_{hs^{-\epsilon_n}xa^{-\eta},s^{\epsilon_n}}$ and $e_{hs^{-\epsilon_n}xa^{-\eta}s^{\epsilon_n},a^\eta}$. Children one and four are both

degenerate, as every edge labeled by the letter s or s^{-1} is degenerate. Child five is also degenerate, as an edge labeled by the letter a or a^{-1} with initial vertex of the form $y = ws^{\epsilon_1} \cdots s^{\epsilon_n}$ for $n > 0$ is degenerate. Recall that in this case, $\psi(e_{h,x}) = (n, |i_n|, m)$ for $m \in \{0, 1\}$ depends on the value of x . If $n > 1$, then the ψ -function value for the second child, $e_{hs^{-\epsilon_n},x}$ is $\psi(e_{hs^{-\epsilon_n},x}) = (n, |i_n| - 1, m)$ and thus $\psi(e_{hs^{-\epsilon_n},x}) <_{\mathbb{N}^3} \psi(e_{h,x})$. When $n > 1$, the argument above showing the fifth child is degenerate suffices to show the third child is degenerate as well. If $n = 1$, then the ψ -function value for both the second and third children is of the form $(0, 0, \ell_{cd}(y_h))$. In this case, we also have both $\psi(e_{hs^{-\epsilon_n},x}) <_{\mathbb{N}^3} \psi(e_{h,x})$ and $\psi(e_{hs^{-\epsilon_n}x,a^{-n}}) <_{\mathbb{N}^3} \psi(e_{h,x})$. Thus for e' a child of the edge $e_{h,x}$ we have $e' <_{\psi} e$.

The four cases above show that for e' a child of an edge e , we have that $e' <_{\psi} e$. \square

We can now begin the proof of Theorem 3.4.

Theorem 3.4. *There exists an autostackable group that does not have type FP_3 .*

Proof. We use Lemma 3.3 to prove that S is autostackable. It is sufficient, then, to show that the set $\text{graph}(\phi)$ is regular. To do this, we use Lemma 1.5. The stacking function for S is relatively straightforward. When the second and third entries are fixed, the completion of each tuple in $\text{graph}(\phi)$ is simply a word in normal form ending with a particular generator. For example, if we fix the stacking image as cbc^{-1} and the generator as b , the language which completes this tuple is exactly those words in normal form that end in a c^{-1} and those that end in $s^{\epsilon_n}a^{i_n}$ for $i_n < 0$.

The language of normal forms can be written as a regular expression. Define the following set of subwords which result in a word not in normal form:

$$B = \left\{ \begin{array}{l} aa^{-1}, a^{-1}a, bb^{-1}, b^{-1}b, cc^{-1}, c^{-1}c, dd^{-1}, d^{-1}d, ss^{-1}, s^{-1}s, ca, ca^{-1}, c^{-1}a, \\ c^{-1}a^{-1}, da, da^{-1}, d^{-1}a, d^{-1}a^{-1}, cb, cb^{-1}, c^{-1}b, c^{-1}b^{-1}, db, db^{-1}, d^{-1}b, d^{-1}b^{-1} \end{array} \right\}$$

Using this set, we can then write the set of normal forms as the following regular expression:

$$\mathcal{N} = A^* \setminus \left\{ \left\{ A^* B A^* \right\} \cup \left\{ A^* s^{\pm 1} A^* x A^* \mid x \in \{b^{\pm 1}, c^{\pm 1}, d^{\pm 1}\} \right\} \right\}$$

We will now split into cases to show that S is autostackable.

Case 1: Fix $x \in b^{\pm 1}$ as the generator, and cxc^{-1} as the stacking image.

Then the words which complete the 3-tuple (y, x, cxc^{-1}) are both words in $w \in \mathcal{N}_G$ which end in a c^{-1} and words of the form $ws^{\epsilon_1}a^{i_1} \dots s^{\epsilon_n}a^{i_n}$ in normal form with $i_n < 0$. Note that $\mathcal{N} \cap \{A^*c^{-1}\} = \mathcal{N}_G \cap \{A^*c^{-1}\}$. The regular expression representing this collection of words is

$$S_{x,cxc^{-1}} = (\mathcal{N} \cap \{A^*c^{-1}\}) \cup (\mathcal{N} \cap A^*s^{\pm 1}A^*a^{-1})$$

The case where the stacking image is $c^{-1}xc$ is very similar.

