5-2023

Prefix-Rewriting: The Falsification by Fellow Traveler Property and Practical Computation

Ash DeClerk

University of Nebraska-Lincoln, declerk.gary@huskers.unl.edu

PREFIX-REWRITING: THE FALSIFICATION BY FELLOW TRAVELER PROPERTY AND PRACTICAL COMPUTATION

by

Ash DeClerk

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 Professor Susan Hermiller

Lincoln, Nebraska

May, 2023

PREFIX-REWRITING: THE FALSIFICATION BY FELLOW TRAVELER PROPERTY AND PRACTICAL COMPUTATION

Ash DeClerk, Ph.D.

University of Nebraska, 2023

Adviser: Susan Hermiller

The word problem is one of the fundamental areas of research in infinite group theory, and rewriting systems (including finite convergent rewriting systems, automatic structures, and autostackable structures) are key approaches to working on the word problem. In this dissertation, we discuss two approaches to creating bounded regular convergent prefix-rewriting systems.

Groups with the falsification by fellow traveler property are known to have solvable word problem, but they are not known to be automatic or to have finite convergent rewriting systems. We show that groups with this geometric property are geodesically autostackable. As a key part of proving this, we show that a wider class of groups, namely groups with a weight non-increasing synchronously regular convergent prefix-rewriting system, have a bounded regular convergent prefix-rewriting system.

Our second approach to creating prefix-rewriting systems is a more general approach. We design a procedure that, when provided with a finitely presented group $G = \langle A \mid R \rangle$ and an ordering $<$ on A^*, searches for a bounded convergent prefix-rewriting system. We also create a class of orderings for which each step of this procedure can be practically computed, and which guarantees that any bounded convergent prefix-rewriting system is an autostackable structure.

ACKNOWLEDGMENTS

I would like to give my thanks to dozens of people for their help over the past several years. This would have been an impossible task without you.

To my advisor, Susan Hermiller, for her patience, her guidance, and her extremely thorough feedback.

To my readers, Mark Brittenham and Alex Zupan, for their advice throughout the years and their feedback on this dissertation in particular.

To the GST seminar's regulars (a group too large to list), for listening to me talk about what I've learned and sharing what they've learned over the years.

To my classmates (from first year all the way to now), for late nights working on homework together, and pointing out every time I overcomplicated things.

To everyone I've taught alongside, for helping me improve as a teacher.

To everyone else in the UNL math department, for providing a community that I am proud to be a part of.

And to my mother, for her lasting support.

The author received partial support from Simons Foundation Collaboration Grant number 581433.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

One of the primary problems in geometric group theory is the word problem, introduced by Dehn in 1911 [11]. Given a group and finite generating set $G = \langle A \rangle$, the word problem asks whether there exists an algorithm which, given a word $w \in A^*$ as input, outputs whether or not w represents the trivial element in G. This problem was shown to be unsolvable for finitely presented groups by Novikov in 1955 [28] and Boone in 1958 [2], but there are procedures which, given a presentation for a group, search for algorithmic solutions to the word problem, with these solutions guaranteed to be correct if they are given as output. Two such procedures are the Knuth-Bendix completion procedure [22], which searches for finite convergent rewriting systems, and the procedure by Epstein, Holt, and Rees [15], which searches for automatic structures. These two types of word problem solutions are key tools in studying groups.

Autostackability was introduced by Brittenham, Hermiller, and Holt [4] as a generalization of automaticity and the property of having a finite convergent rewriting system. Autostackable structures provide a solution to the word problem for a wider class of groups than either automatic structures or finite convergent rewriting systems, and are defined in terms of the Cayley graph $\Gamma_{G,A}$ for a group G with inverse-closed generating set A. Given a spanning tree T in $\Gamma_{G,A}$, a *regular bounded flow function* for the triple (G, A, T) is a function Φ from the set of directed edges of $\Gamma_{G,A}$ to the set of directed paths in $\Gamma_{G,A}$ such that:

- (same endpoints) for any directed edge e of $\Gamma_{G,A}$, the path $\Phi(e)$ has the same initial vertex and the same terminal vertex as e;
- (boundedness) there exists a constant $k \geq 0$ such that for each edge e of $\Gamma_{G,A}$, the path $\Phi(e)$ has length at most k;
- (fixed tree edges) if the undirected edge underlying e lies in the tree T, then $\Phi(e) = e$;
- (termination) there is no infinite sequence of edges $e_1, e_2, e_3, \ldots$ such that each e_{i+1} lies in the path $\Phi(e_i)$ and each e_i lies outside the spanning tree T; and
- (regularity) the language of triples $(\mathrm{nf}_\Phi(\iota(e)), \mathrm{label}(e), \mathrm{label}(\Phi(e)))$ is a synchronously regular language (where $\mathrm{nf}_\Phi(v)$ is the label of the unique non-backtracking path from 1_G to v in T, $\iota(e)$ is the starting vertex of the edge e, and $\mathrm{label}(p)$ is the label of the path p).

(See Section 1.2 for notation and background on regular languages.)

Definition 1.1.1. *[4] A group is* autostackable *if it admits a regular bounded flow function.*

Many classes of groups have been shown to be autostackable for certain generating sets, including all prefix-closed automatic groups [4] and all groups with a finite convergent rewriting system [4], Thompson's group F [8], closed 3-manifold groups

[6], the Baumslag-Gersten group [16], and Stallings' finitely presented group which is not of type FP_3 [5].

Throughout this dissertation, we will also consider convergent prefix-rewriting systems:

Definition 1.1.2. *[4] A* convergent prefix-rewriting system (CP-RS) *for a group G is a pair consisting of a finite alphabet A and a set $R \subseteq A^* \times A^*$ of ordered pairs of words over A such that G is presented as a monoid by*

$$G = Mon\langle A \mid \{u = v \mid (u, v) \in R\}\rangle$$

and the set of rewritings $uw \to_R vw$ with $(u, v) \in R$ and $w \in A^*$ satisfies:

- *(termination) there is no infinite chain of rewritings*

$$w_1 \to_R w_2 \to_R w_3 \to_R \cdots$$

  and

- *(normal forms) each element of G is represented by exactly one irreducible word (i.e. a word which cannot be rewritten) over A.*

A CP-RS R is said to be *length non-increasing* if for each $(u, v) \in R$, we have $|v| \leq |u|$, and *bounded* if there exists a constant k such that for each pair (l, r) in R, we have there exist some words $p, l', r' \in A^*$ such that $l = pl'$, $r = pr'$, and $|l'| + |r'| \leq k$. We will say that a CP-RS is *end-normal* if for all pairs $(u, v) \in R$ with $u = wa$ for some irreducible word $w \in A^*$ and some letter $a \in A$, $\text{last}(v) = \text{last}(\text{nf}_R(v))$. A CP-RS is *(synchronously) regular* if the set R is a (synchronously) regular language. If we have a system of positive finite weights on A (that is, a function $wt : A \to \mathbb{R}^+$), then we define $wt(a_1a_2\cdots a_n) = \sum_{i=1}^{n} wt(a_i)$; then R is *weight non-increasing* if for each $(u,v) \in R$ we have $wt(v) \leq wt(u)$. In particular, a length non-increasing CP-RS is also weight non-increasing, where $wt(a) = 1$ for all $a \in A$.

Bounded synchronously regular convergent prefix-rewriting systems and autostackable groups are intimately connected. In particular:

Proposition 1.1.3. *[4] A group G is autostackable if and only if G admits a bounded synchronously regular convergent prefix-rewriting system. Moreover, there is an algorithm which, given a bounded synchronously regular convergent prefix-rewriting system R, can construct an autostackable structure such that the normal forms of this autostackable structure are the same as the normal forms of R.*

Autostackable structures are powerful tools in tackling the word problem. In this dissertation, we extend the current knowledge of autostackable structures with two key results. The first gives conditions under which we know that a group has an autostackable structure:

Theorem 2.3.1. *Suppose $G = \langle A \rangle$ has a weight non-increasing synchronously regular CP-RS R. Then G has an autostackable structure which has the same normal forms as R.*

We then use this to show that all groups with the falsification by fellow traveler property are autostackable.

Our second result is a procedure to search for autostackable structures, which we call the prefix-Knuth-Bendix procedure. This allows us, on input of a group $G = Mon\langle A \mid R\rangle$ and an ordering $<$ satisfying certain conditions, to have a computer search for a bounded regular prefix-rewriting system and prove that it is convergent, rather than using the ad hoc methods which have been common thus far.

Theorem 3.3.7. *Suppose that $\mathcal{R}_n$ is the result of using the prefix-Knuth-Bendix procedure with inputs of a monoid $M = Mon\langle A \mid R\rangle$ and a well-founded strict partial ordering $<$ which is compatible with concatenation on the right. Then $\mathcal{R}_n$ is a bounded convergent prefix-rewriting system for M.*

We also define a new class of orderings, the *k-bounded regular orderings*, which result in regular prefix-rewriting systems when used with the prefix-Knuth-Bendix procedure. If the monoid is in fact a group, the resulting bounded regular convergent prefix-rewriting system is an autostackable structure.

Corollary 3.4.5. *Suppose that $\mathcal{R}_n$ is the result of using the prefix-Knuth-Bendix procedure with inputs of a monoid $M = Mon\langle A \mid R\rangle$ and a well-founded k-bounded regular strict partial ordering $<$ which is compatible with concatenation on the right, and that no step of the prefix-Knuth-Bendix procedure required the comparison of two words u and v which differ on a suffix of length greater than k. Then $\mathcal{R}_n$ is a bounded regular convergent prefix-rewriting system for M, and is an autostackable structure if M is a group.*

This dissertation is organized as follows: Section 1.2 gives background information that will be relevant throughout the dissertation. Chapter 2 includes discussion of the falsification by fellow traveler property and our results involving it. Chapter 3 includes a description of the prefix-Knuth-Bendix procedure that we have developed and of k-bounded regular orderings, along with our other results that are relevant to these topics. We note that the results of Chapter 2 are also available in [9].

1.2 Background

This section includes definitions, theorems, and notation that will be relevant to both Chapter 2 and Chapter 3.

Throughout this dissertation, let G be a group with finite generating set A which is closed under inversion. For words $u, v \in A^*$, we use $u =_{A^*} v$ or $u = v$ to mean that u and v are identical words, and $u =_G v$ to mean that u and v represent the same group element in G. We will use $\overline{u}$ to denote the group element represented by u and use λ to denote the empty word.

Given a (prefix-) rewriting system R, each element (l, r) in R is called a *rule* of R. We say that $u =_R v$ if there exist words $w_1, \ldots, w_n$ with $u =_{A^*} w_1$, $v =_{A^*} w_n$, and either $w_i \to_R w_{i+1}$ or $w_{i+1} \to_R w_i$ for each i. Note that if R is a prefix-rewriting system for G, then $u =_R v$ if and only if $u =_G v$.

We use $|u|$ to denote the length of u, $u(n)$ to denote the prefix of u with exactly n letters if $n \leq |u|$ and all of u if $n \geq |u|$, $\text{first}(u)$ to denote the first letter of u, and $\text{last}(u)$ to denote the last letter of u; we will use the convention that $\text{first}(\lambda) = \lambda$ and $\text{last}(\lambda) = \lambda$. For a group element g or a word u representing g, we use $|g|_{G,A}$ or $|u|_{G,A}$ to denote the minimum length of any word $w \in A^*$ with $w =_G g$. When we have a set of normal forms $\mathcal{N}$ (i.e. a set of unique representatives of the elements of G), we will use $\text{nf}(u)$ to denote the normal form of the group element represented by u and $\text{nf}(g)$ to denote the normal form of g, and where we have multiple normal form sets we will use subscripts based on the source of these normal forms (e.g. $\text{nf}_R(u)$) to specify which normal forms we mean. We will call any word which is not a normal form *reducible*, and any reducible word of the form ul for some normal form $u \in \mathcal{N}$ and letter $l \in A$ *minimally reducible*.

1.2.1 Formal Language Theory

In this subsection we provide definitions and background related to language theory. The reader can refer to [21] for a more detailed treatment. A *language* over an alphabet A is a subset of A^*. The *regular languages* consist of all finite languages, along with their closure under finitely many unions, intersections, complements, concatenations ($P \cdot S = \{ps \mid p \in P, s \in S\}$), and Kleene stars ($S^0 = \{\lambda\}$, $S^i = S^{i-1} \cdot S$ for all natural numbers i, and $S^* = \bigcup_{i=0}^{\infty} S^i$). While it is not immediate, the class of regular languages is also closed under quotients ($P/S = \{w \in A^* \mid$ there exists $s \in S$ such that $ws \in P\}$) [21, Theorem 3.6] and under homomorphic image (given a map $f : A \to B^*$, we extend f to a map $f' : A^* \to B^*$ by $f'(\lambda) = \lambda$ and $f'(a_1 \dots a_n) = f(a_1) \cdots f(a_n)$; then $f'(P)$ is the homomorphic image of P under f) [21, Theorem 3.5]. We use A^k to denote the set of all words over A of length exactly k, and $A^{\leq k}$ to denote the set of all words over A with length at most k.

Definition 1.2.1. *[21, Page 17] A* finite state automaton (FSA) *consists of a finite set of states Q, a finite input alphabet A, a transition function $\delta : Q \times A \to Q$, an initial state q_0, and a set of accepting states $P \subseteq Q$. Given a finite state automaton M, $\widehat{\delta} : Q \times A^* \to Q$ is the function given by $\widehat{\delta}(q, \lambda) = q$ and $\widehat{\delta}(q, aw) = \widehat{\delta}(\delta(q, a), w)$. A word u is* accepted *by M if $\widehat{\delta}(q_0, u) \in P$, and the* language accepted by M *is the set of all words which are accepted by M.*

We will at times view finite state automata as labeled directed graphs with vertex set Q. Under this view, the edges are given by the transition function, with $\delta(q, a) = q'$ giving a directed edge from q to q' labeled by a. A path in this graph from a state q to a second state q' is labeled by a word w with $\widehat{\delta}(q, w) = q'$.

A language L is regular if and only if L is the language accepted by some finite state automaton [21, Section 2.5]. We will use both the definition of regular languages and the equivalent idea of languages accepted by FSAs throughout this dissertation. We will also frequently use synchronously regular languages:

Definition 1.2.2. *[14, Definition 1.4.4] Given a language* $L \subseteq A_1^* \times \cdots \times A_n^*$ *and padding symbols* $\$_i \notin A_i$ *for each* i*, the language* L^p *over the padded alphabet* $B = (A_1 \cup \{\$_1\}) \times \cdots \times (A_n \cup \{\$_n\})$ *is defined as follows:*

- *For each* n*-tuple* $(w_1, \ldots, w_n) \in L$*, let* $m = max\{|w_i| \mid 1 \leq i \leq n\}$.
- *Let* $\widehat{w_i} := w_i \$_i^{m-|w_i|} = c_{i,1} \cdots c_{i,m}$ *for some* $c_{i,j} \in A_i \cup \{\$_i\}$*. Let* $b_j = (c_{1,j}, \ldots, c_{n,j}) \in B$*. Then* $(w_1, \ldots, w_n)^p := b_1 \cdots b_m$.
- $L^p = \{(w_1, \ldots, w_n)^p \in B^* \mid (w_1, \ldots, w_n) \in L\}$.

L^p *is the* padded extension of L*. A language* L *over* $A_1 \times \cdots \times A_n$ *is* synchronously regular *if* L^p *is a regular language over* B.

When we consider products of languages, we will consider them as padded languages unless otherwise specified. When we consider the concatenation of two languages over a product alphabet $L \cdot K$, we will concatenate each entry of the tuple before padding and subsequently move any pre-existing padding to the end; that is, $L^p \cdot K^p$ is defined to be $(L \cdot K)^p = \{(u_1v_1, \ldots, u_nv_n)^p \mid (u_1, \ldots, u_n) \in L$ and $(v_1, \ldots, v_n) \in K\}$. For any product alphabet $A_1 \times \cdots \times A_n$, the projection function $\pi_i : B \to A_i$ is the function given by $\pi_i(a_1, \ldots, a_n) = a_i$ when $a_i \in A_i$ and $\pi_i(a_1, \ldots, a_n) = \lambda$ when $a_i = \$_i$.

Lemma 1.2.3. *(Pumping Lemma, [21, Lemma 3.1 and Exercise 3.2]) Let* L *be a regular language. Then there exists a natural number* n *such that for any word* z *with* $|z| \geq n$ *and any words* p, s *with* $pzs \in L$*, we can decompose* $pzs =_{A^*} puvws$ *with* $|uv| \leq n$*,* $|v| \geq 1$ *so that for all* $i \geq 0$ *the word* puv^iws *is an element of* L*. In* *particular, given an FSA M accepting L, we can take n to be the number of states of M.*

The smallest such n is the *pumping number* of L.

Chapter 2

The Falsification by Fellow Traveler Property Implies Geodesic Autostackability

2.1 Falsification by Fellow Traveler Property Background

For a group G with a finite inverse-closed set A which generates G, the falsification by fellow traveler property (FFTP) is a purely geometric property of the Cayley graph which was introduced by Neumann and Shapiro [24].

Definition 2.1.1. *[24] The pair* (G, A) *has the* falsification by fellow traveler property (FFTP) *if there exists a constant* k *such that for each non-geodesic word* u *in the generators and their inverses, there exists a word* v *such that* $|v| < |u|$, $u =_G v$, *and* u *and* v *k-fellow travel. That is,* $d(u(n), v(n)) \leq k$ *for all* $n \in \mathbb{N}$. *Such a word* v *is called a* witness *of* u.

Cannon gave an example showing that FFTP depends on the generating set for a group [24], but many choices of group G have at least one generating set A such that the pair (G, A) has FFTP. Some examples include virtually abelian groups and geometrically finite hyperbolic groups [24], Garside groups [19], Artin groups of large type [20], Coxeter groups [27], and groups acting cellularly on locally finite CAT(0) cube complexes with a simply transitive action on the vertices [26]. Certain particularly nice groups are known to have FFTP for every choice of generating set; abelian groups [24] and finite groups are two examples, though even virtually abelian groups can have generating sets such that (G, A) does not have FFTP [24] — we will use one such example in Section 2.5. Furthermore, pairs (G, A) with FFTP have particularly nice properties, including regular geodesic language [24], at most quadratic isoperimetric inequality [13], and almost convexity [13].

We extend Proposition 1.1.3, which shows that groups with a bounded synchronously regular convergent prefix-rewriting system are autostackable, to groups with weight non-increasing synchronously regular convergent prefix-rewriting systems with the following theorem:

Theorem 2.3.1. *Suppose $G = \langle A \rangle$ has a weight non-increasing synchronously regular CP-RS R. Then G has an autostackable structure which has the same normal forms as R.*

We will also discuss a more restrictive version of autostackability in this chapter, namely geodesic autostackability.