Case 2: Fix $x \in b^{\pm 1}$ as the generator, and $z^{-1}xz$ as the stacking image where $z \in \{d^{\pm 1}\}$.

According to the stacking function, the only instance that an edge labeled by a generator x has image $\phi(y, x) = z^{-1}xz$ is when $y \in \mathcal{N}_G$ ends with the letter z . The regular expression for this language is

$$S_{x,z^{-1}xz} = \mathcal{N} \cap \{A^*z\}$$

Case 3: Fix $x = a^{\pm 1}$ as the generator, and $z^{-1}xz$ as the stacking image where $z \in \{c^{\pm 1}, d^{\pm 1}\}$.

The set of recursive edges labeled by x with stacking image $z^{-1}xz$ are precisely those whose initial vertex is labeled by a group element in G , with y_g ending in the

letter z . This language is identical to the previous case:

$$S_{x,z^{-1}xz} = \mathcal{N} \cap \{A^*z\}$$

Case 4: Fix $x \in \{c^{\pm 1}, d^{\pm 1}\}$ as the generator, and $z^{-1}xz$ as the stacking image where $z \in \{a^{\pm 1}\}$.

We require an expression consisting of normal forms of the type $ws^{\epsilon_1}a^{i_1} \dots s^{\epsilon_n}a^{i_n}$ with $i_n > 0$ if $z = a$ and $i_n < 0$ if $z = a^{-1}$. The regular expression for this language is

$$S_{x,z^{-1}xz} = \mathcal{N} \cap \{A^*s^{\pm 1}A^*z\}$$

Case 5: Fix the word $s^{-\epsilon_n}xz^{-1}s^{\epsilon_n}z$ in the image of the function ϕ where either $x \in \{b^{-1}, c^{-1}, d^{-1}\}$ and $z = a^{-1}$ or $x \in \{b, c, d\}$ and $z = a$.

The language of words in $S_{x,s^{-\epsilon_n}xz^{-1}s^{\epsilon_n}z}$ in this case are those in normal form which end in an s^{ϵ_n} . The following is a regular expression for this collection of words.

$$S_{x,s^{-\epsilon_n}xz^{-1}s^{\epsilon_n}z} = \mathcal{N} \cap \{A^*s^{\epsilon_n}\}$$

We now consider the case where we fix $a = x$ in (y, a, x) . These represent degenerate edges.

Case 6: Fix $z \in \{a, a^{-1}\}$ as both the generator and the stacking image.

This gives the set of words y such that yz is in normal form, or those which are in normal form and end with the letter z^{-1} . By the setup of the normal forms for S , all edges except those whose initial vertex has a normal form which ends with a letter $x \in \{c^{\pm 1}, d^{\pm 1}\}$ are degenerate. The regular expression accepting words of this type is

$$S_{z,z} = \mathcal{N} \setminus \{A^*x | x \in \{c^{\pm 1}, d^{\pm 1}\}\}$$

Case 7: Fix $z \in \{b, b^{-1}\}$ as both the generator and the stacking image.

The only instances in which an edge $e_{y,z}$ is degenerate are those where the word y contains only letters from $\{a^{\pm 1}, b^{\pm 1}\}$. The set of such words is

$$S_{z,z} = \mathcal{N} \cap \{a^{\pm 1}, b^{\pm 1}\}^*.$$

Case 8: Fix $z \in \{c^{\pm 1}, d^{\pm 1}\}$ as both the generator and the stacking image.

Any edge labeled by z is degenerate only when it has initial vertex label by a word $w \in \mathcal{N}_G$. The set of these words form a regular language:

$$S_{z,z} = \mathcal{N} \cap (A^* s^{\pm 1} A^*)^c$$

Case 9: Fix $z \in \{s, s^{-1}\}$ as both the generator and the stacking image.

As edges labeled by z are always degenerate, the set of words y such that $\phi(y, z) = z$ is actually the set of normal forms \mathcal{N} . □

Chapter 4

Examples

Free products with amalgamation arise often in topology, via the Seifert-van Kampen theorem. Given three groups $G = \langle A|R \rangle$, $H = \langle B|S \rangle$ and $K = \langle C \rangle$, with A , B , and C all finite, inverse-closed generating sets and injective homomorphisms $\alpha : C \hookrightarrow A$ and $\beta : C \hookrightarrow B$, the free product with amalgamation $G *_K H$ can be presented as

$$G *_K H = \langle A \cup B \cup C | R \cup S \cup \{c_i = \alpha(c_i) = \beta(c_i) | c_i \in C\} \rangle$$

When C is finite, this presentation is finite.

Theorem 4.1. *Let $G = G_1 *_H G_2$ with $\alpha : H \hookrightarrow G_1$ and $\beta : H \hookrightarrow G_2$ any injective homomorphisms. Then if $G_1 = \mathbb{Z}^n$, $G_2 = \mathbb{Z}^m$ and $H = \mathbb{Z}$, then G is autostackable.*

Before we prove this, we require a couple of lemmas.

Lemma 4.2. *Let $\alpha : \mathbb{Z} \hookrightarrow \mathbb{Z}^n$. Then there exists a generating set $\{a_1, \dots, a_n\}$ of \mathbb{Z}^n and $\{c\}$ of \mathbb{Z} such that the map α is defined as $\alpha(c) = a_1^{i_1} a_2^{i_2} \cdots a_n^{i_n}$ for $i_1, \dots, i_n \in \mathbb{Z}$ and $i_n \neq 0$. If $G \cong \mathbb{Z}^n / \langle \alpha(c) \rangle$ then*

$$\mathcal{N} = \{a_1^{j_1} \cdots a_{n-1}^{j_{n-1}} a_n^{j_n} | j_1, \dots, j_{n-1} \in \mathbb{Z}, 0 \leq j_n < |i_n|\}$$

is a set of normal forms for G .

Proof. We can, without loss of generality, assume that $i_n > 0$, as otherwise, we could use inversion to get the relation $a_1^{-i_1} \cdots a_n^{-i_n} = 1$. As G is abelian, we can write each element in G in the form $a_1^{j_1} \cdots a_n^{j_n}$. In fact, since $a_n^{i_n} = a_1^{-i_1} \cdots a_{n-1}^{-i_{n-1}}$, we can force $0 \leq j_n < i_n$. Now, suppose that a group element $g \in G$ has multiple representations by words in \mathcal{N} . That is, suppose $g = a_1^{j_1} \cdots a_n^{j_n} = a_1^{k_1} \cdots a_n^{k_n}$ with $j_1, \dots, j_{n-1}, k_1, \dots, k_{n-1} \in \mathbb{Z}$ and $0 \leq j_n, k_n < i_n$. Without loss of generality, suppose $k_n \geq j_n$. Then moving both words to one side and collecting terms, we get that $a_1^{k_1-j_1} \cdots a_n^{k_n-j_n} = 1$ with $k_1-j_1, \dots, k_{n-1}-j_{n-1} \in \mathbb{Z}$ and $0 \leq k_n-j_n < i_n$. Therefore the representation $w = a_1^{k_1-j_1} \cdots a_n^{k_n-j_n}$ is a representation of the identity. Since $w =_G 1$, we can write w in the free group as a product of conjugates of relators. Commuting relations have exponent sum zero for each generator and applying conjugation does not change exponent sum, so there must be at least one instance of the relator $a_1^{i_1} a_2^{i_2} \cdots a_n^{i_n}$, or else we have $k_i - j_i = 0$ for all $1 \leq i \leq n$. From this, we can see that $0 \leq k_n - j_n < i_n$ implies that $k_n - j_n = k \cdot i_n$ for k the number of times the relator $a_1^{i_1} a_2^{i_2} \cdots a_n^{i_n}$ appears in this product of relators. But as $k \in \mathbb{Z}$, and $k_n - j_n < i_n$, we must have $k = 0$. Therefore $k_1 - j_1 = 0, k_2 - j_2 = 0, \dots, k_n - j_n = 0$ and thus \mathcal{N} is a set of normal forms for G . \square

The following lemma gives the normal forms for a free product with amalgamation of \mathbb{Z}^n with \mathbb{Z}^m over \mathbb{Z} .