Definition 2.1.2. *[3] Define $\alpha : E(\Gamma_{G,A}) \to \mathbb{Q}$ by*

$$\alpha(e) = \frac{1}{2}\left(d_{G,A}(1_G, \iota(e)) + d_{G,A}(1_G, \tau(e))\right).$$

A group G is geodesically autostackable *if G has an autostackable structure with normal form set $\mathcal{N}$ and flow function Φ such that each element of $\mathcal{N}$ labels a geodesic in $\Gamma_{G,A}$, and whenever $e', e \in E(\Gamma_{G,A})$ with $\Phi(e') \neq e'$ and e' an edge in the path $\Phi(e)$, we have $\alpha(e') < \alpha(e)$.*

Geodesically autostackable structures are significantly more restrictive than autostackable structures. Geodesically autostackable structures provide a regular set of geodesic normal forms and require that the pair (G, A) is almost convex.

Proposition 2.1.3. *[3] A geodesically autostackable group $G = \langle A \rangle$ is almost convex with respect to the generating set for which G has a geodesically autostackable structure.*

(A definition of almost convexity is provided as Definition 2.2.2.)

We strengthen this result with Theorem 2.3.3 from the present chapter.

Theorem 2.3.3. *Suppose $G = \langle A \rangle$ has a length non-increasing synchronously regular convergent prefix-rewriting system. Then G is almost convex.*

We also give sufficient conditions for our proof of Theorem 2.3.1 to build a geodesically autostackable structure from a length non-increasing synchronously regular CP-RS with the following theorem:

Theorem 2.3.2. *Suppose that $G = \langle A \rangle$ admits a length non-increasing end-normal synchronously regular CP-RS R. Then the autostackable structure for G constructed in the proof of Theorem 2.3.1 is a geodesically autostackable structure.*

Our main theorem in this chapter is that all pairs (G, A) with FFTP are geodesically autostackable, which we prove by applying Theorem 2.3.2.

Theorem 2.4.2. *[Main Theorem] Suppose the pair (G, A) has the falsification by fellow traveler property. Then*

(a) G admits a length non-increasing end-normal synchronously regular convergent prefix-rewriting system R; and

(b) G is geodesically autostackable.

There are several related implications between the theorems proved in this chapter and other theorems mentioned in this section, Section 1.1, and Section 2.2. In order to better visualize these implications, we have included a flowchart with the results of this chapter and related results from previous papers as Figure 2.1.

The current chapter is organized as follows: In Section 2.2, we provide notation, definitions, and theorems which are used in the remainder of the chapter. In Section 2.3, we give a constructive proof of Theorem 2.3.1, prove Theorem 2.3.2 as a consequence of adding additional hypotheses to Theorem 2.3.1, and prove Theorem 2.3.3. In Section 2.4, we prove Theorem 2.4.2, the main theorem of this chapter. In Section 2.5, we provide an example of a pair (G, A) with length non-increasing synchronously regular CP-RS which do not have FFTP with the associated generating set. As a consequence of this, the converse of part (a) of the main theorem fails.

2.2 Notation, Definitions, and Background

Let $\Gamma_{G,A}$ be the Cayley graph of G with respect to generating set A. Then for a directed edge e of $\Gamma_{G,A}$, we use $\iota(e)$ to denote the initial vertex of e, $\tau(e)$ to denote the terminal vertex of e, and label(e) to denote the label from A associated to e. We also denote the vertex corresponding to the identity element of G by 1_G, and $e_{y,a}$ to denote the directed edge with initial vertex y and label a. For an edge-path p in $\Gamma_{G,A}$, we use label(p) to denote the word in A^* obtained by concatenating the labels of each edge in p. We will use $d_{G,A}(\overline{u}, \overline{v})$ to denote the path metric on $\Gamma_{G,A}$.

Definition 2.2.1. *For an alphabet A with a strict total ordering $\prec$, a word u is said to be* short reverse lexicographically smaller *than a word v ($u <_{srev} v$) if $u \neq v$ and either*

- $|u| < |v|$*; or*

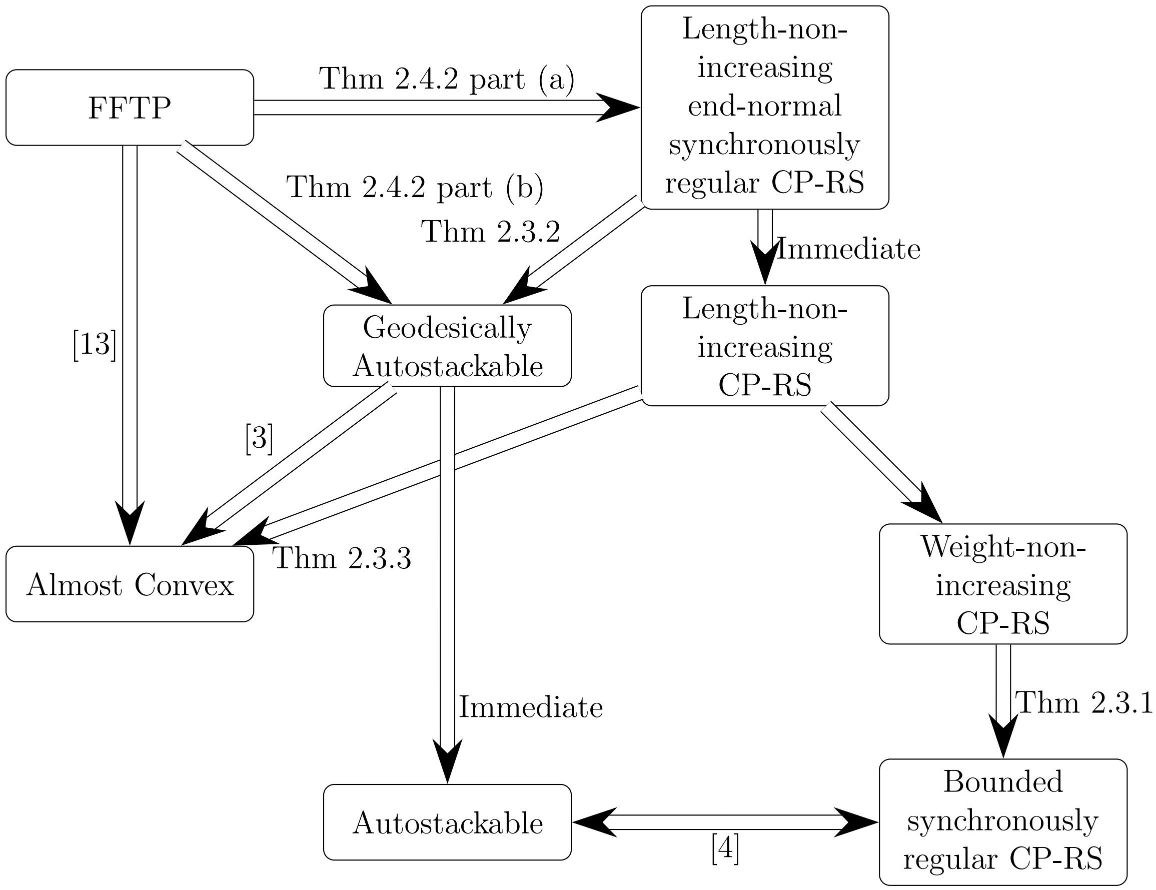

Figure 2.1: A flowchart of related results from this and other papers

- $|u| = |v|$ *and for each* $s \in A^*$ *such that* $u =_{A^*} u's$ *and* $v =_{A^*} v's$ *for some* $u', v' \in A^*$, *we have* $last(u') \preceq last(v')$.

The short reverse lexicographic ordering given above is essentially the shortlex ordering reading right-to-left rather than left-to-right, and we use it in this chapter rather than shortlex to make certain proofs and constructions simpler.

2.2.1 Almost Convexity

Definition 2.2.2. *[7] A group G is* almost convex *with respect to a finite generating set A if there is a constant k such that for all $n \in \mathbb{N}$ and g, h in the sphere $S(n)$ (in $\Gamma_{G,A}$ centered at 1_G) with $d_{G,A}(g, h) \leq 2$, there exists a path inside the ball $B(n)$ (in*

$\Gamma_{G,A}$ *centered at* 1_G*) of length at most* k *from* g *to* h*.*

Almost convexity was introduced by Cannon [7]. Thiel showed that almost convexity is dependent on generating set [30]. Elder showed that given a pair (G, A) with FFTP, G is almost convex with respect to A [13].

2.3 Weight- and length-non-increasing CP-RSs

In this section, we prove that every weight non-increasing regular CP-RS gives a bounded regular CP-RS, which yields an autostackable structure using Proposition 1.1.3, and that additional restrictions show geodesic autostackability and almost convexity.

Theorem 2.3.1. *Suppose* $G = \langle A \rangle$ *has a weight non-increasing synchronously regular CP-RS* R*. Then* G *has an autostackable structure which has the same normal forms as* R*.*

Proof. Let R be a weight non-increasing synchronously regular CP-RS for $G = \langle A \rangle$ with normal form set $\mathcal{N}$. Let

$$R' = \{(u, v) \mid (u, v) \in R \text{ and } u \text{ is minimally reducible}\}$$

Then R' is terminating, as an infinite sequence of rewritings in R' would necessarily be an infinite sequence of rewritings in R. Further, R' is weight non-increasing since it is a subset of a weight non-increasing CP-RS. Further, $R' = R \cap ((\mathcal{N} \cdot A) \times A^*)$, an intersection of synchronously regular languages, hence R' is regular. We have $R' \subseteq R$, so for all $(u, v) \in R'$, we have $u =_G v$, hence G is a quotient of $Mon\langle A \mid R' \rangle$. For any word u which can be reduced with R, we see that $u =_{A^*} u_1 u_2$ for some $u_1, u_2 \in A^*$ with $u_1 \in \mathcal{N}$ and $u_1 \text{first}(u_2) \notin \mathcal{N}$. But $u_1 \text{first}(u_2)$ can be rewritten over R, hence

$u_1\text{first}(u_2)$ is the left side of a pair in R' and is reducible in R'. Therefore, the set of irreducible words over R' is the same as the set of irreducible words over R, so R' has exactly one irreducible word over A for each element of G; this, along with the fact that G is a quotient of $Mon\langle A \mid R'\rangle$, gives that $G = Mon\langle A \mid R'\rangle$, and that the normal forms of G over R' are the same as the normal forms over R. Thus R' is a weight non-increasing synchronously regular CP-RS for G with normal form set $\mathcal{N}$.

Let M be an FSA accepting the padded extension of R', and let k be the number of states in M. Let $N = 2k * \max\{wt(a) \mid a \in A\}$. Let

$$S_1 = \{(u, \text{nf}_{R'}(u)) \mid u \text{ is minimally reducible and } wt(u) < N\}$$

and let

$$S_2 = \{(u_1u_2l, u_1\text{nf}_{R'}(u_2lv_2^{-1})v_2) \mid (u_1u_2l, v_1v_2) \in R', wt(u_2) < N, \\ wt(u_2l) \geq N, \text{ and } v_1 = v(\min\{|u_1|, |v| - 1\})\}$$

(A visual representation of the rules in S_2 in the Cayley graph $\Gamma_{G,A}$ is given in Figure 2.2.) Then let $S = S_1 \cup S_2$. We claim that S is a bounded regular convergent prefix-rewriting system for G with normal forms $\mathcal{N}$. For this, we must prove five things: that $G = Mon\langle A \mid \{u = v \mid (u, v) \in S\}\rangle$; that S is bounded; that S is synchronously regular; that S is terminating; and that $\mathcal{N}$ is the set of irreducible words over S.

We first prove that

$$G = Mon\langle A \mid \{u = v \mid (u, v) \in S\}\rangle.$$

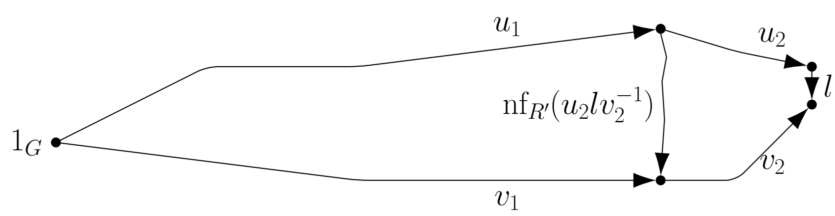

Figure 2.2: For pairs in S_2, the left word follows the top path from 1_G to u_1u_2l, while the right word reroutes via the $\mathrm{nf}_{R'}(u_2lv_2^{-1})$ bridge.

We prove this by showing that each relation given by S is a relation in G, and that each relation in R' (which we know is a prefix-rewriting system for G) is a result of relations in S.

We first consider relations $uw =_S vw$ arising from pairs (u, v) in S_1: these are relations in G, since $uw =_G \mathrm{nf}_{R'}(u)w =_{A^*} vw$. Relations arising from pairs (u, v) in S_2 are also relations in G, as

$$\begin{aligned} uw &=_{A^*} u_1u_2lw \\ &=_G u_1u_2lv_2^{-1}v_2w \\ &=_{R'} u_1\mathrm{nf}_{R'}(u_2lv_2^{-1})v_2w \\ &=_{A^*} vw \end{aligned}$$

Since R' is a prefix-rewriting system for G, we see that the third equality is also true in G, so $u =_G v$. Thus, equality of two words over S implies equality of the same two words in G.

Now, let (u, v) be a pair from R'. Our next goal is to show that $u =_S v$. We proceed by induction on $wt(u)$. Note that the set $\{wt(u) \mid u \in A^*\}$ is the set of non-negative integer combinations of finitely many positive values, hence is well-founded, so induction is a viable proof technique here.

For our base case, suppose that $wt(u) < N$; then S_1 contains the pair $(u, \mathrm{nf}_{R'}(u))$.

We also see that $wt(v) \leq wt(u) < N$ because R' is weight non-increasing, so either $v =_{A^*} \mathrm{nf}_{R'}(v)$ or S_1 contains the pair $(v, \mathrm{nf}_{R'}(v))$. Since $u =_G v$, we have that $\mathrm{nf}_{R'}(u) =_{A^*} \mathrm{nf}_{R'}(v)$, and we have both $u =_S \mathrm{nf}_{R'}(u)$ and $v =_S \mathrm{nf}_{R'}(u)$, so $u =_S v$.

Now suppose that $wt(u) = n \geq N$, and that whenever $u' =_{R'} v'$ with $wt(u') < n$ and $wt(v') < n$, we have that $u' =_S v'$. Then let $l = \mathrm{last}(u)$, let $u_2 l$ be the shortest suffix of u with weight at least N, and let u_1 be the prefix of u such that $u = u_1 u_2 l$. If $|v| \geq |u_1|$, let $v_1 = v(|u_1|)$, and v_2 be the suffix of v such that $v =_{A^*} v_1 v_2$; otherwise, let $v_1 = v(|v| - 1)$ and let v_2 be the suffix of v with length 1. We need to show that $u_1 u_2 l =_S v_1 v_2$. We see that the pair $(u_1 u_2 l, u_1 \mathrm{nf}_{R'}(u_2 l v_2^{-1}) v_2)$ is an element of S_2, and that $\mathrm{nf}_{R'}(u_1 \mathrm{nf}_{R'}(u_2 l v_2^{-1})) =_{A^*} \mathrm{nf}_{R'}(v_1)$.

Consider $wt(\mathrm{nf}_{R'}(u_2 l v_2^{-1}))$. Recall that M is the finite state automaton accepting all padded pairs from R'. Also recall from Definition 1.2.1 that $\widehat{\delta}(q, \lambda) = q$ for all states q, and $\widehat{\delta}(q, aw) = \widehat{\delta}(\delta(q, a), w)$; that is, $\widehat{\delta}(q, w)$ is the state of M after starting at the state q and reading the word w. Suppose that $|v_1| = |u_1|$; then we see that $q = \widehat{\delta}(q_0, (u_1, v_1))$ is a state in M from which we can reach a state in P — in particular, $\widehat{\delta}(q, (u_2 l, v_2)^p) \in P$ since $(u_1 u_2 l, v_1 v_2)^p$ is accepted by M. But since M has only k states, and we can replace any path in M by a path which does not repeat states, we can replace this path by one of length at most $k - 1$ (potentially touring through every state, and using $k - 1$ edges in total). This shorter path corresponds to a pair of words (u_2'', v_2'') such that $(u_1 u_2'', v_1 v_2'') \in R'$ and $|u_2''| \leq k - 1$ and $|v_2''| \leq k - 1$. Then we see that $u_1 u_2'' =_G v_1 v_2''$, hence $u_2'' v_2''^{-1} =_G u_1^{-1} v_1$. We also have $u_1 u_2 l =_G v_1 v_2$, hence $u_2 l v_2^{-1} =_G u_1^{-1} v_1$. Combining these gives us $u_2'' v_2''^{-1} =_G u_2 l v_2^{-1}$, so $\left|\mathrm{nf}_{R'}(u_2 l v_2^{-1})\right| \leq \left|u_2'' v_2''^{-1}\right| \leq 2k - 2$. On the other hand, suppose that $|v_1| < |u_1|$. Then $|v_2| = 1$, and $\widehat{\delta}(q_0, (u_1, v)^p)$ is a state in M from which we can reach a state in P. This again gives a path in M of length at most $k - 1$, corresponding to a path in $\Gamma_{G,A}$ from $\overline{u_1}$ to $\overline{v}$ of length at most $2k - 2$, which can be extended with a single edge to a path from $\overline{u_1}$ to $\overline{v_1}$ of length at most $2k-1$. Either way, we have

$$\left|\mathrm{nf}_{R'}(u_2 l v_2^{-1})\right| \leq 2k-1 \tag{2.1}$$

This gives us that $wt(\mathrm{nf}_{R'}(u_2 l v_2^{-1}) \leq (2k-1) * \max\{wt(a) \mid a \in A\} < N$. We can now use our induction hypothesis: $wt(u_1 \mathrm{nf}_{R'}(u_2 l v_2^{-1})) < wt(u_1 u_2 l) = n$, so $u_1 \mathrm{nf}_{R'}(u_2 l v_2^{-1}) =_S \mathrm{nf}_{R'}(u_1 \mathrm{nf}_{R'}(u_2 l v_2^{-1}))$. Similarly, $wt(v_1 v_2) \leq n$, and since we have $|v_2| \geq 1$, we must have $wt(v_1) < n$, so by our induction hypothesis $v_1 =_S nf_{R'}(v_1)$. Because $v_1 =_G u_1 \mathrm{nf}_{R'}(u_2 l v_2^{-1})$, we have $\mathrm{nf}_{R'}(v_1) =_{A^*} \mathrm{nf}_{R'}(u_1 \mathrm{nf}_{R'}(u_2 l v_2^{-1}))$. Stringing this all together, we have

$$\begin{aligned} u &=_{A^*} u_1 u_2 l \\ &=_S u_1 \mathrm{nf}_{R'}(u_2 l v_2^{-1}) v_2 \\ &=_S \mathrm{nf}_{R'}(u_1 \mathrm{nf}_{R'}(u_2 l v_2^{-1})) v_2 \\ &=_{A^*} \mathrm{nf}_{R'}(v_1) v_2 \\ &=_S v_1 v_2 \\ &=_{A^*} v \end{aligned}$$

Thus, equality of two words over R' implies equality of the same two words over S, so equality of two words in G implies equality over S. Combining this with the fact that equality over S implies equality in G (proved above), we see that S is a prefix-rewriting system for G.