Lemma 4.3. *Let $\mathbb{Z}^n = \langle a_1, \dots, a_n \rangle, \mathbb{Z}^m = \langle b_1, \dots, b_m \rangle$ and $\mathbb{Z} = \langle c \rangle$ be three groups with injective homomorphisms $\alpha : \mathbb{Z} \hookrightarrow \mathbb{Z}^n$ and $\beta : \mathbb{Z} \hookrightarrow \mathbb{Z}^m$ defined by $\alpha(c) = a_1^{i_1} a_2^{i_2} \cdots a_n^{i_n}$, and $\beta(c) = b_1^{j_1} b_2^{j_2} \cdots b_m^{j_m}$, respectively, with $i_1, \dots, i_n, j_1, \dots, j_m \in \mathbb{Z}$. Let \mathcal{N}_n and \mathcal{N}_m be the normal form sets described in Lemma 4.2 for groups $\mathbb{Z}^n / \langle \alpha(c) \rangle$ and $\mathbb{Z}^m / \langle \beta(c) \rangle$*

respectively. Then a set of normal forms for $H = \mathbb{Z}^n *_{\mathbb{Z}} \mathbb{Z}^m$ is

$$\begin{aligned} \mathcal{N} = \{w_1 v_1 w_2 v_2 \cdots w_k v_k c^s \mid & w_2, \dots, w_k \in \mathcal{N}_n \setminus \{1\}, v_1, \dots, v_{k-1} \in \mathcal{N}_m \setminus \{1\} \\ & w_1 \in \mathcal{N}_n, v_k \in \mathcal{N}_m, c = \alpha(c) =_H \beta(c), k, s \in \mathbb{Z}\} \end{aligned}$$

Proof. This is true by the previous lemma, and the normal form theorem for free products with amalgamation [23, Chapter IV, Theorem 2.6]. \square

Lemma 4.4. Let $\mathbb{Z}^m = \langle A \rangle$, $\mathbb{Z}^n = \langle B \rangle$, and $\mathbb{Z} = \langle c \rangle$ be free abelian groups where $A = \{a_1, \dots, a_m\}$ and $B = \{b_1, \dots, b_n\}$. Let $\mathbb{Z}^m *_{\mathbb{Z}} \mathbb{Z}^n$ be the free product with amalgamation with injective homomorphisms $\alpha : \mathbb{Z} \hookrightarrow \mathbb{Z}^m$ and $\beta : \mathbb{Z} \hookrightarrow \mathbb{Z}^n$ defined by $\alpha(c) = a_1^{i_1} \cdots a_m^{i_m}$ and $\beta(c) = b_1^{j_1} \cdots b_n^{j_n}$, respectively for $i_1, \dots, i_m, j_1, \dots, j_n \in \mathbb{Z}$. Define the sets \mathcal{A} , \mathcal{B} and \mathcal{C} as

$$\begin{aligned} \mathcal{A} = \{e_{g,x} \mid y_g \in (A \cup B)^* \text{ and } x \in A, \text{ with } y_g = w \cdot w_A, \text{ where } w_A \\ \text{is the maximal suffix of } y_g \text{ in } A^* \text{ and } w_A x, y_{w_A x} x^{-1} \notin \mathcal{N}_{\mathbb{Z}^m / \alpha(c)}\} \end{aligned}$$

$$\begin{aligned} \mathcal{B} = \{e_{g,x} \mid y_g \in (A \cup B)^* \text{ and } x \in B, \text{ with } y_g = w \cdot w_B, \text{ where } w_B \\ \text{is the maximal suffix of } y_g \text{ in } B^* \text{ and } w_B x, y_{w_B x} x^{-1} \notin \mathcal{N}_{\mathbb{Z}^n / \beta(c)}\} \end{aligned}$$

$$\mathcal{C} = \{e_{g,x} \mid y_g = w c^\ell, \text{ with } \ell > 0 \text{ and } x \neq c, c^{-1}\}$$

Then using the normal form set described in Lemma 4.3, the set of recursive edges of $\mathbb{Z}^m *_{\mathbb{Z}} \mathbb{Z}^n$ is $\vec{E}_r = \mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$.

Proof. First, observe that $\mathcal{A} \cup \mathcal{B} \cup \mathcal{C} \subset \vec{E}_r$. Indeed, for an edge $e_{g,x}$ to be degenerate, we have either $y_g x = y_{gx}$ or $y_{gx} x^{-1} = y_g$; however in the sets \mathcal{A} , \mathcal{B} and \mathcal{C} , neither equality holds by assumption. For the reverse inclusion, let $e_{g,x} \in \vec{E}_r$. Then by definition, neither $y_g x = y_{gx}$ nor $y_{gx} x^{-1} = y_g$. This implies that $y_g x$ is not in normal form, nor does y_g end with the letter x^{-1} . Without loss of generality, suppose $x \in B$. Then either the word wx is not in normal form in $\mathbb{Z}^n / \beta(c)$, for w the maximal suffix in B^* at the end of y_g or y_g ends with the letter c and $x \neq c, c^{-1}$. Hence, $e_{g,x}$ is in

one of \mathcal{A} , \mathcal{B} or \mathcal{C} . □

We now have enough to prove the theorem.

Proof. (Of Theorem 4.1) Recall that the normal forms for each group $\mathbb{Z}^m/\alpha(c)$ and $\mathbb{Z}^n/\beta(c)$ give an ordering on the $<$ on the generators. Define the flow function ϕ on $\mathcal{N} \times (A \cup B \cup \{c^{\pm 1}\})$ as

$$\phi(y_g, x) = \begin{cases} z^{-1}xz & \text{for } y_g = wz \text{ where either } x, z \in A \text{ or} \\ & x, z \in B \text{ with } x < z \text{ and } z \neq a_m, b_n \\ a_m^{-(i_m-1)} a_1^{i_1} \dots a_{m-1}^{i_{m-1}} c & \text{for } y_g = wa_m^{i_m-1} \text{ and } x = a_m \\ a_m^{i_m-1} a_1^{-i_1} \dots a_{m-1}^{-i_{m-1}} c^{-1} & \text{for } y_g = wa_i, \text{ where } i \neq m, \text{ and } x = a_m^{-1} \\ b_n^{-(i_n-1)} b_1^{j_1} \dots b_{n-1}^{j_{n-1}} c & \text{for } y_g = wb_n^{i_n-1} \text{ and } x = b_n \\ b_n^{(i_n-1)} b_1^{j_1} \dots b_{n-1}^{j_{n-1}} c^{-1} & \text{for } y_g = wb_i, \text{ where } i \neq n \text{ and } x = b_n^{-1} \\ z^{-1}xz & \text{for } y_g = wz \text{ where } z \in \{c, c^{-1}\} \text{ and} \\ & x \in A \cup B \\ x & \text{for either } y_g x = y_{gx} \text{ or } y_{gx} x^{-1} = y_g \\ & \text{where } x \in A \cup B \cup \{c^{\pm 1}\} \end{cases}$$

From the definition of ϕ above it follows that $\phi(y_g, x) =_G x$ and $\phi(y_g, x) = x$ if and only if $e_{g,x}$ is degenerate. To see that $<_\phi$ is a strict well-founded partial order on \vec{E}_r , define a function $\psi : \vec{E}_r \rightarrow \mathcal{N}^2$ by

$$\psi(e_{g,x}) = \begin{cases} (0, \ell(y_g)) & \text{for } g \in G \text{ with } y_g \in \{A \cup B\}^* \text{ and } x \in A \cup B \setminus \{a_m^{\pm 1}, b_n^{\pm 1}\} \\ (1, 0) & \text{for } g \in G \text{ with } y_g \in \{A \cup B\}^* \text{ and } x \in \{a_m^{\pm 1}, b_n^{\pm 1}\} \\ (1, \ell_c(y_g)) & \text{for } g \in G \text{ with } y_g = wd^\epsilon \text{ for } \epsilon \in \{\pm 1\} \text{ and } x \in A \cup B \end{cases}$$

where $\ell(y_g)$ is the length of the word y_g and $\ell_c(y_g)$ is the length of the c -tail of the word y_g . Define an ordering $<_\psi$ on the edges in \vec{E}_r by $e' <_\psi e$ if and only if $\psi(e') <_{\mathbb{N}^2} \psi(e)$

for $<_{\mathbb{N}^2}$ the lexicographical ordering on \mathbb{N}^2 . We show here that for e' a child of an edge e , we have $e' <_{\psi} e$. It follows that $<_{\phi}$ is a strict well-founded partial order.

Case 1: Let $e_{g,x}$ be a recursive edge with $y_g = wa_i$ and $x = a_j$ for $a_i, a_j \in A$ and $j < i \neq m$.