We now consider boundedness. Each rule in S_1 rewrites a word of weight at most N (and thus length at most $\frac{N}{\min\{wt(a) | a \in A\}}$) to another word of weight at most N, hence each rule in this subset of S rewrites a substring of bounded length to a substring of bounded length.

Now, consider a pair $(u_1u_2l, u_1\mathrm{nf}_{R'}(u_2lv_2^{-1})v_2)$ from S_2. We note that since $wt(u_2) < N$, we have $|u_2| < \frac{N}{\min\{wt(a)|a\in A\}}$, so $|u_2l|$ is bounded. By equation 2.1 above, $\left|\mathrm{nf}_{R'}(u_2lv_2^{-1})\right| \leq 2k-1$. Suppose for sake of contradiction that $|v_2| > |u_2l|+1+(k-1)$; then let $v_3 = v_2(|u_2l|)$, and let v_4 be the suffix of v_2 such that $v_2 = v_3v_4$. Then the path in M starting at $\widehat{\delta}(q_0, (u, v_1v_3))$ and labeled by $(\lambda, v_4)^p$ has length at least k, and must repeat a state. This gives us a decomposition $v_4 = pzs$ with $|z| > 0$ such that $(u, v_1v_3pz^ns)^p$ is accepted by M for all n, which is impossible since, for sufficiently large n, $v_1v_3pz^ns$ has weight larger than $wt(u)$. Thus, $|v_2| \leq |u_2l| + k$. Thus, each rule in this subset rewrites a substring of bounded length to a substring of bounded length, so S is bounded. Let $b = \max\{|u_2l|, \left|\mathrm{nf}_{R'}(u_2lv_2^{-1})v_2\right|\}$.

We next consider regularity. Let $D = \{(a, a) \mid a \in A\}$. Let

$$L_{(u_2l,v_2)} = ((R')^p \cap ((A \times A)^* \cdot (\{u_2l\} \times \{v_2\})^p)) / (\{u_2l\} \times \{v_2\})^p$$

for each $l \in A$ and $u_2l, v_2 \in A^{\leq b}$. We note that $L_{(u_2l,v_2)}$ is the set $\{(u_1, v_1) \mid (u_1u_2l, v_1v_2) \in R'\}$. We then let $P_{(u_2l,v_2)} = \pi_1(L_{(u_2l,v_2)})$ where π_1 is the projection map onto the first coordinate. Then $P_{(u_2l,v_2)}$ is the set of all u_1 such that there exists some v_1 with $(u_1u_2l, v_1v_2) \in R'$. Then S_2 is

$$\bigcup_{u_2l\in A^{\leq b}} \bigcup_{v_2\in A^{\leq b}} \left((P_{(u_2l,v_2)} \times P_{(u_2l,v_2)}) \cap D^*\right)^p \cdot (\{u_2l\} \times \{\mathrm{nf}_{R'}(u_2lv_2^{-1})v_2\})^p$$

Because synchronously regular languages are closed under finite unions, finite intersections, products, projections, quotients, Kleene stars, and concatenation on the right by finite languages, and S_2 is built from regular languages (namely R', D, and several finite languages) using finitely many of these operations, S_2 is synchronously regular. Since S_1 is finite, S_1 is also synchronously regular. Thus, S is a union of two synchronously regular languages, hence S is synchronously regular.

We now consider termination. We define a strict partial order $\prec$ on directed edges in $\Gamma_{G,A}$ as follows: For an edge e, define $w(e) = \mathrm{nf}_{R'}(\iota(e))\mathrm{label}(e)$. Then $e' \prec e$ if $wt(w(e')) < wt(w(e))$, or $wt(w(e')) = wt(w(e))$ and there exists some rewriting sequence in R'

$$w(e) = w_0 \to w_1 \to \cdots \to w_n = w(e')$$

Because there is no infinite chain of rewritings over R', any infinite descending chain $e_0 \succ e_1 \succ e_2 \succ \cdots$ must have some i with $wt(e_i) < wt(e_0)$; repeating this argument gives a subchain $e_0 = e'_0 \succ e'_1 \succ e'_2 \succ \cdots$ with $wt(e'_i) < wt(e'_{i-1})$, contradicting the well-foundedness of the possible weights of words in A^*. Thus, $\prec$ admits no infinite descending chains, hence $\prec$ is well-founded.

Now, we extend this order on edges to an order on words in A^* as follows: Assign to each word u the set of directed edges E_u contained in the path in $\Gamma_{G,A}$ starting at 1_G and labeled by u. Then $u' < u$ if there exists some $e \in E_u$ such that for all $e' \in E_{u'}$, $e' \prec e$. The relation $<$ is transitive, since whenever we have $u < v$ and $v < w$, there exist $e_v \in E_v$ and $e_w \in E_w$ with the property that for all $e' \in E_u$, $e' \prec e_v \prec e_w$, hence $e' \prec e_w$. The relation $<$ is also antisymmetric, since $u < v$ and $v < u$ would give the existence of edges $e_u \in E_u$ and $e_v \in E_v$ with $e_u \prec e_v$ and $e_v \prec e_u$, contradicting antisymmetry of $\prec$. Further, $<$ is irreflexive, since for any $e_u \in E_u$, we have $e_u \not\prec e_u$, hence $u \not< u$. Thus, $<$ is a strict partial order on A^*. Further, any infinite descending chain $u_0 > u_1 > u_2 > \cdots$ would give an infinite descending chain $e_0 \succ e_1 \succ e_2 \succ \cdots$ with $e_i \in E_{u_i}$, violating well-foundedness of $\prec$; thus, $<$ is well-founded.

We now show that S decreases $<$; that is, for every $(u, v) \in S$, we have $v < u$. We have two cases to consider:

Suppose $(u, \mathrm{nf}_{R'}(u)) \in S_1$. Let $l = \mathrm{last}(u)$ and u_1 be the prefix of u such that

$u_1 l = u$. Then let e be the edge in $\Gamma_{G,A}$ starting at u_1 and labeled by l. Then we have $w(e) =_G u$, hence $wt(w(e)) \geq wt(\mathrm{nf}_{R'}(u))$. This gives that for all $e' \in E_{\mathrm{nf}_{R'}(u)}$ we have $wt(w(e')) < wt(w(e))$, with possibly one exception: the final edge of the path in $\Gamma_{G,A}$ starting at 1_G that is labeled by $\mathrm{nf}_{R'}(u)$. As such, $e' \prec e$ for all but possibly this final edge. Now, let e' be the final edge of the path in $\Gamma_{G,A}$ starting at 1_G that is labeled by $\mathrm{nf}_{R'}(u)$. We notice that because normal forms of R' are closed under prefixes, we have $w(e') = \mathrm{nf}_{R'}(u)$. Then there is some rewriting sequence in R' taking $w(e)$ to $\mathrm{nf}_{R'}(w(e)) =_{A^*} \mathrm{nf}_{R'}(u) =_{A^*} w(e')$, so $e' \prec e$. Thus, all edges in $E_{\mathrm{nf}_{R'}(u)}$ are smaller than $e \in E_u$, so $\mathrm{nf}_{R'}(u) < u$, as desired.

Now, suppose $(u_1u_2l, u_1\mathrm{nf}_{R'}(u_2lv_2^{-1})v_2) \in S_2$. Let e be the edge labeled by l starting at u_1u_2, and let $e' \in E_{u_1\mathrm{nf}_{R'}(u_2lv_2^{-1})v_2}$. Note that $w(e) = u_1u_2l$, since $\mathrm{nf}_{R'}(\iota(e)) =_{A^*} \mathrm{nf}_{R'}(\overline{u_1u_2}) =_{A^*} u_1u_2$ and $\mathrm{label}(e) = l$. We have four cases, depending on the location of e' in the path starting at 1_G and labeled by $u_1\mathrm{nf}_{R'}(u_2lv_2^{-1})v_2$:

Case 1: Suppose e' is an edge in the subpath starting at 1_G and labeled by u_1. Then $w(e')$ is a prefix of u_1 since u_1 is a normal form of R' and normal forms of R' are closed under prefixes. Thus $w(e')$ is a proper prefix of $w(e)$, so $wt(w(e')) < wt(w(e))$, showing that $e' \prec e$.

Case 2: Suppose e' is an edge in the subpath starting at u_1 and labeled by $\mathrm{nf}_{R'}(u_2lv_2^{-1})$. Then

$$\begin{aligned} wt(w(e')) &\leq wt(u_1) + 2k * \max\{wt(a) \mid a \in A\} \\ &= wt(u_1) + N \\ &< wt(u_1) + wt(u_2l) \\ &= wt(w(e)), \end{aligned}$$

so $e' \prec e$.

Case 3: Suppose e' is an edge in the path starting at $u_1\mathrm{nf}_{R'}(u_2lv_2^{-1})$ and labeled by v_2, but is not the last edge of this path. Then $wt(w(e')) < wt(v_1v_2) \leq wt(u) = wt(w(e))$, hence $e' \prec e$.

Case 4: Suppose e' is the final edge of the path starting at $u_1\mathrm{nf}_{R'}(u_2lv_2^{-1})$ and labeled by v_2. Then $wt(w(e')) \leq wt(v_1v_2) \leq wt(u) = wt(w(e))$. Moreover, there is a single rewriting from R' which rewrites $w(e) =_{A^*} u_1u_2l$ to v_1v_2. Let $v =_{A^*} v_1v_2$, $l' = \mathrm{last}(v)$ and v' be the prefix of v such that $v'l' =_{A^*} v$. Then there is a sequence of rewritings from R' taking v' to $\mathrm{nf}_{R'}(v')$. Appending this rewriting sequence to our rewriting from $w(e)$ to v_1v_2 gives a rewriting sequence from R' that starts at $w(e)$ and ends at $\mathrm{nf}_{R'}(v')\mathrm{last}(v) =_{A^*} w(e')$. Thus, $e' \prec e$.

In all cases, we have $e' \prec e$ (and, in particular, if R was length non-increasing, $d_{G,A}(1_G, \iota(e')) \leq d_{G,A}(1_G, \iota(e))$, which will be useful in the proof of Theorem 2.3.2); thus, $u_1\mathrm{nf}_{R'}(u_2lv_2^{-1})v_2 < u_1u_2l$, as desired. So S decreases a well-founded strict partial ordering, hence S is terminating.

Finally, we show that S has a set of unique normal forms, namely the normal forms over R'.

Suppose that w is reducible over R'. Then w has some maximal prefix u satisfying that u is irreducible over R'. Since u is a proper prefix, there exists some $l \in A$ such that ul is a prefix of w. Then ul is not irreducible over R', hence is a minimally reducible prefix of w over R'. Then there is some rule (ul, v) in R' which we can apply to w. Further, if $wt(ul) < N$, then $(ul, \mathrm{nf}_{R'}(ul))$ is a rule in S_1, so w is also reducible over S; and if $wt(ul) \geq N$, then ul is the left-hand side of a rule in S_2, so w is reducible over S. Either way, we have that w is reducible over S.

Alternatively, suppose that w is reducible over S. Then some prefix v of w is the left-hand side of a rule from S_1 or from S_2, so v is reducible over R'. As v is a prefix of w, this gives us that w is also reducible over R'.

Thus, the words which are reducible over R' are the same as the words which are reducible over S. This gives that the irreducible words over both rewriting systems are the same, so S has a set of unique normal forms, and $\mathcal{N}_S = \mathcal{N}_{R'} = \mathcal{N}_R$.

With all of the above, we see that S is a bounded regular convergent prefix-rewriting system for G. Applying Proposition 1.1.3, we see that G is autostackable with normal form set $\mathcal{N}$.

□

Note that, since any length non-increasing CP-RS is a weight non-increasing CP-RS with each generator having length 1, this theorem shows that any group with a length non-increasing CP-RS is autostackable. We now prove the following theorem as an extension to Theorem 2.3.1 when we have additional restrictions on R:

Theorem 2.3.2. *Suppose that $G = \langle A \rangle$ admits a length non-increasing end-normal synchronously regular CP-RS R. Then the autostackable structure for G constructed in the proof of Theorem 2.3.1 is a geodesically autostackable structure.*

Proof. Suppose $G = \langle A \rangle$ admits a length non-increasing end-normal synchronously regular CP-RS R. Let S be the bounded regular CP-RS for G constructed in the proof of Theorem 2.3.1, and let Φ be the flow function constructed from S as in the proof of [4, Theorem 5.3]. That is, for each group element $y \in G$ and each letter $a \in A$,

$$\Phi(e_{y,a}) = \begin{cases} e_{y,a} & \text{if } \mathrm{nf}_R(y)a \in \mathcal{N}_R \text{ or } \mathrm{last}(\mathrm{nf}_R(y)) = a^{-1} \\ p_{y,s^{-1}t} & \text{otherwise} \end{cases}$$

where $p_{y,s^{-1}t}$ is the path starting at the vertex corresponding to y and labeled by the word $s^{-1}t$, with $(\mathrm{nf}(y)a, y') \in S$, $\mathrm{nf}(y) =_{A^*} ws$, $y' =_{A^*} wt$, and the words s and t do not start with the same letter. Then each directed edge e in $\Gamma_{G,A}$ falls into one of the following cases:

Case 1: Suppose $\mathrm{nf}_S(\iota(e))\mathrm{label}(e) \in \mathcal{N}_R$. In this case, $\Phi(e) = e$.

Case 2: Suppose $|\mathrm{nf}_S(\iota(e))\mathrm{label}(e)| \leq k+1$ and $\mathrm{nf}_S(\iota(e))\mathrm{label}(e) \notin \mathcal{N}_R$. In this case, $\mathrm{label}(\Phi(e)) = s^{-1}s'$ for some suffix s of $\mathrm{nf}_S(\iota(e))$ and some suffix s' of $\mathrm{nf}_S(\iota(e)\mathrm{label}(e))$. Recall from Definition 2.1.2 that $\alpha(e) = \frac{1}{2}\left(d_{G,A}(1_G, \iota(e)) + d_{G,A}(1_G, \tau(e))\right)$. For each edge e' in the path $\Phi(e)$ except for possibly the final edge, we have that e' is an edge in the path starting at 1_G and labeled by a geodesic with length at most $|\mathrm{nf}_S(\iota(e))|$, hence $\alpha(e') < |\mathrm{nf}_S(\iota(e))| \leq \alpha(e)$. For the final edge e_f of the path $\Phi(e)$, we have that e_f is a portion of the normal form of $\iota(e)\mathrm{label}(e)$, so $\Phi(e_f) = e_f$.

Case 3: Suppose $|\mathrm{nf}_S(\iota(e))\mathrm{label}(e)| > k+1$ and $\mathrm{nf}_S(\iota(e))\mathrm{label}(e) \notin \mathcal{N}_R$. In this case, $\mathrm{label}(\Phi(e))$ is the word w obtained by freely reducing $u_2^{-1}\mathrm{nf}_S(u_2lv_2^{-1})v_2$ for some u_2, v_2, and l as in the definition of S_2. Each edge in the path starting at $\iota(e)$ and labeled by w is also an edge in the path starting at $\iota(e)$ and labeled by $u_2^{-1}\mathrm{nf}_S(u_2lv_2^{-1})v_2$. As we proved in the termination subsection of the proof of Theorem 2.3.1, each edge e' in the path starting at $\iota(e)$ and labeled by $u_2^{-1}\mathrm{nf}_S(u_2lv_2^{-1})v_2$ has $d_{G,A}(1_G, \iota(e')) \leq d_{G,A}(1_G, \iota(e))$ and $d_{G,A}(1_G, \tau(e')) \leq d_{G,A}(1_G, \tau(e))$, with equality only at possibly the final edge e_f of this path. From this, we see that $\alpha(e') < \alpha(e)$ except when $e' = e_f$. But e_f is the edge labeled by $\mathrm{last}(\mathrm{nf}_S(\iota(e)\mathrm{label}(e))$ with endpoint $\iota(e)\mathrm{label}(e)$, so e_f is a portion of the normal form of $\iota(e)\mathrm{label}(e)$, giving $\Phi(e_f) = e_f$.

In all three cases, we have that whenever e' is an edge in the path $\Phi(e)$, we have either $\alpha(e') < \alpha(e)$ or $\Phi(e') = e'$. Thus, Φ is the flow function for a geodesically autostackable structure. □

We can also use the proof of Theorem 2.3.1 to prove almost convexity in the following theorem. The proof is similar to those found in [18, Theorem B] and [3, Theorem 4.4] that groups with geodesic finite complete rewriting systems and geodesically stackable groups (respectively) are almost convex. We include the details of the proof for the sake of completeness.

Theorem 2.3.3. *Suppose $G = \langle A \rangle$ has a length non-increasing synchronously regular CP-RS. Then G is almost convex with respect to A.*

Proof. Suppose $G = \langle A \rangle$ has a length non-increasing synchronously regular CP-RS. Define a bounded regular CP-RS S for $G = \langle A \rangle$ as in the proof of Theorem 2.3.1, and let $g, h \in G$ with $g, h \in S(n)$ and $d_{G,A}(g,h) \leq 2$.

In the case that $d_{G,A}(g,h) = 1$, let $a_0 \in A$ such that $ga_0 =_G h$. Then $\text{nf}_S(g) = w_0u_0$ with $(w_0u_0a_0, w_0v_0a_1)$ being a rule in S for some $w_0, u_0, v_0 \in A^*$, and $a_1 \in A$. As a consequence of the proof of termination in the proof of Theorem 2.3.1, the path starting at $\overline{w_0}$ and labeled by v_0 lies entirely within $B(n)$. If $a_1 = \text{last}(\text{nf}_S(h))$, then $e_{ha_1^{-1},a_1}$ lies in $B(n)$, and we see that the path labeled by $u_0^{-1}v_0a_1$ starting at g ends at h, has length at most $2k + 2 + 2k + 2 + 1 = 4k + 5$, and lies entirely within $B(n)$. Otherwise, we can repeat this process: $w_0v_0a_1 =_G \text{nf}_S(w_0v_0)a_1 =_{A^*} w_1u_1a_1$, with some rule $(w_1u_1a_1, w_1v_2a_2)$ in S. In this way, we get a chain of equalities and rewritings

$$w_0u_0a_0 \to_S w_0v_0a_1 =_G w_1u_1a_1 \to_S \cdots \to_S w_jv_ja_{j+1}$$

where $a_{j+1} = \text{last}(\text{nf}_S(h))$, and $|w_iu_i| \leq n$ and $|w_iv_i| \leq n$ for all i. This gives a path from g to h labeled by $u_0^{-1}v_0u_1^{-1}v_1 \ldots u_j^{-1}v_ja_{j+1}$ which lies entirely within $B(n)$. Each $u_i^{-1}v_i$ piece of the path has length at most $4k + 4$ (from the proof of boundedness in Theorem 2.3.1). We cannot repeat any a_i because having $a_i = a_k$ for some $k > i$ would give a loop of rewritings $w_iu_ia_i \to_S \cdots \to_S w_ku_ka_k$, where both w_iu_i and w_ku_k are the unique normal form of ha_i^{-1}. Thus, there are at most $|A|$ pieces in our path of the form $u_i^{-1}v_i$, plus the final edge a_{j+1}. In this case, we have a path in $B(n)$ from g to h of length at most $(4k+4)\,|A|+1$.

Now, we consider the case that $d(g,h)=2$. Then $h=gab$ for some $a,b\in A$. There are three subcases. If $d(1_G,ga)=n-1$, we have a path of length 2 from g to h lying within $B(n)$, namely the path starting at g and labeled by ab. If $d(1_G,ga)=n$, we can apply the distance 1 case twice, giving a path of length at most $2(4k+4)\,|A|$ from g to h lying within $B(n)$. This leaves the case where $d(1_G,ga)=n+1$. Let $c=\mathrm{last}(\mathrm{nf}_S(ga))$, and $g'=gac^{-1}$. Note that $g'\in B(n)$ because $|\mathrm{nf}_S(ga)|=n+1$, and g' is represented by the prefix of $\mathrm{nf}_S(ga)$ with length n. We now need to provide a path in $B(n)$ from g to g' of bounded length, and can repeat the process to make a path from g' to h. If $g=g'$, then the path of length 0 from g to g' lies within $B(n)$, so suppose instead that $g\neq g'$. Then $\mathrm{nf}_S(g)a$ is not itself a normal form, hence $\mathrm{nf}_S(g)a=w_0u_0a_0$; this rewrites to $w_0v_0a_1$, and again v_0 lies entirely within $B(n)$, by the same argument as the distance 1 case. Again, we have a chain of equalities and rewritings

$$w_0u_0a_0\to w_0v_0a_1=_G w_1u_1a_1\to\cdots w_jv_ja_{j+1}$$

where $a_{j+1}=\mathrm{last}(\mathrm{nf}_S(g'))$, and each $w_i,u_i,v_i\in B(n)$. This again gives a path of length at most $(4k+4)\,|A|$ from g to g' lying within $B(n)$. Repeating the process for a path from g' to h gives a path from g to h of length at most $2(4k+4)\,|A|$.

Thus, we have a path of length at most $2(4k+4)\,|A|$ from g to h in $B(n)$ whenever $g,h\in S(n)$ and $d_{G,A}(g,h)\leq 2$, so G is almost convex with respect to A. □

2.4 Proof of Theorem D

In this section, we produce a length non-increasing regular CP-RS with short reverse lexicographic normal forms for a pair (G,A) with FFTP. We begin with a lemma that will allow us to handle geodesics which are not short reverse lexicographic normal forms:

Lemma 2.4.1. *Suppose the pair (G, A) has FFTP with fellow traveler constant k, A is totally ordered, and $u \in A^*$ is a word representing $g \in G$ which is not a short reverse lexicographic normal form. Then there exists some word v with $v =_G u$, $v <_{srev} u$, and u and v $2k$-fellow travel.*

Proof. Suppose (G, A) has FFTP with fellow traveler constant k, and $u \in A^*$ is a word representing $g \in G$ which is not a short reverse lexicographic normal form. If u is not geodesic, we take v to be any witness of u; then $v =_G u$, $|v| < |u|$ so $v <_{srev} u$, and u and v k-fellow travel, hence also $2k$-fellow travel.

Now, suppose u is geodesic. Then let w be the short reverse lexicographic normal form of g, and u_2 the longest common suffix of u and w, so $u = u_1 u_2$ and $w = w_1 l u_2$ for some $w_1 \in A^*$ and $l \in A$. Then $u_1 l^{-1}$ is not a geodesic, having length $|u_1|+1$ while a geodesic representative of the same element has length $|u_1| - 1$. Because (G, A) has FFTP, there exists some word v' with $|v'| < |u_1| + 1$, $v' =_G u_1 l^{-1}$, and v' and $u_1 l^{-1}$ k-fellow travel. If $|v'| = |u_1| - 1$, we take $v'' = v'$; otherwise, $|v'| = |u_1|$, so there exists some v'' with $|v''| = |u_1| - 1$, $v'' =_G v'$, and v'' and v' k-fellow travel. Now, let $v = v'' l u_2$. Then v is short reverse lexicographically smaller than u, since the two have the same length, v has a longer common suffix with w than u does with w, and w is short reverse lexicographically smaller than u. Further, u and v $2k$-fellow travel, since u_1 and $v'' l$ $2k$-fellow travel and u and v extend these by the same suffix. □

The proof of Lemma 2.4.1 is illustrated in Figure 2.3. We now prove the main theorem.

Theorem 2.4.2 (Main Theorem)**.** *Suppose the pair (G, A) has the falsification by fellow traveler property. Then*

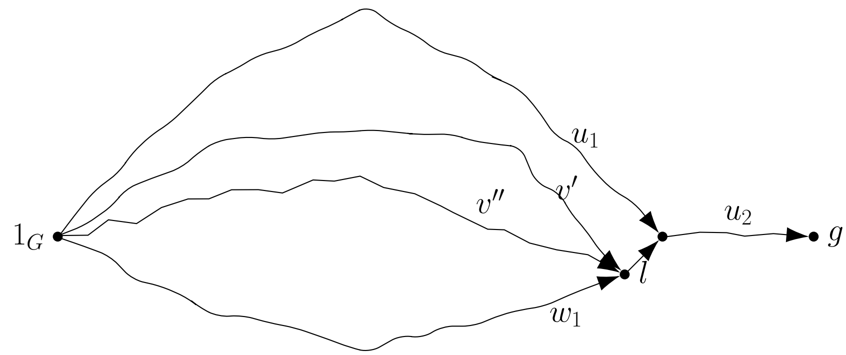

Figure 2.3: While u_1 and $w_1 l$ might be quite far apart, we see that u_1 and v' k-fellow travel, and similarly v' and v'' k-fellow travel.

(a) G admits a length non-increasing end-normal synchronously regular CP-RS R; and

(b) G is geodesically autostackable.

Proof. Suppose (G, A) has FFTP with fellow traveler constant k. Let $\prec$ be a total ordering on A, and let $\mathcal{N}$ be the set of short reverse lexicographic normal forms of G. Let

$$L = \{(u, v) \mid u =_G v, v <_{srev} u, \text{ and } u \text{ and } v \text{ } 4k\text{-fellow travel}\}.$$

We create an FSA M accepting L^p as follows:

The alphabet of M is $A \times (A \cup \{\$\})$. For each element $g \in G$ with $|g| \leq 4k$, we have two states: one labeled by $\text{nf}_{srev}(g)$, and one labeled by $\text{nf}_{srev}(g)\$$. We also have one fail state F and one additional state λ'. The accept states of M are λ' and $\lambda\$$, and the initial state is λ. We define the transition function δ with three parts:

$$\text{diff}(s, (a, b)) = \begin{cases} \text{nf}_{srev}(a^{-1}gb) & \text{if } s \text{ is labeled by } \text{nf}_{srev}(g), \\ & \quad \text{by } \text{nf}_{srev}(g)\$, \text{ or by } \text{nf}_{srev}(g)' \text{ for some } g \in G \\ F & \text{if } s = F \end{cases}$$

$$\text{vterm}(s,(a,b)) = \begin{cases} F & \text{if } s = F \\ \$ & \text{if } s \neq F \text{ and } b = \$ \\ F & \text{if } s = \text{nf}_{srev}(g)\$ \text{ for some } g \in G \text{ and } b \neq \$ \\ \lambda & \text{otherwise} \end{cases}$$

$$\text{srev}(s,(a,b)) = \begin{cases} 1 & \text{if } a, b \in A \text{ and } b \prec a \\ 1 & \text{if } s = \lambda' \text{ and } a = b \\ 0 & \text{otherwise} \end{cases}$$

(Note that in the first case for $\text{diff}(s,(a,b))$, we only have $\text{nf}_{srve}(g)'$ when $g =_G 1$.) Now, we define our transition function by

$$\delta(s,(a,b)) = \begin{cases} F & \text{if } \text{diff}(s,(a,b)) = F \\ & \quad \text{or } \text{vterm}(s,(a,b)) = F \\ & \quad \text{or } |\text{diff}(s,(a,b))| > 4k \\ \lambda' & \text{if } \text{diff}(s,(a,b)) = \lambda, \\ & \quad \text{vterm}(s,(a,b)) = \lambda, \\ & \quad \text{and } \text{srev}(s,(a,b)) = 1 \\ \text{diff}(s,(a,b))\text{vterm}(s,(a,b)) & \text{otherwise} \end{cases}$$

The function $\text{diff}(s,(a,b))$ tracks the word difference between the two input words u and v as long as these words $4k$-fellow travel, u does not terminate before v, and v does not have a padding letter between letters from A. The function $\text{vterm}(s,(a,b))$ tracks whether v has terminated and whether v has a padding letter between letters from A. The function $\text{srev}(s,(a,b))$ tracks whether the most recent pair of non-identical letters was in decreasing order.

We now prove that the language accepted by M is L^p. We first notice that vterm prevents M from accepting any words which are not padded pairs, so we can restrict our proof to deal only with padded pairs.

Suppose $(u,v)^p$ is accepted by M. Recall that $\widehat{\delta}(s,w)$ is defined to be the state of M after starting at a state s and reading a word w. Then $\widehat{\delta}(\lambda,(u,v)^p)$ is either $\lambda\$$ or λ'. In particular, this requires $u^{-1}v =_G 1$, since diff (which tracks the word difference between prefixes of u and of v) ended at λ. Further, reading $(u,v)^p$ avoids landing at F, so $|\text{diff}| \leq 4k$ at each step; thus, u and v must $4k$-fellow travel. We now consider the role of vterm: either this function ended at \$, or vterm ended at λ and srev ended at 1. In the first case, $(u,v)^p$ reached a padding letter in the second coordinate, meaning that $|v| < |u|$, so $v <_{srev} u$. In the second case, $(u,v)^p$ had no padding symbols, so $|u| = |v|$, but srev ended at 1. Then either $\text{last}(v) \prec \text{last}(u)$, so that $v <_{srev} u$, or the state before reading the last letter of $(u,v)^p$ was also λ' and $\text{last}(u) = \text{last}(v)$. Continuing this reasoning, we see that $u =_{A^*} u_1u_2$, $v =_{A^*} v_1u_2$, and $\text{last}(v_1) \prec \text{last}(u_1)$ for some words $u_1, u_2, v_1 \in A^*$. In this case, we again have that $v <_{srev} u$. Thus, the language accepted by M is a subset of L^p.

Now, suppose that $(u,v)^p$ satisfies $v <_{srev} u$, $u =_G v$, and u and v $4k$-fellow travel. Then starting at λ and reading $(u,v)^p$, $|\text{diff}| \leq 4k$ at each step, since u and v $4k$-fellow travel and vterm is never F because $(u,v)^p$ is a padded pair, so we never reach the state F. Since $u =_G v$, $\widehat{\delta}(\lambda,(u,v)^p)$ must be λ, $\lambda\$$, or λ'. If $|v| < |u|$, then we have vterm reaches \$ after v ends, so we must end at $\lambda\$$. Otherwise, since $v <_{srev} u$, we must have that v is reverse lexicographically smaller than u. Thus, srev ends at 1, meaning that we end at λ'. In either case, M accepts $(u,v)^p$, so L^p is a subset of the language accepted by M. Therefore, M accepts exactly the language L^p.

Now, we consider L. We first note that $u =_G v$ for all $(u, v) \in L$. Further, for each word u which is not a short reverse-lexicographic normal form, there exists some $v \in A^*$ with $v <_{srev} u$ and u and v $2k$-fellow travel by Lemma 2.4.1, so $(u, v) \in L$. The ordering $<_{srev}$ is well-founded, so each word which is not a short reverse-lexicographic normal form can be rewritten to its normal form using finitely many rules from L, so any two words which are equal in G can be rewritten to each other using finitely many relations of the form $u = v$ with $(u, v) \in L$. Thus, $G = \text{Mon}\langle A|L\rangle$, so L is a prefix-rewriting system for G. Because $<_{srev}$ is a well-founded ordering and for all $(u, v) \in L$ we have $v <_{srev} u$, L is terminating, and because each word $u \in A^*$ which is not the minimal representative of a group element under this strict total ordering is the left-hand side of a pair, we have that L has unique normal forms, hence L is convergent. Further, since L^p is the language accepted by M, L is synchronously regular. Finally, since L never increases length, we have that L is a length non-increasing synchronously regular convergent prefix-rewriting system for G with generating set A. By Theorem 2.3.1, G is autostackable.

To create a length non-increasing synchronously regular CP-RS which is also end-normal , we consider a sequence of languages. First, let $L' = L \cap (\mathcal{N}A \times A^*)^p$; that is, L' consists of pairs (u, v) such that every proper prefix of u is a short reverse lexicographic normal form. By a similar argument as in the proof of Theorem 2.3.1, L' is still a length non-increasing synchronously regular prefix-rewriting system for G with generating set A, with normal form set $\mathcal{N}$. Next, we define $L'_2 = L' \cap (A^* \times A^*\$\$)$, $L'_1 = L' \cap (A^* \times A^*\$)$, and $L'_0 = L' \cap (A^* \times A^*)$. That is, L'_i is the subset of L' consisting of pairs (u, v) where $|u| = |v| + i$. We next recursively define three languages for each letter in A, and two languages L_i for $i = 1, 2$: let

$$L_{2,a} = (L_2' \cap (A^* \times A^*a)^p) \setminus \left(\bigcup_{b \prec a} \pi_1(L_{2,b}) \times A^* \right)^p,$$

$$L_2 = \bigcup_{c \in A} \pi_1(L_{2,c}) \times A^*,$$

$$L_{1,a} = (L_1' \cap (A^* \times A^*a)^p) \setminus \left(L_2 \cup \bigcup_{b \prec a} (\pi_1(L_{1,b}) \times A^*) \right)^p,$$

$$L_1 = \bigcup_{c \in A} \pi_1(L_{1,c}) \times A^*, \text{ and}$$

$$L_{0,a} = (L_0' \cap (A^* \times A^*a)^p) \setminus \left(L_2 \cup L_1 \cup \bigcup_{b \prec a} (\pi_1(L_{0,b}) \times A^*) \right)^p.$$

That is, $L_{i,a}$ is the set of all $(u, v)^p \in L'$ such that $|u| = |v| + i$, $\text{last}(v) = a$, and there is no $v' \in A^*$ such that $(u, v') \in L'$ and $v' <_{srev} v$. Again, using a similar argument as at the start of the proof of Theorem 2.3.1, we have that $L'' = \cup_{i=0}^{2} \cup_{a \in A} L_{i,a}$ is a length non-increasing synchronously regular prefix-rewriting system for G with generating set A. Moreover, in Lemma 2.4.1 we showed that any non-normal form geodesic u $2k$-fellow travels a word v with $u =_G v$, $|v| = |u|$ and $\text{last}(v) = \text{last}(\text{nf}_R(u))$, so for any pair $(u, v) \in L''$ where u is a geodesic, we have that $\text{last}(v) = \text{last}(\text{nf}_R(u))$. In the case that u is not geodesic, since u is minimally reducible, we have that u must have length at most $|\text{nf}_R(u)| + 2$, hence it k-fellow travels some shorter word v_1, which then k-fellow travels some geodesic v_2, which then $2k$-fellow travels some geodesic v with $\text{last}(v) = \text{last}(\text{nf}_R(u))$. Thus, each non-geodesic u $4k$-fellow travels a geodesic word v with $\text{last}(v) = \text{last}(\text{nf}_R(u))$, so each rule (u, v) in L'' has $\text{last}(v) = \text{last}(\text{nf}_R(u))$. Thus, L'' is a length non-increasing end-normal synchronously regular CP-RS, completing the proof of part (a) of the theorem. Now, L'' satisfies the hypotheses of Theorem

2.3.2, so there is a geodesically autostackable structure for G with generating set A. $\square$

When working with the finite state automata constructed through Theorem 2.3.1, we notice that there is some potential room for improvement when working with the rewriting systems from pairs (G, A) with FFTP. In particular, rather than using the pumping number for an FSA accepting L (or L''), we can use four times the fellow traveler constant. This can decrease the size of the automata created in Theorem 2.3.1. There are pairs (G, A) for which Theorem 2.3.1 is useful which do not have FFTP, so we opted for a proof covering a wider class of groups rather than the more efficient construction in Section 2.3.

2.5 Disproving the converse of part (a) of the Main Theorem

A natural question, given the first part of Theorem 2.4.2, is whether having a length non-increasing regular CP-RS for a pair (G, A) implies that (G, A) has FFTP. In this section, we answer the question in the negative, using the following example.