The edge $e_{g,x}$ in this case has ψ -function value $\psi(e_{g,x}) = (0, \ell(y_g))$. The edge $e_{g,x}$ has three children: $e_{g,a_i^{-1}}$, $e_{ga_i^{-1},x}$ and $e_{ga_i^{-1}x,a_i}$. The first and third children are degenerate. The second child, $e_{ga_i^{-1},x}$ has ψ function value of $\psi(e_{ga_i^{-1},x}) = (0, \ell(y_{ga_i^{-1}}))$. As $\ell(y_{ga_i^{-1}}) = \ell(y_g) - 1$, we have $\psi(e_{ga_i^{-1},x}) <_{\mathbb{N}^2} \psi(e_{g,x})$. Therefore $e_{ga_i^{-1},x} <_{\psi} e_{g,x}$.

The case where $y_g = wb_i$ and $x = b_j$ for $b_i, b_j \in B$ with $j < i \neq n$ follows a similar argument.

Case 2: Let $e_{g,x}$ be a recursive edge with $y_g = wa_m^{i_m-1}$ and $x = a_m$.

In this case, the ψ -function value for the edge $e_{g,x}$ is $\psi(e_{g,x}) = (1, 0)$. The first i_m-1 children of this edge are degenerate, as they are peeling off the i_m-1 appearances of a_m on the word y_g . The final child, labeled by the letter c is a degenerate edge as well. The intermediate children, labeled by $a_j^{i_j}$ for $j = 1, \dots, i_m-1$ are not necessarily degenerate, depending on the exact word y_g , but all have ψ function value of the form $(0, t)$ for $t \in \mathcal{N}$ the length of the normal form of the initial vertex of each child. However, $(0, t) < (1, 0)$ and so each recursive child e' of the edge $e_{g,x}$ has $\psi(e') <_{\mathbb{N}^2} \psi(e_{g,x})$. Thus $e' <_{\psi} e_{g,x}$.

The cases where $y_g = wb_n^{i_n-1}$ with $x = b_n$, $y_g = wa_i$ for $i \neq m$ and $x = a_m^{-1}$, and $y_g = wb_i$ for $i \neq n$ and $x = b_n^{-1}$ follow similar arguments.

Case 3: Let $e_{g,x}$ be a recursive edge with $y_g = wc$ with $x \in A \cup B$.

The ψ -function value for the edge $e_{g,x}$ in this case is $\psi(e_{g,x}) = (1, \ell_c(y_g))$. The stacking function gives that this edge has three children: $e_{g,c^{-1}}$, $e_{gc^{-1},x}$ and $e_{gc^{-1}x,c}$. The first and third children are both degenerate. If the second child is recursive, then $\psi(e_{gc^{-1},x}) = (1, \ell_c(y_{gc^{-1}}))$. Notice that $\ell_c(y_{gc^{-1}}) = \ell_c(y_g) - 1$ and so $\psi(e_{gc^{-1},x}) <_{\mathbb{N}^2}$

$\psi(e_{g,x})$. Thus $e_{gc^{-1},x} <_\psi e_{g,x}$.

The case where $y_g = wc^{-1}$ follows a similar argument.

Now, we show that

$$\text{graph}(\phi) = \{(y, a, \phi(y, a)) | y \in \mathcal{N}, a \in (A \cup B \cup \{c^{\pm 1}\})\}$$

is a regular language. We will first show that the set \mathcal{N} of normal forms is a regular language and then we proceed using Lemma 1.5.

Let \mathcal{N}_1 and \mathcal{N}_2 be the languages of normal forms for the groups $\mathbb{Z}^m / \langle \alpha(c) \rangle$ and $\mathbb{Z}^n / \langle \beta(c) \rangle$ described in Lemma 4.2, respectively. Let $\mathcal{N}_3 := \{c^k | k \in \mathbb{Z}\}$. We can write

$$\mathcal{N}_1 = \bigcup_{k=0}^{i_m-1} \left\{ \{a_1\}^* \cup \{a_1^{-1}\}^* \right\} \cdots \left\{ \{a_{m-1}\}^* \cup \{a_{m-1}^{-1}\}^* \right\} \cdot a_m^k$$

and

$$\mathcal{N}_2 = \bigcup_{k=0}^{j_n-1} \left\{ \{b_1\}^* \cup \{b_1^{-1}\}^* \right\} \cdots \left\{ \{b_{n-1}\}^* \cup \{b_{n-1}^{-1}\}^* \right\} \cdot b_n^k$$