Example 2.5.1. *The group $G = \mathbb{Z}^2 \rtimes \mathbb{Z}_2 = \langle a, b, t \mid [a, b] = 1, t^2 = 1, tat = b\rangle$ has a length non-increasing regular CP-RS with generating set $A = \{a, t\}$, but the pair (G, A) does not have FFTP.*

Proof. Consider $G = \mathbb{Z}^2 \rtimes \mathbb{Z}_2$ and $A = \{a, t\}$. Elder uses this example and proves that the pair (G, A) does not have FFTP in [13]. Consider the shortlex normal form set $\mathcal{N}$ with $a < a^{-1} < t < t^{-1}$ as our ordering on $A^{\pm 1}$. An illustration of part of the Cayley graph, with normal forms indicated, is given as Figure 2.4. We see that $\mathcal{N}$ consists of $\{a^i t a^j t \mid i, j \in \mathbb{Z}, j \neq 0\}$ together with all prefixes of words in this language. Notably, $\mathcal{N}$ is regular. Further, whenever $u, v \in \mathcal{N}$ with $d(u, v) = 1$, we have that

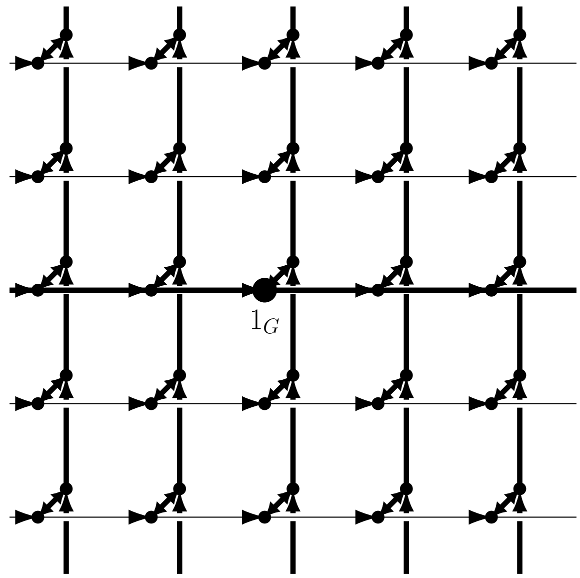

Figure 2.4: A portion of the Cayley graph $\Gamma_{G,A}$, with spanning tree of normal forms drawn in bold. All horizontal and vertical arrows are labeled by a and point right or up, while all diagonal arrows are labeled by t and are bidirectional.

u and v 4-fellow travel. The only non-trivial case to check for 4-fellow traveling is when $u = a^i ta^j t$ and $v = a^{i+1}ta^j t$, with $v =_G ua$. In this case, if i is non-negative, u and v follow a common path of length i, then have a word difference of ta for a single step, then have a word difference of $tata$ for j letters, and finally have a word difference of a at the final pair of vertices. If i is negative, we have a similar scenario. All other options for u and v have either u as a prefix of v or v as a prefix of u, so u and v 1-fellow travel. Thus, we have a regular language of shortlex normal forms, with any pair of normal forms that differ by a single edge 4-fellow traveling, hence a shortlex automatic structure [14, Theorem 2.3.5]. Every shortlex automatic structure is a length non-increasing regular CP-RS (this follows from the proof of [29, Lemma 5.1]), so this example has a length non-increasing regular CP-RS but not FFTP. The set of rules R for this CP-RS is given below, for sake of completeness:

$$
\begin{aligned}
a^i a^{-\mathrm{sgn}(i)} &\to a^{i-\mathrm{sgn}(i)} && \text{for all } i \in \mathbb{Z} \setminus \{0\} \\
a^i t a^j a^{-\mathrm{sgn}(j)} &\to a^i t a^{j-\mathrm{sgn}(j)} && \text{for all } i \in \mathbb{Z} \text{ and } j \in \mathbb{Z} \setminus \{0\} \\
a^i t t &\to a^i && \text{for all } i \in \mathbb{Z} \\
a^i t a^j t^2 &\to a^i t a^j && \text{for all } i \in \mathbb{Z} \text{ and } j \in \mathbb{Z} \setminus \{0\} \\
a^i t a^j t a^\epsilon &\to a^{i+\epsilon} t a^j t && \text{for all } i \in \mathbb{Z},\ j \in \mathbb{Z} \setminus \{0\}, \text{ and } \epsilon \in \{1, -1\}
\end{aligned}
$$

It is worth noting that this CP-RS is end-normal , hence G is geodesically autostackable with the given generating set. □

Chapter 3

A Completion Procedure for Autostackable Structures

3.1 Introduction

Autostackable structures are a generalization of both automatic structures and finite convergent rewriting systems. Both of these special cases have well-known procedures to search for a structure. In the case of finite convergent rewriting systems (FCRSs), the Knuth-Bendix completion procedure can, given a finite presentation $G = \langle A|R\rangle$ and a well-founded ordering $<$ on A^* which is compatible with concatenation, search for a finite set of rules and determine whether such a set is an FCRS. In the case of automatic groups, a procedure detailed by Epstein, Holt, and Rees in [15] can search for a word acceptor and multiplier automata and decide whether a set of such automata is an automatic structure.

The Knuth-Bendix completion procedure has been extended by several authors. For example, Needham [23] extended the Knuth-Bendix completion procedure to work with infinite convergent rewriting systems, while Andrianarivelo and Réty [1] extended the Knuth-Bendix completion procedure to prefix-preserving prefix-constrained term rewriting systems.

This chapter does much the same for bounded convergent prefix-rewriting systems in groups and monoids. We have created a completion procedure, which we call the prefix-Knuth-Bendix procedure, to search for bounded prefix-rewriting systems and determine whether such a rewriting system is convergent, given a presentation $M = Mon\langle A|R\rangle$ and an ordering $<$ on A^* satisfying certain properties. This new procedure has been implemented in Python, and details of this implementation are provided in [10].

In contrast to Needham's work, our procedure creates bounded rewriting systems. Needham's procedure involves rules such as $dacd(ba)^n c \to adacd(ba)^n$, which rewrite arbitrarily long subwords, while our procedure only creates prefix-rewriting systems which rewrite subwords of bounded length. Needham's procedure also requires orderings which are compatible with concatenation on both sides (as is the case with standard Knuth-Bendix), while our procedure allows a larger class of orderings because we require only compatibility with concatenation on the right. In contrast to Andrianarivelo and Réty's work, our procedure does not require prefix-preserving rules; that is, when interpreted in terms of string rewriting rather than the more general term rewriting, Andrianarivelo and Réty's work requires that for any pair of prefix-constrained rules (notation defined in Section 3.3) $L_1 : l_1 \to r_1$ and $L_2 : l_2 \to r_2$, if there exists $w = pl_1s$ such that $p \in L_1$, $s \in A^*$, and $w \in L_2$, then pr_1s must also be in L_2. Our work avoids this limitation.

In addition to defining the prefix-Knuth-Bendix procedure, we define a class of orderings (k-bounded regular orderings) and show that when one of these orderings is used in the prefix-Knuth-Bendix procedure, each step of the procedure can be calculated, and any resulting rewriting system is regular. In the case of groups, these are autostackable structures.

This chapter is organized as follows: In Section 3.2, we provide notation, definitions, and theorems which are used in the remainder of the chapter. In Section 3.3, we describe the completion procedure that we have developed for prefix-rewriting systems and prove that it is sound and that under certain conditions it gives a bounded convergent prefix-rewriting system. In Section 3.4, we define k-bounded regular orderings, which allow us to effectively compute each step of our completion procedure as long as we only need to compare words which differ on a suffix of length at most k, and we show that when the prefix-Knuth-Bendix procedure halts with an output while using one of these orderings with no step having required comparison of words which differ on a suffix of length greater than k, then the result is a bounded regular convergent prefix-rewriting system.

3.2 Background

In this section, we discuss definitions and theorems relevant to this chapter, along with a description of the standard Knuth-Bendix completion procedure. Some of the definitions from this section were discussed in less detail in Section 1.1, but it is valuable to have a more detailed discussion here.

Let $\to$ be a binary relation on A^*, i.e. a collection of rules of the form $l \to r$ for some $l, r \in A^*$. Let $\to^*$ be the transitive and reflexive closure of $\to$ and let $\leftrightarrow^*$ be the transitive, reflexive, and symmetric closure of $\to$. A relation $\to$ is confluent if whenever $u \leftrightarrow^* v$, there exists some $w \in A^*$ such that $u \to^* w$ and $v \to^* w$; is locally confluent if whenever $x \to u$ and $x \to v$, there exists some $w \in A^*$ such that $u \to^* w$ and $v \to^* w$; and is terminating if there is no infinite sequence of words $u_1 \to u_2 \to u_3 \to \cdots$. If $\to$ is both confluent and terminating, then $\to$ is *convergent*.

Lemma 3.2.1. *[25] A terminating relation is confluent if and only if it is locally confluent.*

We will deal primarily with binary relations in the form of prefix-rewriting systems. A *prefix-rewriting system* is a relation $\to$ on A^* such that whenever $l \to r$, we have

$ls \to rs$ for all $s \in A^*$. For the context of the Knuth-Bendix completion procedure, we will also deal with *rewriting systems*; these are relations $\to$ on A^* such that whenever $l \to r$, we have $pls \to prs$ for all $p, s \in A^*$. A *finite convergent rewriting system* is a set $\mathcal{R}$ of finitely many rules $l \to r$ such that the relation

$$\{u \to_{\mathcal{R}} v \mid u = pls \text{ and } v = prs \text{ for some } p, s \in A^* \text{ and } l \to r \text{ in } \mathcal{R}\}$$

is convergent. Similarly, a *convergent prefix-rewriting system* is a set $\mathcal{R}$ of rules $l \to r$ such that the relation

$$\{u \to_{\mathcal{R}} v \mid u = ls \text{ and } v = rs \text{ for some } s \in A^* \text{ and } l \to r \text{ in } \mathcal{R}\}$$

is convergent.

A *completion procedure* is a procedure which takes as input a set $\mathcal{E}$ of equations over A^* and a set $\mathcal{R}$ of rules over A^*, and repeatedly modifies $\mathcal{E}$ and $\mathcal{R}$ through *inference rules*. A single inference rule modifying the pair $(\mathcal{E}, \mathcal{R})$ to the pair $(\mathcal{E}', \mathcal{R}')$ is denoted as $(\mathcal{E}, \mathcal{R}) \vdash (\mathcal{E}', \mathcal{R}')$ or

$$\frac{(\mathcal{E}, \mathcal{R})}{(\mathcal{E}', \mathcal{R}')}.$$

A completion procedure is *sound* if $u =_{\mathcal{E} \cup \mathcal{R}} v$ if and only if $u =_{\mathcal{E}' \cup \mathcal{R}'} v$ whenever $(\mathcal{E}, \mathcal{R}) \vdash (\mathcal{E}', \mathcal{R}')$.

A *critical pair* is a pair of rewriting rules R_1 and R_2 in $\mathcal{R}$ whose left-hand sides have non-empty overlap, which results in a word $x \in A^*$ with rewritings $x \to_{\{R_1\}} u$ and $x \to_{\{R_2\}} v$ such that u and v do not have an immediate common descendant under $\{R_1, R_2\}$. This has the potential to cause the (prefix-) rewriting system $\mathcal{R}$ to not be locally confluent.

A strict partial order $<$ is *compatible with concatenation on the right* (*on the left*) if whenever $u < v$ and $w \in A^*$, we have $uw < vw$ (respectively $wu < wv$).

The Knuth-Bendix completion procedure [22] can be used to search for finite convergent rewriting systems for groups and monoids; given a monoid presentation $M = Mon\langle A \mid R \rangle$ and a well-founded strict partial ordering $<$ on A^* which is compatible with concatenation on both the left and the right (i.e. $a < b$ implies $pas < pbs$ for all $p, s \in A^*$), it uses a collection of inference rules to create rewriting systems for M, and terminates if one of these rewriting systems is convergent. These inference rules are Orient (which replaces an equation $u = v$ by a rule $u \to v$ or $v \to u$, using $<$ to determine which direction the rule goes), Deduce (which adds an equation from a critical pair), Simplify (which rewrites either side of an equation or the right side of a rule), and Delete (which deletes an equation of the form $u = u$). The Knuth-Bendix completion procedure uses two types of critical pairs: *external critical pairs*, which are pairs of rules $l_1 \to r_1$ and $l_2 \to r_2$ with $l_1 = uv$, $l_2 = vw$, and $v \neq \lambda$; and *internal critical pairs*, which are pairs of rules $l_1 \to r_1$ and $l_2 \to r_2$ with $l_1 = ul_2v$ and $l_2 \neq \lambda$. Our prefix-Knuth-Bendix procedure (detailed in Section 3.3) builds on the Knuth-Bendix completion procedure, though we are able to avoid the need for $<$ to be compatible with concatenation on the left, allowing us to use a wider class of orderings.

3.3 The Prefix-Knuth-Bendix Completion Procedure

In this section, we introduce a procedure to build bounded regular prefix-rewriting systems, which we call the prefix-Knuth-Bendix completion procedure, and prove that this procedure is valid and results in a convergent prefix-rewriting system if it halts with an output.

Throughout the prefix-Knuth-Bendix completion procedure, we will require information about prefixes. In order to keep track of this prefix information, we introduce prefix-constrained equations and rules:

Definition 3.3.1. *Let A be a finite alphabet. A* prefix-constrained equation *is a triple $P : u = v$ for some $P \subseteq A^*$ and $u, v \in A^*$, and a* prefix-constrained rule *is a triple $P : u \rightarrow v$ for some $P \subseteq A^*$ and $u, v \in A^*$. Given a collection $\mathcal{S}$ of prefix-constrained equations and prefix-constrained rules, we say that $x \leftrightarrow_{\mathcal{S}} y$ if $x = pus$ and $y = pvs$ for some p, u, v with $p \in P$ for some language P such that $P : u = v$, $P : v = u$, $P : u \rightarrow v$, or $P : v \rightarrow u$ is an element of $\mathcal{S}$. We use $=_{\mathcal{S}}$ to denote the transitive closure of $\leftrightarrow_{\mathcal{S}}$.*

We note that equality relative to a set of prefix-constrained equations and rules $\mathcal{S}$ is not generally the same as equality in the monoid

$$M = Mon\langle A \mid \{u = v \mid P : u = v \text{ or } P : u \rightarrow v \text{ is in } \mathcal{S} \text{ for some } P \subseteq A^*\}\rangle.$$

If each prefix language is A^* then the two equalities $=_{\mathcal{S}}$ and $=_M$ coincide.

Recall that $D = \{(a, a) \mid a \in A\}$ is the diagonal of A, π_i is the projection function given by $\pi_i(a_1, \ldots, a_n) = a_i$ when $a_i \in A_i$ and $\pi_i(a_1, \ldots, a_n) = \lambda$ when $a_i = \$_i$, and the quotient L/S is the set $\{w \in A^* \mid \text{ there exists } s \in S \text{ such that } ws \in L\}$. We next define three types of critical pairs:

Definition 3.3.2. *Let $P_1, P_2 \subseteq A^*$ and $l_1, l_2, r_1, r_2 \in A^*$. Consider the pair of rules $P_1 : l_1 \rightarrow r_1$ and $P_2 : l_2 \rightarrow r_2$.*

Suppose that l_2 is a non-empty subword of l_1, i.e. $l_1 = ul_2v$ for some $u, v \in A^*$. Then we have an interior critical pair *between our rules, with resulting prefix-constrained equation $P_1 \cap (P_2/\{u\}) : r_1 = ur_2v$.*

Suppose that some non-empty proper suffix of l_1 *is a proper prefix of* l_2*, i.e. there exist non-empty words* $u, v, w \in A^*$ *with* $v \neq \lambda$*,* $l_1 = uv$ *and* $l_2 = vw$*. Then we have an* exterior critical pair *between our rules, with resulting prefix-constrained equation* $P_1 \cap (P_2/\{u\}) : r_1 w = u r_2$*.*

Suppose that there exists some word in P_1 *which can be rewritten using* $P_2 : l_2 \to r_2$ *to a word in* $A^* \setminus P_1$*, i.e. the set* $(((P_2 \times P_2) \cap D^*) \cdot (l_2, r_2) \cdot D^*) \cap (P_1 \times (A^* \setminus P_1))$ *is non-empty. Then we have a* prefix critical pair *between our rules, with resulting prefix-constrained equation*

$$\pi_2\left((((P_2 \times P_2) \cap D^*) \cdot (l_2, r_2) \cdot D^*) \cap (P_1 \times (A^* \setminus P_1))\right) : l_1 = r_1$$

We note that interior and exterior critical pairs correspond to the critical pairs used in the standard Knuth-Bendix completion procedure. Prefix critical pairs are important for prefix-Knuth-Bendix because we use prefix-constrained rules.

Lemma 3.3.3. *Given a critical pair between rules* $R_1 = P_1 : l_1 \to r_1$ *and* $R_2 = P_2 : l_2 \to r_2$*, the set of equalities implied by the prefix-constrained equation resulting from this critical pair is also implied by the pair of rules* R_1 *and* R_2*.*

Proof. We consider the three types of critical pairs in turn:

Interior critical pairs: Consider an interior critical pair with $l_1 =_{A^*} u l_2 v$. Suppose $x =_{A^*} p r_1 s$ and $y =_{A^*} p u r_2 v s$ for some $p \in P_1 \cap (P_2/\{u\})$ and $s \in A^*$. Let $z =_{A^*} p u l_2 v s =_{A^*} p l_1 s$. Then $p \in P_1$, so $z =_{\{R_1\}} p r_1 s =_{A^*} x$. Also, $pu \in P_2$, so $z =_{\{R_2\}} p u r_2 v s =_{A^*} y$. Thus, $x =_{\{R_1, R_2\}} y$, as desired.

Exterior critical pairs: Consider an exterior critical pair with $l_1 =_{A^*} uv$ and $l_2 =_{A^*} vw$ for some non-empty word v. Suppose $x =_{A^*} p r_1 w s$ and $y =_{A^*} p u r_2 s$ for some $p \in P_1 \cap (P_2/\{u\})$ and $s \in A^*$. Let $z =_{A^*} p u v w s =_{A^*} p l_1 w s =_{A^*} p u l_2 s$. Then $p \in P_1$, so $z =_{\{R_1\}} pr_1ws =_{A^*} x$, and $pu \in P_2$, so $z =_{\{R_2\}} pur_2s =_{A^*} y$. Thus, $x =_{\{R_1,R_2\}} y$, as desired.

Prefix critical pairs: Consider a prefix critical pair. Let $x =_{A^*} pl_1s$ and $y =_{A^*} pr_1s$ for some $p \in \pi_2\left((((P_2 \times P_2) \cap D^*)(l_2, r_2)D^*) \cap (P_1 \times (A^* \setminus P_1))\right)$, and let $p_1 \in A^*$ such that $(p_1, p) \in (((P_2 \times P_2) \cap D^*)(l_2, r_2)D^*) \cap (P_1 \times (A^* \setminus P_1))$. Let $z =_{A^*} p_1s$. Then $z =_{A^*} p_2l_2p_3l_1s$ for some $p_2 \in P_2$ and $p_3 \in A^*$ such that $p_2l_2p_3 \in P_1$ but $p_2r_2p_3 \notin P_1$. Then $z =_{\{R_2\}} p_2r_2p_3l_1s =_{A^*} x$ and $z =_{\{R_1\}} p_2l_2p_3r_1s =_{\{R_2\}} p_2r_2p_3r_1s =_{A^*} y$. Thus, $x =_{\{R_1,R_2\}} y$, as desired. □

Note that the prefix language associated with each equation resulting from a critical pair is exactly the set of prefixes on which that critical pair is relevant. For interior and exterior critical pairs, this language is the language of prefixes where both relevant rules can be applied. For prefix critical pairs, this language is all of the words w which are not in P_1, but which are the result of applying the rule $P_2 : l_2 \to r_2$ to a word in P_1. In general these languages may not be computable, but in the case that P_1 and P_2 are regular, closure properties of regular languages (see Subsection 1.2.1) show that all of these languages are themselves regular and can be computed.

Definition 3.3.4. *A* prefix-Knuth-Bendix completion procedure *is a procedure which starts with a finite set of equations $\mathcal{E}_0$ of the form $P : u = v$ for some $P \subseteq A^*$ and some $u, v \in A^*$, a finite set $\mathcal{R}_0$ of rules of the form $P : u \to v$ for some $P \subseteq A^*$ and some $u, v \in A^*$, and a well-founded strict partial ordering $<$ compatible with concatenation on the right, and constructs a derivation $(\mathcal{E}_0, \mathcal{R}_0) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash \cdots$, where $(\mathcal{E}_{i+1}, \mathcal{R}_{i+1})$ is obtained from $(\mathcal{E}_i, \mathcal{R}_i)$ by applying one of the following inference rules:*

Augmentation*:*

$$\frac{(\mathcal{E}, \mathcal{R})}{(\mathcal{E} \cup \{P : u = v\}, \mathcal{R})}$$

where $P : u = v$ is the result of an interior critical pair, exterior critical pair, or prefix critical pair between rules of $\mathcal{R}$.

Orientation*:*

$$\frac{(\mathcal{E} \cup \{P : u = v\}, \mathcal{R})}{(\mathcal{E} \cup \{P \setminus (L_{v<u} \cup L_{u<v}) : u = v\}, \mathcal{R} \cup \{P \cap L_{u<v} : v \to u, P \cap L_{v<u} : u \to v\}}$$

where $L_{u<v}$ (respectively, $L_{v<u}$) is a set of words w such that $wu < wv$ (respectively $wv < wu$).

Word simplification*:*

$$\frac{(\mathcal{E} \cup \{P : u = v\}, \mathcal{R})}{(\mathcal{E} \cup \{P \cap P' : u' = v, P \setminus P' : u = v\}, \mathcal{R})}$$

where $wu \to_{\mathcal{R}} wu'$ for all $w \in P'$.

Prefix simplification*:*

$$\frac{(\mathcal{E} \cup \{P : u = v\}, \mathcal{R})}{(\mathcal{E} \cup \{P_1 : u = v, P_2 : u = v\}, \mathcal{R})}$$

where P_3 is $(((P' \times P') \cap D^*) \cdot (l', r') \cdot D^*) \cap (P \times A^*)$ for some rule $P' : l' \to r'$ in $\mathcal{R}$, $P_1 = \pi_2(P_3)$, and P_2 is $P \setminus \pi_1(P_3)$. That is, P_1 is the set of words which are the result of applying the rule $P' : l' \to r'$ to a word in P, and P_2 is the set of words in P which cannot be rewritten using the rule $P' : l' \to r'$.

Boundary simplification*:*

$$\frac{(\mathcal{E} \cup \{P : u = v\}, \mathcal{R})}{(\mathcal{E} \cup \{P_1 : u' = v', P_2 : u = v\}, \mathcal{R})}$$

where $u' = r'z$, $v' = xv$, $P_1 = P' \cap (P/\{x\})$, and $P_2 = P \setminus P_1 \cdot \{x\}$ for some rule $P' : l' \to r'$ in $\mathcal{R}$ with $l' = xy$ and $u = yz$ for some $y \neq_{A^*} \lambda$. That is, u' is the result of applying the rule $l' \to r'$ to the word xu, P_1 is the set of all words $w \in P'$ such that

$wx \in P$, and P_2 is the set of all words $w \in P$ such that either w does not end with the suffix x or w is not $w'x$ for any $w' \in P'$.

Equal word deletion*:*

$$\frac{(\mathcal{E} \cup \{P : u = u\}, \mathcal{R})}{(\mathcal{E}, \mathcal{R})}.$$

Equation empty prefix deletion*:*

$$\frac{(\mathcal{E} \cup \{\emptyset : u = v\}, \mathcal{R})}{(\mathcal{E}, \mathcal{R})}.$$

Rule empty prefix deletion*:*

$$\frac{(\mathcal{E}, \mathcal{R} \cup \{\emptyset : u \to v\})}{(\mathcal{E}, \mathcal{R})}.$$

It is worth noting that the Orientation inference rule requires the computation of languages $L_{u<v}$ and $L_{v<u}$, along with their intersections with P and the language $P \setminus (L_{v<u} \cup L_{u<v})$. Likewise, the three simplification inference rules each require a computation which may not be possible in general, but is possible if P and P' are regular. Detection of the empty set may not be possible in general, but is possible if all of the languages involved are regular and given as regular expressions or as the languages accepted by finite state automata. In Section 3.4 we will discuss a class of orderings (the k-bounded orderings) which, when used in a prefix-Knuth-Bendix completion procedure with no Orientation steps requiring comparison of words which differ on a suffix of length greater than k, ensure that each prefix language we generate is regular.

We now show that a prefix-Knuth-Bendix completion procedure is sound, and that under appropriate hypotheses it produces bounded convergent prefix-rewriting systems.

Theorem 3.3.5. *If* $(\mathcal{E}, \mathcal{R}) \vdash (\mathcal{E}', \mathcal{R}')$*, then the equivalence relations* $=_{\mathcal{E}\cup\mathcal{R}}$ *and* $=_{\mathcal{E}'\cup\mathcal{R}'}$ *are identical.*

Proof. Suppose that $(\mathcal{E}, \mathcal{R}) \vdash (\mathcal{E}', \mathcal{R}')$. We must show that if this inference removed a rule or equation, then all words which were equal under the removed equation remain equal under $(\mathcal{E}', \mathcal{R}')$, and that if this inference added a rule or equation, then all words which are equal under the new rule or equation were already equal under $(\mathcal{E}, \mathcal{R})$. We approach this one inference rule at a time.

Augmentation: This inference rule adds a single equation $P : u = v$ which is the result of a critical pair between rules of $\mathcal{R}$. Suppose that $w \in P$. Then by Lemma 3.3.3, we have $wu =_{\mathcal{R}} wv$.

Orientiation: This inference rule replaces an equation $P : u = v$ by a equation and two rules. Suppose that $w \in P$; then w is in one of $P \cap L_{v<u}$, $P \cap L_{u<v}$, or $P \setminus (L_{u<v} \cup L_{v<u})$, so $wu =_{\mathcal{E}'\cup\mathcal{R}'} wv$. Alternatively, if w is in one of $P \cap L_{v<u}$, $P \cap L_{u<v}$, or $P \setminus (L_{u<v} \cup L_{v<u})$, then $w \in P$, so $wu =_{\mathcal{E}} wv$.

Word simplification: This inference rule replaces an equation $P : u = v$ by two different equations.

Suppose $w \in P$, and consider the equation $wu =_{\mathcal{E}\cup\mathcal{R}} wv$. Then if $w \notin P'$, we have $wu =_{\mathcal{E}'} w_v$. Alternatively, if $w \in P'$, we have $wu =_{\mathcal{R}'} wu' =_{\mathcal{E}'} w_v$. In either case, $wu =_{\mathcal{E}'\cup\mathcal{R}'} wv$.

Suppose $w \in P \cap P'$, and consider the equation $wu' =_{\mathcal{E}'\cup\mathcal{R}'} wv$. We see that $wu' =_{\mathcal{R}} wu =_{\mathcal{E}} wv$, so $wu' =_{\mathcal{E}\cup\mathcal{R}} wv$.

Suppose $w \in P \setminus P'$ and consider the equation $wu =_{\mathcal{E}'\cup\mathcal{R}'} wv$. We see that $w \in P$, so $wu =_{\mathcal{E}\cup\mathcal{R}} wv$.

Prefix simplification: This inference rule replaces an equation $P : u = v$ by two equations.

Suppose $w \in P$, and consider the equation $wu =_{\mathcal{E}\cup\mathcal{R}} wv$. Then if $w = pl's$ for some $p \in P'$ and $s \in A^*$, we have $wu =_{\mathcal{R}'} pr'su =_{\mathcal{E}}' pr'sv =_{\mathcal{R}'} wv$. Alternatively, if $w \neq pl's$ for any $p \in P'$ and $s \in A^*$, then $w \notin \pi_1(P_3)$, so $w \in P_2$, giving $wu =_{\mathcal{E}'} wv$. In either case, $wu =_{\mathcal{E}'\cup\mathcal{R}'} wv$.

Suppose $w \in P_1$, and consider the equation $wu =_{\mathcal{E}'\cup\mathcal{R}'} wv$. Then there exists some w' such that $(w', w) \in P_3$; in particular, $w' = pl's$ and $w = pr's$ for some $p \in P'$ and $s \in A^*$. Thus, we have $wu =_{\mathcal{R}} w'u =_{\mathcal{E}} w'v =_{\mathcal{R}} wv$, so $wu =_{\mathcal{E}\cup\mathcal{R}} wv$.

Suppose $w \in P_2$, and consider the equation $wu =_{\mathcal{E}'\cup\mathcal{R}'} wv$. Then $w \in P$, so $wu =_{\mathcal{E}} wv$.

Boundary simplification: This inference rule replaces an equation $P : u = v$ by two equations.

Suppose $w \in P$, and consider the equation $wu =_{\mathcal{E}\cup\mathcal{R}} wv$. Then if $w = px$ for some $p \in P'$, we have $pxyz =_{\mathcal{R}'} pr'z =_{\mathcal{E}'} pxv =_{A^*} wv$. Alternatively, if w is not px for any $p \in P'$, we have $wu =_{\mathcal{E}'} wv$. In either case, $wu =_{\mathcal{E}'\cup\mathcal{R}'} wv$.

Suppose $w \in P_1$, and consider the equation $wu' = wv'$. Then $w \in P'$, so $wu' =_{A^*} wr'z =_{\mathcal{R}} wl'z =_{A^*} wxyz =_{A^*} wxv =_{A^*} wv'$, so $wu' =_{\mathcal{E}\cup\mathcal{R}} wv'$.

Suppose $w \in P_2$, and consider the equation $wu = wv$. Then $w \in P$, so $wu =_{\mathcal{E}} wv$.

Equal word deletion: This inference rule removes an equation $P : u = u$. We note that $wu =_{A^*} wu$, hence $wu =_{\mathcal{E}'\cup\mathcal{R}'} wu$.

Equation empty prefix deletion and Rule empty prefix deletion: These inference rules delete an equation or rule S of the form $\emptyset : u = v$ or $\emptyset : u \to v$. As the prefix language is empty, there are no words $x, y \in A^*$ such that $x \leftrightarrow_S y$, so removing S from $\mathcal{E} \cup \mathcal{R}$ does not affect the relation $\leftrightarrow_{\mathcal{E}\cup\mathcal{R}}$, hence does not affect the relation $=_{\mathcal{E}\cup\mathcal{R}}$. $\square$

Definition 3.3.6. *A derivation* $(\mathcal{E}_0, \mathcal{R}_0) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash \cdots \vdash (\mathcal{E}_n, \mathcal{R}_n)$ *is* fair *if:*

(a) $\mathcal{R}_0 = \emptyset$*;*

(b) For each equation $P : u = v$ *in* $\mathcal{E}_0$*, we have* $P = A^*$*;*

(c) $\mathcal{E}_n = \emptyset$*; and*

(d) $\mathcal{E}_0 \cup \cdots \cup \mathcal{E}_n$ *contains a complete set of equations resulting from critical pairs in* $\mathcal{R}_n$*.*

Note that, because we remove at most one equation from $\mathcal{E}$ with each inference rule, any fair derivation must have a finite initial set of equations $\mathcal{E}_0$. Additionally, because we add at most two rewriting rules to $\mathcal{R}$ with each inference rule, $\mathcal{R}_n$ must be finite.

Theorem 3.3.7. *Suppose that* $(\mathcal{E}_0, \mathcal{R}_0) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash \cdots \vdash (\mathcal{E}_n, \mathcal{R}_n)$ *is a fair derivation using the prefix-Knuth-Bendix completion procedure. Then* $\mathcal{R}_n$ *is a bounded convergent prefix-rewriting system for the monoid*
$M = Mon\langle A \mid \{l = r \mid A^* : l = r \text{ is in } \mathcal{E}_0\}\rangle$.

Proof. We first consider termination. Each rule $P : l \to r$ in $\mathcal{R}_n$ was added in an Orientation step. In this Orientation step, we required that P be a set of words w such that $wr < wl$; thus, we have $pl > pr$ for all $p \in P$. Because $<$ is compatible with concatenation on the right, we can extend this to $pls > prs$ for all $p \in P$ and $s \in A^*$. Since $<$ is well-founded, we see that $\to_{\mathcal{R}_n}$ is a terminating relation.

We next consider local confluence. Let $z \in A^*$, and suppose that $z \to_{\mathcal{R}} x$ and $z \to_{\mathcal{R}} y$. That is, there exist rules $P_1 : l_1 \to r_1$ and $P_2 : l_2 \to s_2$ and words $p_1 \in P_1$, $p_2 \in P_2$, $s_1 \in A^*$, and $s_2 \in A^*$ such that

$$z =_{A^*} p_1 l_1 s_1 \to_{\{P_1 : l_1 \to r_1\}} p_1 r_1 s_1 = x$$

and

$$z =_{A^*} p_2 l_2 s_2 \to_{\{P_2 : l_2 \to r_2\}} p_2 r_2 s_2 = y.$$

Without loss of generality, suppose that $|p_1| \leq |p_2|$, so that p_1 is a subword of p_2. Note that $|l_1| \neq 0$ and $|l_2| \neq 0$ because if $\lambda > w$ for some word w, then by compatibility with concatenation on the right we have $w > w^2 > w^3 > \cdots$, which would violate well-foundedness of $<$. We have several cases to consider:

Case 1: Suppose that $|p_1 l_1| \leq |p_2|$, so that $p_1 l_1$ is a subword of p_2. In this case, $p_2 = p_1 l_1 u$, so $y \to_{\mathcal{R}} p_1 r_1 u r_2 s_2$; additionally, $x = p_1 r_1 u l_2 s_2$. If $p_1 r_1 u \in P_2$, then $x \to_{\mathcal{R}} p_1 r_1 u r_2 s_2$, so x and y have a common descendant. Otherwise, we have a prefix cricital pair between our rules, and we have $p_1 l_1 u \in P_2$ while $p_1 r_1 u \notin P_2$, with $p_1 \in P_1$. This gives

$$(p_1 l_1 u, p_1 r_1 u) \in (((P_1 \times P_1) \cap D^*) \cdot (l_1, r_1) \cdot D^*) \cap (P_2 \times (A^* \setminus P_2)),$$

so $p_1 r_1 u l_2 s_2 = p_1 r_1 u r_2 s_2$ is contained in the associated prefix-constrained equation.

Case 2: Suppose that $|p_1| < |p_2| < |p_1 l_1| < |p_2 l_2|$. In this case, the subwords l_1 and l_2 overlap so that we have an external critical pair. In particular, $l_1 = uv$ and $l_2 = vw$ for some $u, v, w \in A^*$, and we have $z = p_1 uvw s_2$, $x = p_1 r_1 w s_2$, and $y = p_1 u r_2 s_2$. Since $p_1 \in P_1$ and $p_1 u = p_2 \in P_2$, we have $p_1 \in P_1 \cap (P_2/\{u\})$, so $x = y$ is contained in the prefix-constrained equation associated with this critical pair.

Case 3: Suppose that $|p_1| < |p_2|$ and $|p_2 l_2| \leq |p_1 l_1|$. In this case, l_2 is a subword of l_1, so we have an interior critical pair. In particular, $l_1 = u l_2 v$ for some $u, v \in A^*$, and we have $z = p_1 u l_2 v s_1$, $x = p_1 r_1 s_1$, and $y = p_1 u r_1 v s_1$. Since $p_1 \in P_1$ and $p_1 u = p_2 \in P_2$, we have $p_1 \in P_1 \cap (P_2/\{u\})$, so $x = y$ is contained in the prefix-constrained equation associated with this critical pair.

Case 4: Suppose that $|p_1| = |p_2|$ and $|p_1 l_1| < |p_2 l_2|$, so that $p_1 = p_2$ and l_1 is a prefix of l_2.. In this case, l_1 is a subword of l_2, so we have an interior critical pair, with $l_2 = u l_1 v$ for some $v \in A^*$ and $u =_{A^*} \lambda$. We have $z = p_2 l_1 v s_2$, $x = p_2 r_1 v s_2$, and $y = p_2 r_2 s_2$. Since $p_2 \in P_2$ and $p_2 = p_1 \in P_1$, we have $p_2 \in P_2 \cap (P_2/\{u\}$, so $x = y$ is contained in the prefix-constrained equation associated with this critical pair.

Case 5: Suppose that $|p_1| = |p_2|$ and $|p_2 l_2| < |p_1 l_1|$, so that $p_1 = p_2$ and l_2 is a prefix of l_1. This case is identical to case 4, with with x and y swapped.

Case 6: Suppose that $|p_1| = |p_2|$ and $|p_1 l_1| = |p_2 l_2|$, so that $p_1 = p_2$, $l_1 = l_2$, and $s_1 = s_2$. In this case, we have an interior critical pair with $u = v = \lambda$. Then $z = p_1 l_1 s_1$, $x = p_1 r_1 s_1$, and $y = p_1 r_2 s_1$. Since $p_1 \in P_1$ and $p_1 u = p_1 \in P_2$, we have $p_1 \in P_1 \cap (P_2/\{u\})$, so $x = y$ is contained in the prefix-constrained equation associated with this critical pair.

In each of these cases, x and y either have a common descendant from existing rules, or they are included in the prefix-constrained equation resulting from a critical pair in $\mathcal{R}_n$. In the latter case, they are included in a prefix-constrained equation in $\mathcal{E}_i$ for some i, and this equation was simplified to a certain point, rewriting x and y to some descendents x' and y', and either these common descendents were equal as words or they were oriented in an Orientation inference rule. Either way, x and y have a common descendent using the entirety of $\mathcal{R}_n$, so $\mathcal{R}_n$ is locally confluent. By Lemma 3.2.1, $\mathcal{R}_n$ is a confluent rewriting system.