Thus \mathcal{N}_1 and \mathcal{N}_2 are regular languages. The language \mathcal{N}_3 is also regular, as we can express it as $\mathcal{N}_3 = \{c\}^* \cup \{c^{-1}\}^*$. The normal form set \mathcal{N} can be viewed as a free product of the languages \mathcal{N}_1 and \mathcal{N}_2 concatenated with \mathcal{N}_3 . In the proof of Theorem 2.7, we show that the free product of regular languages is regular and therefore the language \mathcal{N} is regular.

Case 1: Let $a \in A \cup B$, and $d^{-1}ad \in \text{im}(\phi)$ for $d \in (A \cup B \cup \{c, c^{-1}\}) \setminus \{a_m, b_n, a_m^{-1}, b_n^{-1}\}$ be fixed.

The words y that complete this tuple are those in normal form which end in the letter d . The regular expression for this language is

$$S_{a,d^{-1}ad} = \mathcal{N} \cap \left[(A \cup B \cup \{c, c^{-1}\})^* d \right]$$

Case 2: Let $a = a_m$, and $a_m^{-(i_m-1)} a_1^{i_1} \dots a_{m-1}^{i_{m-1}} c \in \text{im}(\phi)$

Then the words y desired here are those which are in normal form which end with $a_m^{i_m-1}$. The regular expression for this language is

$$S_{a_m, a_m^{-(i_m-1)} a_1^{i_1} \dots a_{m-1}^{i_{m-1}} c} = \mathcal{N} \cap \left[(A \cup B \cup \{c, c^{-1}\})^* \setminus ((A \cup B \cup \{c, c^{-1}\})^* a_m) \cdot a_m^{i_m-1} \right]$$

Notice that i_m is a finite, predetermined number, and so this is, in fact, a regular expression. The 3-tuples $(y, a_m^{-1}, a_m^{i_m-1} a_1^{-i_1} \dots a_{m-1}^{-i_{m-1}} c^{-1})$, $(y, b_n, b_n^{-(i_n-1)} b_1^{-j_1} \dots b_{n-1}^{-j_{n-1}} c)$ and $(y, b_n^{-1}, b_n^{(i_n-1)} b_1^{j_1} \dots b_{n-1}^{j_{n-1}} c^{-1})$ are similar to case 2.

Case 3: Let $x \in A \cup B$.

In this case, we seek degenerate edges of the form $e_{g,x}$. Let $x = a_i$ for some i , and for now, assume that $i \neq m$. For e_{g,a_i} to be degenerate, we need either $y_g = wb_k$ or $y_g = wa_j$ with $j \leq i$. Then

$$S_{a_i, a_i} = \mathcal{N} \cap \left[\left((A \cup B)^* (B^* \setminus \{1\}) \right) \cup \left(\bigcup_{j=1}^i (A \cup B)^* a_j^{\pm 1} \right) \right]$$

The above works identically if $x = a_i^{-1}$. If $i = m$, we must be more careful, as we now must keep track of the length of an a_m tail. Note that in our ordering, $a_m \geq a_i$ for all i , and thus the set of normal forms we are looking for are those which do not end in a c, c^{-1} and those which do not end in $a_m^{i_m-1}$. We can account for both of these situations:

$$S_{a_m, a_m} = \mathcal{N} \cap \left[(A \cup B)^* \setminus (A \cup B)^* a_m^{i_m-1} \right]$$

For the case that $x = a_m^{-1}$, the same idea applies. In fact, this same idea will also cover the cases of $x \in B$.

Case 4: Let $x \in \{c, c^{-1}\}$.

As mentioned previously, edges labeled by $c^{\pm 1}$ are degenerate, and thus $S_{x,x}$ is the

normal form set.