Finally, we must show that $\mathcal{R}_n$ is a bounded confluent prefix-rewriting system for the monoid $M = Mon\langle A \mid \{l = r \mid A^* : l = r \text{ is in } \mathcal{E}_0\}\rangle$. Because our derivation is fair, we have that $P = A^*$ for each equation $P : u = v$ in $\mathcal{E}_0$, so equality in $\mathcal{E}_0$ is identical to equality in M. By Theorem 3.3.5, equality in $\mathcal{R}_n$ is identical to equality in $\mathcal{E}_0$, hence is identical to equality in M. Thus,

$$M \cong Mon\langle A \mid \{l = r \mid P : l \to r \text{ is in } \mathcal{R}_n \text{ for some } P \subseteq A^*\}\rangle.$$

This gives us that $\mathcal{R}_n$ is a confluent prefix-rewriting system for M. Additionally, the maximum length of l or r in any rule $P : l \to r$ in $\mathcal{R}_n$ is bounded by some finite number because we only have finitely many prefix-constrained rules in $\mathcal{R}_n$, so $\mathcal{R}_n$ is a bounded CP-RS. □

3.4 A new class of orderings for prefix-Knuth-Bendix

In this section, we define a class of strict partial orderings which, when used in the prefix-Knuth-Bendix completion procedure, allow effective computation of each inference rule, and give examples of orderings in this class. In order to collect related definitions together, we first define strict partial orderings with ties, which we will not use until later in the section:

Definition 3.4.1. *A* strict partial ordering with ties *is a pair of relations* $(<, \sim)$ *such that* $<$ *is a strict partial order, and* $\sim$ *is an equivalence relation satisfying all of the following:*

- *If* $x \sim y$*, then* x *and* y *are incomparable under* $<$*.*
- *Whenever* $x \sim y$ *and* $x < z$*, then* $y < z$*.*
- *Whenever* $x \sim y$ *and* $z < x$*, then* $z < y$*.*

If $u \sim v$, we will say that u and v are *tied under* $(<, \sim)$. We note that every strict partial ordering $<$ gives a strict partial ordering with ties $(<, =)$, and every strict partial ordering with ties gives a strict partial ordering by ignoring $\sim$. If $(<, \sim)$ is a strict partial ordering with ties over the set of words A^*, then we say that $(<, \sim)$ is *compatible with concatenation on the right* (respectively, *on the left*) if $<$ is compatible with concatenation on the right (respectively, left) and whenever $x \sim y$ and $z \in A^*$, we have $xz \sim yz$ (respectively, $zx \sim zy$).

We next introduce k-bounded regularity, which allows effective computation in prefix-Knuth-Bendix:

Definition 3.4.2. *We say that a strict partial ordering $<$ is* (synchronously) k-bounded regular *if the language*

$$L_{<,k} = \{(u,v) \mid u < v\} \cap \left(D^* \cdot \left(A^{\{0,1,\ldots,k\}} \times A^{\{0,1,\ldots,k\}}\right)\right)$$

is (synchronously) regular.

We say that a strict partial ordering with ties $(<, \sim)$ is (synchronously) k-bounded regular *if $<$ is (synchronously) k-bounded regular and the language*

$$L_{\sim,k} = \{(u,v) \mid u \sim v\} \cap \left(D^* \cdot \left(A^{\{0,1,\ldots,k\}} \times A^{\{0,1,\ldots,k\}}\right)\right)$$

is (synchronously) regular.

Moreover, if $<$ or $(<, \sim)$ is k-bounded regular for all $k \in \mathbb{N}$, *then we say that $<$ or $(<, \sim)$ is* bounded regular. *If the set $\{(u,v) \mid u < v\}$ ($\{(u,v) \mid u \sim v\}$, in the case of an equivalence relation $\sim$) is regular, then we say that $<$ (respectively $\sim$) is* regular. *A strict partial ordering with ties $(<, \sim)$ is regular if both $<$ and $\sim$ are regular.*

The language $L_{<,k}$ (respectively, $L_{\sim,k}$) is the set of all pairs of words (u, v) with $u < v$ (respectively $u \sim v$) such that u and v differ only on a suffix of length at most k. Note that bounded regularity is a stronger condition than k-bounded regularity for any specific value of k. It is also worth noting that $(k + 1)$-bounded regularity implies k-bounded regularity (since $L_{<,k} = L_{<,k+1} \cap \left(D^* \cdot \left(A^{\{0,1,\ldots,k\}} \times A^{\{0,1,\ldots,k\}}\right)\right)$, and $L_{\sim,k}$ can similarly be constructed from $L_{\sim,k+1}$), but k-bounded regularity does not imply $(k+1)$-bounded regularity. Similarly, regularity implies bounded regularity, but bounded regularity does not imply regularity (see Example 3.4.6 below).

We are interested in k-bounded regular orderings because they ensure regularity (and thus computability) of the various prefix languages involved in the prefix-Knuth-Bendix completion procedure, provided our derivation begins with regular prefix constraints for each rule and equation and does not require the comparison of words which differ on a suffix of length more than k.

Theorem 3.4.3. *Let $<$ be a k-bounded regular strict partial ordering. Then, given a finite set $\mathcal{E}$ of prefix-constrained equations and a finite set $\mathcal{R}$ of prefix-constrained rules such that the prefix language for each equation in $\mathcal{E}$ and for each rule in $\mathcal{R}$ is regular, and for each equation $P : u = v$ in $\mathcal{E}$ we have both $|u| \leq k$ and $|v| \leq k$, any inference rule $(\mathcal{E}, \mathcal{R}) \vdash (\mathcal{E}', \mathcal{R}')$ in the prefix-Knuth-Bendix completion procedure is computable and results in each prefix language from $(\mathcal{E}', \mathcal{R}')$ being regular.*

Proof. Suppose that $<$ is k-bounded regular and the prefix languages for all equations in $\mathcal{E}$ and all rules in $\mathcal{R}$ are regular and given as regular expressions or finite state automata. We approach this one inference rule at a time.

For Augmentation, we need to compute the prefix-constrained equation associated with a critical pair between $P_1 : l_1 \to r_1$ and $P_2 : l_2 \to r_2$. The words in these prefix-constrained equations are each concatenations of subwords of l_1, l_2, r_1, and r_2, which are computable. The prefix language is a regular language constructed from P_1, P_2, and appropriate subwords of l_1, l_2, r_1, and r_2 (using constructions under which regular languages are closed) in all three cases, so P is both computable and regular.

For Orientation, suppose that $P : u = v$ is an equation in $\mathcal{E}$, and that u and v differ only on a suffix of length at most k. Let $L_{v<u} = \pi_1(L_{<,k}/\{(v,u)^p\})$ and let $L_{u<v} = \pi_1(L_{<,k}/\{(u,v)^p\})$. Then $L_{v<u}$ is the set of all words $w \in A^*$ such that $wv < wu$, and similarly $L_{u<v}$ is the set of all words $w \in A^*$ such that $wu < wv$. We note that $L_{v<u}$ and $L_{u<v}$ are regular (because regular languages are closed under quotients and homomorphic images — see Subsection 1.2.1), so the languages $P \cap L_{v<u}$, $P \cap L_{u<v}$ and $P \setminus (L_{v<u} \cup L_{u<v})$ are aslo regular. Thus, the three languages that we add as prefix languages in this Orientation step are regular, and can be computed.

We note that if we have any equation $P : u = v$ such that u and v differ on a suffix of length more than k, then the prefix languages required in Orientation may not be regular. In our Python implementation of the prefix-Knuth-Bendix procedure, such equations are skipped in hopes of simplifying the equation using one of the three simplification inference rules.

For Word simplification, if $wu \to_{\mathcal{R}} wu'$ as a result of a rule $P_1 : l_1 \to r_1$ in $\mathcal{R}$, then we must have $u = pl_1s$ with $wp \in P_1$. We note that $P' = P_1/\{p\}$ is the set of all w such that $wu \to wu'$ via $P_1 : l_1 \to r_1$, and is a regular language. This also gives regularity for $P \cap P'$ and $P \setminus P'$, so the two prefix-constrained equations added with this inference rule are computable and have regular prefix languages.

For Prefix simplification and Boundary simplification, we construct P_1 and P_2 with regular languages (using only constructions under which regular languages are closed), so these are also computable and regular.

For Equal word deletion, we delete an entire rule, with no limitation on the prefix language P.

For Equation empty prefix deletion and Rule empty prefix deletion, we note that the empty language detection problem is solvable for regular languages by finding the breadth-first search normal form of the associated finite state automaton. Thus, we can detect when P is empty for any equation $P : u = v$ or rule $P : u \to v$.

Thus, each of our inference rules can be computed, and result in regular prefix languages for all equations in $\mathcal{E}'$ and rules in $\mathcal{R}'$ for which we can construct the appropriate regular expressions or finite state automata. □

Note that we do not bound the length of words in prefix-constrained equations or rules in $(\mathcal{E}', \mathcal{R}')$. This means that while a k-bounded regular ordering can be sufficient so long as our words do not grow too long, it can be helpful to have a bounded regular ordering to avoid the problem of words growing too long.

Combining Theorem 3.4.3 with Theorem 3.3.7 gives the following useful corollary:

Corollary 3.4.4. *Suppose that $\mathcal{R}_n$ is the result of using the prefix-Knuth-Bendix procedure with inputs of a monoid $M = Mon\langle A \mid R\rangle$ and a well-founded k-bounded regular strict partial ordering $<$ which is compatible with concatenation on the right, and that no step of the prefix-Knuth-Bendix procedure required the comparison of two words u and v which differ on a suffix of length greater than k. Then $\mathcal{R}_n$ is a bounded regular convergent prefix-rewriting system for M, and is an autostackable structure if M is a group.*

We proceed next to examples of k-bounded regular orderings. We first show that all strict partial orderings which are compatible with concatenation on both sides are bounded regular:

Lemma 3.4.5. *Let $<$ be a strict partial order which is compatible with concatenation on both the left and the right. Then $<$ is k-bounded regular for all k.*

Proof. We note that the set $S_k = \{(u, v) \mid u < v \text{ and } |u| \leq k \text{ and } |v| \leq k\}$ is finite, since S_k is a subset of the set of pairs of words of length at most k, which is itself finite. Then the set $L_{<,k}$ is $D^* \cdot S_k$. This is a concatenation of two regular languages, hence is regular, so $<$ is k-bounded regular. □

(It is perhaps worth noting that the finite state automata for $L_{<,k}$ can only be constructed if $<$ is recursive.) This shows that any ordering which can be used for standard Knuth-Bendix can also be used for prefix-Knuth-Bendix (though the resulting rules will necessarily have A^* as their prefix languages, so prefix-Knuth-Bendix gives no rewriting systems beyond those given by standard Knuth-Bendix if we restrict ourselves to these orderings). We can use this to find examples of bounded regular orderings which are not regular.

Example 3.4.6. *Consider the recursive path ordering (see, for example, [12] for a definition) $<_{rpo}$ on $\{a, b\}^*$ given by $a < b$. Then $<_{rpo}$ is bounded regular but not regular.*

Proof. By Lemma 3.4.5, $<_{rpo}$ is bounded regular. Consider the pair of words $a^n b^n$ and $b^n a^n$. We note that $a^n b^n <_{rpo} b^n a^n$, so $(a^n b^n, b^n a^n) \in L_{<_{rpo}}$, but for any decomposition $uvw = (a^n b^n, b^n a^n)$ with $|uv| \leq n$ and $|v| \geq 1$, we have $v = (a^i, b^i)$. Then $uv^0 w = (a^{n-i} b^n, b^{n-i} a^n)$, and $a^{n-i} b^n >_{rpo} b^{n-i} a^n$, so $uv^0 w \notin L_{<_{rpo}}$. This is a violation of the pumping lemma, so $L_{<_{rpo}}$ cannot be regular. Thus, $<_{rpo}$ is bounded regular but not regular. □

We next introduce a modification of shortlex orderings which is compatible with concatenation on the right but not on the left.

Definition 3.4.7. *Let A be a finite alphabet, let M be a finite state automaton over A with start state 1_M, and for each state s of M, let $\prec_s$ be an ordering on A. We define the* regular-split shortlex ordering *$<_M$ on A^* as follows:*

- *If $|u| < |v|$, then $u <_M v$.*

- *If $|u| = |v|$, let p be the longest common prefix of u and v, and let u' and v' be suffixes of u and v such that $u = pu'$ and $v = pv'$. Let $s = \widehat{\delta}(1_M, p)$ (recall that $\widehat{\delta}(q, p)$ is the state reached by starting at state q and reading the word p). If first$(u') \prec_s$ first(v'), then $u' <_M v'$.*

These regular-split shortlex orderings have already proven useful; we will discuss a Coxeter group which has no finite convergent rewriting system with a shortlex ordering on the standard generating set, but which does have a bounded regular CP-RS with a regular-split shortlex ordering in Example 3.4.11.

Theorem 3.4.8. *Every regular-split shortlex ordering is a well-founded regular strict total ordering compatible with concatenation on the right.*

Proof. Let $<_M$ be a regular-split shortlex ordering on A^*. We must prove several things: that $<_M$ is a strict total ordering; that $<_M$ is well-founded; that $<_M$ is compatible with concatenation on the right; and that $<_M$ is regular.

$<_M$ is a strict total ordering: We begin by proving irreflexivity. When comparing a to itself, we have $|a| = |a|$. The longest common prefix of a and itself is a, and this gives $u' = v' = \lambda$. By definition, first$(\lambda) = \lambda$, and $\lambda \not\prec_s \lambda$ for all states s of M. Thus, $a \not<_M a$.

We next consider asymmetry. Suppose $a <_M b$. Then we have two cases:

Case 1: Suppose $|a| < |b|$. Then $|b| \not< |a|$ and $|b| \neq |a|$, so $b \not<_M a$.

Case 2: Suppose $|a| = |b|$. Then a and b have a longest common prefix p with $a = pa'$ and $b = pb'$ for some words a' and b'. Let s be the state in M reached by p. We see that first$(a') \prec_s$ first(b'), and thus first$(b') \not\prec_s$ first(a'). Thus, $b \not<_M a$. In both cases, $b \not<_M a$, so $<_M$ is asymmetric.

We next consider transitivity. Suppose $a <_M b$ and $b <_M c$. There are two cases to consider.

Case 1: Suppose $|a| < |b|$ or $|b| < |c|$. Then $|a| < |c|$, so $a <_M c$.

Case 2: Suppose $|a| = |b|$ and $|b| = |c|$. Then $|a| = |c|$. Let p_1 be the longest common subword of a and b, let p_2 be the longest common subword of b and c, and let p_3 be the longest common subword of a and c. We have several subcases:

Subcase i: Suppose that $|p_1| = |p_2| = |p_3|$; then $a = p_1a'$, $b = p_1b'$, and $c = p_1c'$ for some words a', b', c', and $p_1 = p_2 = p_3$. Let s be the state in M reached by p_1. Then $\text{first}(a') \prec_s \text{first}(b') \prec_s$ first(c'), so $\text{first}(a') \prec_s \text{first}(c')$, and hence $a <_M c$.

Subcase ii: Suppose that $|p_1| = |p_2| < |p_3|$. Then $p_1 = p_2$ is a prefix of p_3; in particular, $a = p_1a'$, $b = p_1b'$, and $c = p_1c'$ for some words a', b', and c'. But we see that $\text{first}(a') = \text{first}(c')$ because $|p_3| > |p_1|$, while $\text{first}(a') \prec_s \text{first}(b')$ and $\text{first}(b') \prec_s \text{first}(c')$, so that $\prec_s$ cannot be an ordering. Thus, this case is impossible.

Subcase iii: Suppose that $|p_1| = |p_2| > |p_3|$. Then $p_1 = p_2$ is a common prefix of a, b, and c, which is longer than p_3. This case is impossible.

Subcase iv: Suppose that $|p_1| < |p_2|$. Then p_1 is the longest common prefix of a and c. Let s be the state of M reached by p_1. We see that $a = p_1a'$, $b = p_1b'$, and $c = p_1c'$ for some a', b', c'; moreover, $\text{first}(a') \prec_s \text{first}(b') = \text{first}(c')$. Thus, $a <_M c$.

Subcase v: Suppose that $|p_1| > |p_2|$. Then p_2 is the longest common prefix of a and c. Let s be the state of M reached by p_2. We see that $a = p_2a'$, $b = p_2b'$, and $c = p_2c'$ for some a', b', c'; moreover, $\text{first}(a') = \text{first}(b') \prec_s \text{first}(c')$. Thus, $a <_M c$.

In all cases we have $a <_M c$, so $<_M$ is transitive.

Next, we show that $<_M$ is a strict total (rather than partial) ordering. Suppose that $a \neq b$. If $|a| < |b|$ then $a <_M b$, and if $|a| > |b|$ then $b <_M a$. If $|a| = |b|$, then the two words have some longest common prefix p, which reaches some state s in M. We have $a = pa'$ and $b = pb'$ for some words non-empty words a' and b'

in A^*, and $\text{first}(a') \neq \text{first}(b')$. Either $\text{first}(a') \prec_s \text{first}(b')$, in which case $a <_M b$, or $\text{first}(a') \succ_s \text{first}(b')$, in which case $b <_M a$. Thus, either $a <_M b$ or $b <_M a$, so $<_M$ is a strict total ordering.

$<_M$ *is well-founded:* Suppose, for sake of contradiction, that there exists an infinite sequence $a_1 >_M a_2 >_M a_3 >_M \cdots$. For each n, there are only finitely many words of length at most n, and by transitivity and asymmetry of $<_M$ we cannot repeat any word in this sequence, so this infinite sequence must contain some word a_i of length at least $|a_1| + 1$; but $a_i >_M a_1$, which is a contradiction. Thus, $<_M$ must be well-founded.

$<_M$ *is compatible with concatenation on the right:* Suppose $a <_M b$ and $u \in A^*$. We have two cases to consider:

Case 1: Suppose $|a| < |b|$. Then $|au| < |bu|$, so $au <_M bu$.

Case 2: Suppose $|a| = |b|$. Then let p be the longest common prefix of a and b, and let $s = \widehat{\delta}(1_M, p)$. Then $a = pa'$ and $b = pb'$ for some non-empty words a' and b' with $\text{first}(a') \prec_s \text{first}(b')$. We have $|au| = |bu|$, and the longest common prefix of au and bu is p. Moreover, $au = pa'u$ and $bu = pb'u$. We see that $\text{first}(a'u) = \text{first}(a') \prec_s \text{first}(b') = \text{first}(b'u)$, so $au <_M bu$.

In both cases, $au <_M bu$, so $<_M$ is compatible with concatenation on the right.

$<_M$ *is regular:* We construct a finite state automaton accepting the language $L_{<_M} = \{(u, v) \mid u <_M v\}$ as follows:

There are $|M| + 4$ states: one state corresponding to each state from M, along with states labeled "FAIL", "u SHORT", "u LEX", and "v LEX".

The initial state is the state corresponding to the initial state of M.

The accept states are the states labeled "u SHORT" and "u LEX".

We define the transition function δ as follows:

For each state s corresponding to a state in M and each letter $a \in A$, let $\delta(s, (a, a))$ be the state corresponding to $\delta_M(s, a)$.

For each state s except for the state labeled "FAIL" and each letter $a \in A$, let $\delta(s, (\$, a))$ be the state labeled "u SHORT".

For each state s corresponding to a state in M and each pair of letters $a, b \in A$ with $a \prec_s b$, let $\delta(s, (a, b))$ be the state labeled "u LEX", and let $\delta(s, (b, a))$ be the state labeled "v LEX".

For each pair of letters $a, b \in A$, let $\delta($"u SHORT"$, (a, b)) =$ "u FAIL", let $\delta($"u LEX"$, (a, b)) =$ "u LEX", and let $\delta($"v LEX"$, (a, b)) =$ "v LEX".

For each pair of letters $a, b \in A \cup \{\$\}$, let $\delta($"FAIL"$, (a, b)) =$ "FAIL".

For each state s and each letter $a \in A$, let $\delta(s, (a, \$)) =$ "FAIL".

For each state s, $\delta(s, (\$, \$)) =$ "FAIL".

This finite state automaton emulates M on the longest common prefix p of u and v, then shifts to checking shortlex with the ordering $\prec_s$ associated with the state s that is reached by starting at 1_M and reading p. The four added states are a fail state "FAIL", a state "u SHORT" which is reached whenever $|u| < |v|$, and states "u LEX" and "v LEX" which are reached whenever the appropriate word is lexicographically smaller under $\prec_s$. Thus, we have a finite state automaton accepting $L_{<_M}$, so $<_M$ is regular. □

We can extend the number of k-bounded regular strict partial orderings significantly with the following theorem:

Theorem 3.4.9. *Given two k-bounded regular well-founded strict partial orderings with ties $(<_1, \sim_1)$ and $(<_2, \sim_2)$ over A^* which are both compatible with concatenation on the right, the pair of relations $(<_{1,2}, \sim_{1,2})$ given by $a <_{1,2} b$ if $a <_1 b$, or $a \sim_1 b$*

and $a <_2 b$*; and* $a \sim_{1,2} b$ *if* $a \sim_1 b$ *and* $a \sim_2 b$*; is a* k*-bounded regular well-founded strict partial ordering with ties which is compatible with concatenation on the right.*

Proof. We must show that $(<_{1,2}, \sim_{1,2})$ satisfies several properties: that $<_{1,2}$ is a strict partial ordering; that $\sim_{1,2}$ is an equivalence relation; that if $x \sim_{1,2} y$ then x and y are incomparable under $<_{1,2}$; that if $x \sim_{1,2} y$ and $x <_{1,2} z$ then $y <_{1,2} z$; that if $x \sim_{1,2} y$ and $z <_{1,2} x$, then $z <_{1,2} y$; that $(<_{1,2})$ is well-founded; that both $<_{1,2}$ and $\sim_{1,2}$ are compatible with concatenation on the right; and that $(<_{1,2}, \sim_{1,2})$ is k-bounded regular.

$<_{1,2}$ *is a strict partial ordering:* We first show irreflexivity. We note that $a \not<_1 a$, $a \sim_1 a$, and $a \not<_2 a$, so $a \not<_{1,2} a$. Thus, $<_{1,2}$ is irreflexive.

We next show asymmetry. Suppose $a <_{1,2} b$. Then either $a <_1 b$, or $a \sim_1 b$ and $a <_2 b$. In the first case, we see that $b \not<_1 a$ and $b \not\sim_1 a$, so $b \not<_{1,2} a$. In the second case, we see that $b \not<_1 a$, $b \sim_1 a$, and $b \not<_2 a$, so $b \not<_{1,2} a$. Thus, $<_{1,2}$ is asymmetric.

Finally, we show transitivity. Suppose $a <_{1,2} b$ and $b <_{1,2} c$. We have four cases.

Case 1: Suppose that $a <_1 b$ and $b <_1 c$. By transitivity of $<_1$, we have that $a <_1 c$, and hence $a <_{1,2} c$.

Case 2: Suppose that $a <_1 b$ and $b \not<_1 c$. Then $b \sim_1 c$, so $a <_1 c$, and hence $a <_{1,2} c$.

Case 3: Suppose that $a \not<_1 b$ and $b <_1 c$. Then $a \sim_1 b$, so $a <_1 c$, and hence $a <_{1,2} c$.

Case 4: Suppose that $a \not<_1 b$ and $b \not<_1 c$. Then $a \sim_1 b$ and $b \sim_1 c$, so $a \sim_1 c$. Further, $a <_2 b$ and $b <_2 c$, so $a <_2 c$. Thus, $a <_{1,2} c$.

Thus, $<_{1,2}$ is transitive, and is a strict partial ordering.

$\sim_{1,2}$ *is an equivalence relation:* We see that for any $a \in A^*$, $a \sim_1 a$ and $a \sim_2 a$, so $a \sim_{1,2} a$, so $\sim_{1,2}$ is reflexive.

Now, suppose $a \sim_{1,2} b$. Then $a \sim_1 b$ and $a \sim_2 b$, so $b \sim_1 a$ and $b \sim_2 a$, so $b \sim_{1,2} a$, so $\sim_{1,2}$ is symmetric.

Finally, suppose that $a \sim_{1,2} b$ and $b \sim_{1,2} c$. Then $a \sim_1 b \sim_1 c$ and $a \sim_2 b \sim_2 c$, so $a \sim_{1,2} c$. Thus, $\sim_{1,2}$ is transitive, and is in fact an equivalence relation.

$a \sim_{1,2} b$ implies that a and b are incomparable under $<_{1,2}$: Suppose that $a \sim_{1,2} b$. Then $a \sim_1 b$ and $a \sim_2 b$, so $a \not<_1 b$ and $a \not<_2 b$, so $a \not<_{1,2} b$. Similarly, $b \not<_{1,2} a$.

$a \sim_{1,2} b$ and $a <_{1,2} c$ implies $b <_{1,2} c$: Suppose $a \sim_{1,2} b$ and $a <_{1,2} c$. We have two cases:

Case 1: Suppose that $a <_1 c$. Then $b <_1 c$ as well, so $b <_{1,2} c$.

Case 2: Suppose that $a \sim_1 c$ and $a <_2 c$. Then $b \sim_1 c$ and $b <_2 c$ as well, so $b <_{1,2} c$.

In either case, we have that $b <_{1,2} c$.

$a \sim_{1,2} b$ and $c <_{1,2} a$ implies $c <_{1,2} b$: This is analogous to the previous point, with direction of inequalities reversed.

$<_{1,2}$ is well-founded: Suppose, for sake of contradiction, that there exists an infinite sequence $a_1 >_{1,2} a_2 >_{1,2} a_3 >_{1,2} \cdots$. If we consider this sequence in terms of $(<_1, \sim_1)$, we see that we have $a_i >_1 a_{i+1}$ or $a_i \sim_1 a_{i+1}$ for all $i \in \mathbb{N}$. This gives two cases:

Case 1: Suppose that there are infinitely many indices i such that $a_i >_1 a_{i+1}$. By transitivity of $\sim_1$, this gives an infinite sequence $a_1 \sim_1 a_{i_1} >_1 a_{i_1+1} \sim_1 a_{i_2} >_1 \cdots$ for some indices $i_1, i_2, \ldots$. By compatibility of $\sim_1$ with $<_1$, this simplifies to an infinite sequence $a_{i_1} >_1 a_{i_2} >_1 \cdots$, contradicting well-foundedness of $>_1$.

Case 2: Suppose, on the contrary, that there are only finitely many indices i such that $a_i >_1 a_{i+1}$. Then there exists some $n \in \mathbb{N}$ such that for all $j \geq n$, we have $a_j \sim_1 a_{j+1}$ and $a_j >_2 a_{j+1}$. This gives an infinite sequence $a_n >_2 a_{n+1} >_2 a_{n+2} >_2 \cdots$, contradicting well-foundedness of $>_2$.

In either case, we reach a contradiction, so $>_{1,2}$ is well-founded.

$(<_{1,2}, \sim_{1,2})$ is compatible with concatenation on the right: Suppose that $a <_{1,2} b$ and $u \in A^*$. We have two cases:

Case 1: Suppose $a <_1 b$. Then by compatibility with concatenation on the right of $<_1$, we have $au <_1 bu$. Thus, $au <_{1,2} bu$.

Case 2: Suppose $a \sim_1 b$ and $a <_2 b$. Then by compatibility with concatenation on the right of $\sim_1$ and $<_2$, we have $au \sim_1 bu$ and $au <_2 bu$, so $au <_{1,2} bu$.

Thus, $<_{1,2}$ is compatible with concatenation on the right.

Now, suppose that $a \sim_{1,2} b$ and $u \in A^*$. Then $a \sim_1 b$ and $a \sim_2 b$, so by compatibility with concatenation on the right of both $\sim_1$ and $\sim_2$, we have $au \sim_1 bu$ and $au \sim_2 bu$, so $au \sim_{1,2} bu$. Thus, $\sim_{1,2}$ is also compatible with concatenation on the right.

The pair $(<_{1,2}, \sim_{1,2})$ *is* k*-bounded regular*: We construct the languages $L_{<_{1,2},k}$ and $L_{\sim_{1,2},k}$ as follows:

$$L_{<_{1,2},k} = L_{<_1,k} \cup (L_{\sim_1,k} \cap L_{<_2,k})$$

and

$$L_{\sim_{1,2},k} = L_{\sim_1} \cap L_{\sim_2,k}$$

We note that both of these are constructed by finitely many intersections and unions of regular langauges, hence are regular. Thus, $(<_{1,2}, \sim_{1,2})$ is a k-bounded regular strict partial ordering with ties. □

We say that an ordering constructed in this way is a *tiebreak ordering*, with $<_2$ *breaking ties from* $(<_1, \sim_1)$. We can use this to construct new orderings from familiar orderings. For example:

Definition 3.4.10. *Let A be a finite alphabet and let $wt : A \to \mathbb{R}_+$ be a system of positive weights on A, and for any word $u = a_1 a_2 \cdots a_n$, let $wt(u) = \sum_{i=1}^{n} w(a_i)$. Define the weight strict partial order with ties $(<_{wt}, \sim_{wt})$ by $u <_{wt} v$ whenever $wt(u) < wt(v)$, and $u \sim_{wt} v$ whenever $wt(u) = wt(v)$. Let $<_M$ be any regular-split shortlex* *ordering. Then the* regular-split weightlex ordering *given by wt and M is the tiebreak ordering with $<_M$ breaking ties from $<_{wt}$.*

Regular-split weightlex orderings are a natural analog of weightlex orderings.

We conclude with the following example of a bounded regular CP-RS which we discovered by using our implementation of prefix-Knuth-Bendix with a regular-split shortlex ordering.

Example 3.4.11. *Let M be the finite state automaton with four states labeled A, B, C, and D, alphabet $\{a, b, c, d\}$, start state A, and transition function $\delta(s, a) = A$, $\delta(s, b) = B$, $\delta(s, c) = C$, and $\delta(s, d) = D$ for all states s. Then the Coxeter group*

$$\begin{aligned} C = \langle a, b, c, d \,| a^2 = b^2 = c^2 = d^2 = 1, abab = baba, acac = caca, \\ ada = dad, bcb = cbc, bdbd = dbdb, cdcd = dcdc\rangle \end{aligned}$$

has a bounded regular CP-RS using the regular-split shortlex ordering given by the finite state automaton M with orderings $a \prec_A b \prec_A c \prec_A d$, $b \prec_B c \prec_B d \prec_B a$, $c \prec_C d \prec_C a \prec_C b$, and $d \prec_D a \prec_D b \prec_D c$.

Hermiller noted that this Coxeter group has no finite convergent rewriting system with the standard generating set [17]. Our procedure, as implemented in Python, finds an autostackable structure; we have provided the set of rules in Appendix A.

Appendix A

Bounded Regular Convergent Prefix-Rewriting System for C

The rules below form a bounded regular CP-RS for the Coxeter group given in Example 3.4.11. They have been grouped based on prefix languages for sake of readability.

P	Rules $u \to v$ such that $P : u \to v$
A^*	$aa \to \lambda, bb \to \lambda, cc \to \lambda, dd \to \lambda$
$A^*\{a\} \cup \{\lambda\}$	$dad \to ada, dacdcd \to adacdc, dabdbd \to adabdb,$ $dabdbcdcd \to adabdbcdc$
$A^*\{b, c, d\}$	$ada \to dad$
$A^*\{b\}$	$abab \to baba$
$A^*\{a, c, d\} \cup \{\lambda\}$	$baba \to aba, babcaca \to ababcac, babdad \to ababda,$ $babcacdad \to ababcacda, babdacdcd \to ababdacdc,$ $babdabdbd \to ababdabdb, babcacdacdcd \to ababcacdacdc,$ $babcacdabdbd \to ababcacdabdb, babdabdbcdcd \to ababdabdbcdc,$ $babcacdabdbcdcd \to ababcacdabdbcdc$

(Rules continue on next page.)

$A^*\{c\}$	$bcb \to cbc, bcdbdb \to cbcdbd, bcabab \to cbcaba,$ $bcdbdabab \to cbcdbdaba$
$A^*\{a,b,d\} \cup \{\lambda\}$	$cbc \to bcb$
$A^*\{d\}$	$cdcd \to dcdc$
$A^*\{a,b,c\} \cup \{\lambda\}$	$dcdc \to cdcd, dcdacac \to cdcdaca, dcdbcb \to cdcdbc,$ $dcdacabcb \to cdcdacabc, dcdbcdbdb \to cdcdbcdbd,$ $dcdbcabab \to cdcdbcaba, dcdacabcdbdc \to cdcdacabcdbd,$ $dcdacabcabab \to cdcdacabcaba, dcdbcdbdabab \to cdcdbcdbdaba,$ $dcdacabcdbdabab \to cdcdacabcdbdaba$
$A^*\{a,b\} \cup \{\lambda\}$	$dbdb \to bdbd, dbdabab \to bdbdaba$
$A^*\{c,d\}$	$bdbd \to dbdb, bdbcdcd \to dbdbcdc$
$A^*\{a,d\} \cup \{\lambda\}$	$caca \to acac, cacdad \to acacda, cacdacdcd \to acacdacdc,$ $cacdabdbd \to acacdabdb, cacdabdbcdcd \to acacdabdbcdc$
$A^*\{b,c\}$	$acac \to caca, acabcb \to cacabc, acabcdbdb \to cacabcdbd,$ $acabcabab \to cacabcaba, acabcdbdabab \to cacabcdbdaba$

Bibliography

[1] N. Andrianarivelo and P. Réty. Confluence of prefix-constrained rewrite systems. In *3rd International Conference on Formal Structures for Computation and Deduction*, volume 108 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 6, 15. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2018.

[2] W. W. Boone. The word problem. *Proc. Nat. Acad. Sci. U.S.A.*, 44:1061–1065, 1958.

[3] M. Brittenham and S. Hermiller. A uniform model for almost convexity and rewriting systems. *J. Group Theory*, 18(5):805–828, 2015.

[4] M. Brittenham, S. Hermiller, and D. Holt. Algorithms and topology of Cayley graphs for groups. *J. Algebra*, 415:112–136, 2014.

[5] M. Brittenham, S. Hermiller, and A. Johnson. Homology and closure properties of autostackable groups. *J. Algebra*, 452:596–617, 2016.

[6] M. Brittenham, S. Hermiller, and T. Susse. Geometry of the word problem for 3-manifold groups. *J. Algebra*, 499:111–150, 2018.

[7] J. W. Cannon. Almost convex groups. *Geom. Dedicata*, 22(2):197–210, 1987.

[8] N. Corwin, G. Golan, S. Hermiller, A. Johnson, and Z. Šunić. Autostackability of Thompson's group F. *J. Algebra*, 545:111–134, 2020.

[9] A. DeClerk. The falsification by fellow traveler property implies geodesic autostackability, 2022. arXiv: 2209.11912.

[10] A. DeClerk. A completion procedure for autostackable structures. In preparation.

[11] M. Dehn. Über unendliche diskontinuierliche Gruppen. *Math. Ann.*, 71(1):116–144, 1911.

[12] N. Dershowitz. Orderings for term-rewriting systems. *Theoret. Comput. Sci.*, 17(3):279–301, 1982.

[13] M. Elder. Regular geodesic languages and the falsification by fellow traveler property. *Algebr. Geom. Topol.*, 5:129–134, 2005.

[14] D. B. A. Epstein, J. W. Cannon, D. F. Holt, S. V. F. Levy, M. S. Paterson, and W. P. Thurston. *Word processing in groups.* Jones and Bartlett Publishers, Boston, MA, 1992.

[15] D. B. A. Epstein, D. F. Holt, and S. E. Rees. The use of Knuth-Bendix methods to solve the word problem in automatic groups. volume 12, pages 397–414. 1991. Computational group theory, Part 2.

[16] S. Hermiller and C. Martínez-Pérez. HNN extensions and stackable groups. *Groups Geom. Dyn.*, 12(3):1123–1158, 2018.

[17] S. M. Hermiller. Rewriting systems for Coxeter groups. *J. Pure Appl. Algebra*, 92(2):137–148, 1994.

[18] S. M. Hermiller and J. Meier. Tame combings, almost convexity and rewriting systems for groups. *Math. Z.*, 225(2):263–276, 1997.

[19] D. F. Holt. Garside groups have the falsification by fellow-traveller property. *Groups Geom. Dyn.*, 4(4):777–784, 2010.

[20] D. F. Holt and S. Rees. Artin groups of large type are shortlex automatic with regular geodesics. *Proc. Lond. Math. Soc. (3)*, 104(3):486–512, 2012.

[21] J. E. Hopcroft and J. D. Ullman. *Introduction to automata theory, languages, and computation*. Addison-Wesley Series in Computer Science. Addison-Wesley Publishing Co., Reading, Mass., 1979.

[22] D. E. Knuth and P. B. Bendix. Simple word problems in universal algebras. In *Computational Problems in Abstract Algebra (Proc. Conf., Oxford, 1967)*, pages 263–297. Pergamon, Oxford, 1970.

[23] R. E. Needham. Infinite complete group presentations. *J. Pure Appl. Algebra*, 110(2):195–218, 1996.

[24] W. D. Neumann and M. Shapiro. Automatic structures, rational growth, and geometrically finite hyperbolic groups. *Invent. Math.*, 120(2):259–287, 1995.

[25] M. H. A. Newman. On theories with a combinatorial definition of "equivalence.". *Ann. of Math. (2)*, 43:223–243, 1942.

[26] G. A. Noskov. Growth of certain non-positively curved cube groups. *European J. Combin.*, 21(5):659–666, 2000.

[27] G. A. Noskov. Bounded shortening in Coxeter complexes and buildings. In *Mathematical structures and modeling, No. 8 (Russian)*, pages 10–14. Omsk. Gos. Univ., Omsk, 2001.

[28] P. S. Novikov. *Ob algoritmičeskoĭ nerazrešimosti problemy toždestva slov v teorii grupp*. Izdat. Akad. Nauk SSSR, Moscow, 1955. Trudy Mat. Inst. Steklov. no. 44.

[29] F. Otto. On s-regular prefix-rewriting systems and automatic structures. In *Computing and combinatorics (Tokyo, 1999)*, volume 1627 of *Lecture Notes in Comput. Sci.*, pages 422–431. Springer, Berlin, 1999.

[30] C. Thiel. *Zur fast-Konvexität einiger nilpotenter Gruppen*, volume 234 of *Bonner Mathematische Schriften [Bonn Mathematical Publications]*. Universität Bonn, Mathematisches Institut, Bonn, 1992. Dissertation, Rheinische Friedrich-Wilhelms-Universität Bonn, Bonn, 1991.