□

Bibliography

- [1] Juan M. Alonso. Combings of groups. In *Algorithms and classification in combinatorial group theory (Berkeley, CA, 1989)*, volume 23 of *Math. Sci. Res. Inst. Publ.*, pages 165–178. Springer, New York, 1992.
- [2] David J. Anick. On the homology of associative algebras. *Trans. Amer. Math. Soc.*, 296:641–659, 1986.
- [3] Robert Bieri. *Homological dimension of discrete groups*. Queen Mary College Mathematical Notes. Queen Mary College Department of Pure Mathematics, London, second edition, 1981.
- [4] Mark Brittenham and Susan Hermiller. Stackable groups, tame filling invariants, and algorithmic properties of groups. *arXiv:1109.6309*.
- [5] Mark Brittenham and Susan Hermiller. Algorithms and topology of Cayley graphs for groups. *Preprint*, 2013.
- [6] Kenneth S. Brown. *Cohomology of groups*, volume 87 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1982.
- [7] Kenneth S. Brown. The geometry of rewriting systems: a proof of the Anick-Groves-Squier theorem. In *Algorithms and classification in combinatorial group*

- theory (Berkeley, CA, 1989)*, volume 23 of *Math. Sci. Res. Inst. Publ.*, pages 137–163. Springer, New York, 1992.
- [8] Laura Ciobanu and Susan Hermiller. Conjugacy growth series and languages in groups. *Trans. Amer. Math. Soc.*, to appear; arXiv:1205.3857.
 - [9] Daniel E. Cohen. String rewriting and homology of monoids. *Math. Structures Comput. Sci.*, 7:207–240, 1997.
 - [10] Will Dison, Murray Elder, Timothy R. Riley, and Robert Young. The Dehn function of Stallings’ group. *Geom. Funct. Anal.*, 19:406–422, 2009.
 - [11] Murray Elder and Susan Hermiller. Minimal almost convexity. *J. Group Theory*, 8:239–266, 2005.
 - [12] David B. A. Epstein, James W. Cannon, Derek F. Holt, Silvio V. F. Levy, Michael S. Paterson, and William P. Thurston. *Word processing in groups*. Jones and Bartlett Publishers, Boston, MA, 1992.
 - [13] Daniel R. Farkas. The Anick resolution. *J. Pure Appl. Algebra*, 79:159–168, 1992.
 - [14] S. M. Gersten. Dehn functions and l_1 -norms of finite presentations. In *Algorithms and classification in combinatorial group theory (Berkeley, CA, 1989)*, volume 23 of *Math. Sci. Res. Inst. Publ.*, pages 195–224. Springer, New York, 1992.
 - [15] Elisabeth Green. Graph products of groups. *PhD thesis, University of Leeds*, 1990.
 - [16] J. R. J. Groves. Rewriting systems and homology of groups. In *Groups—Canberra 1989*, volume 1456 of *Lecture Notes in Math.*, pages 114–141. Springer, Berlin, 1990.

- [17] J. R. J. Groves and G. C. Smith. Soluble groups with a finite rewriting system. *Proc. Edinburgh Math. Soc. (2)*, 36:283–288, 1993.
- [18] Susan Hermiller and John Meier. Algorithms and geometry for graph products of groups. *J. Algebra*, 171:230–257, 1995.
- [19] Susan M. Hermiller and John Meier. Artin groups, rewriting systems and three-manifolds. *J. Pure Appl. Algebra*, 136:141–156, 1999.
- [20] Tim Hsu and Daniel T. Wise. On linear and residual properties of graph products. *Michigan Math. J.*, 46:251–259, 1999.
- [21] Yuji Kobayashi. Complete rewriting systems and homology of monoid algebras. *J. Pure Appl. Algebra*, 65:263–275, 1990.
- [22] Yves Lafont. A new finiteness condition for monoids presented by complete rewriting systems (after Craig C. Squier). *J. Pure Appl. Algebra*, 98:229–244, 1995.
- [23] Roger C. Lyndon and Paul E. Schupp. *Combinatorial group theory*. Classics in Mathematics. Springer-Verlag, Berlin, 2001.
- [24] Charles F. Miller, III. *On group-theoretic decision problems and their classification*. Princeton University Press, Princeton, N.J., 1971. Annals of Mathematics Studies, No. 68.
- [25] Michael Sipser. *Introduction to the Theory of Computation*. Cengage Learning, 2012.
- [26] Craig C. Squier. Word problems and a homological finiteness condition for monoids. *J. Pure Appl. Algebra*, 49:201–217, 1987.

- [27] John Stallings. A finitely presented group whose 3-dimensional integral homology is not finitely generated. *Amer. J. Math.*, 85:541–543, 1963.