

LOCAL SIMILARITY GROUPS WITH CONTEXT-FREE CO-WORD PROBLEM

DANIEL FARLEY

ABSTRACT. Let G be a group, and let S be a finite subset of G that generates G as a monoid. The *co-word problem* is the collection of words in the free monoid S^* that represent non-trivial elements of G .

A current conjecture, based originally on a conjecture of Lehnert and modified into its current form by Bleak, Matucci, and Neuhöffer, says that Thompson's group V is a universal group with context-free co-word problem. In other words, it is conjectured that a group has a context-free co-word problem exactly if it is a finitely generated subgroup of V .

Hughes introduced the class \mathcal{FSS} of groups that are determined by finite similarity structures. An \mathcal{FSS} group acts by local similarities on a compact ultrametric space. Thompson's group V is a representative example, but there are many others.

We show that \mathcal{FSS} groups have context-free co-word problem under a minimal additional hypothesis. As a result, we can specify a subfamily of \mathcal{FSS} groups that are potential counterexamples to the conjecture.

1. INTRODUCTION

Let G be a group, and let S be a finite subset that generates G as a monoid. The *word problem of G with respect to S* , denoted $WP_S(G)$, is the collection of all positive words w in S such that w represents the identity in G ; the *co-word problem of G with respect to S* , denoted $CoWP_S(G)$, is the set of all positive words that represent non-trivial elements of G . In this point of view, both the word and the co-word problem of G are formal languages, which suggests the question of placing these problems within the Chomsky hierarchy of languages.

Anisimov [1] proved that $WP_S(G)$ is a regular language if and only if G is finite. A celebrated theorem of Muller and Schupp [8] says that a finitely generated group G has context-free word problem if and only if it is virtually free. (A language is *context-free* if it is recognized by a pushdown automaton.) In this case, as noted in [5], the word problem is actually a deterministic context-free language. Shapiro [12] described sufficient conditions for a group to have a context-sensitive word problem.

Since the classes of regular, deterministic context-free, and context-sensitive languages are all closed under taking complements, it is of no additional interest to study groups with regular, deterministic context-free, or context-sensitive co-word problems, since the classes of groups in question do not change. The (non-deterministic) context-free languages are not closed under taking complements, however, so groups with context-free co-word problem are not (a priori, at least) the same as groups with context-free word problem.

2000 *Mathematics Subject Classification.* 20F10, 03D40 .

Key words and phrases. Thompson's groups, context-free languages, push-down automata.

Holt, Rees, Röver, and Thomas [5] introduced the class of groups with context-free co-word problem, denoted $co\mathcal{CF}$. They proved that all finitely generated virtually free groups are $co\mathcal{CF}$, and that the class of $co\mathcal{CF}$ groups is closed under taking finite direct products, passage to finitely generated subgroups, passage to finite index overgroups, and taking restricted wreath products with virtually free top group. They proved negative results as well: for instance, the Baumslag-Solitar groups $BS(m, n)$ are not $co\mathcal{CF}$ if $|m| \neq |n|$, and polycyclic groups are not $co\mathcal{CF}$ unless they are virtually abelian. They conjectured that $co\mathcal{CF}$ groups are not closed under the operation of taking free products, and indeed specifically conjectured that $\mathbb{Z} * \mathbb{Z}^2$ is not a $co\mathcal{CF}$ group.

Lehnert and Schweitzer [7] later showed that the Thompson group V is $co\mathcal{CF}$. Since V seems to contain many types of subgroups (among them all finite groups, all countable free groups, and all countably generated free abelian groups), this raised the possibility of showing that $\mathbb{Z} * \mathbb{Z}^2$ is $co\mathcal{CF}$ by embedding the latter group into V . Bleak and Salazar-Díaz [3], motivated at least in part by these considerations, proved that $\mathbb{Z} * \mathbb{Z}^2$ does not embed in V (leaving the conjecture from [5] open), and also established the existence of many embeddings into V . The basic effect of their embedding theorems is to show that the class \mathcal{V} of finitely generated subgroups of Thompson's group V is closed under the same operations as those from [5], as listed above.

The similarity between the classes \mathcal{V} and $co\mathcal{CF}$ seems to have led to the following conjecture:

Conjecture 1.1. The classes \mathcal{V} and $co\mathcal{CF}$ are the same; i.e., Thompson's group V is a universal $co\mathcal{CF}$ group.

Lehnert had conjectured in his thesis that a certain closely related group Q of quasi-automorphisms of the infinite binary tree is a universal $co\mathcal{CF}$ group. Bleak, Matucci, and Neuhöffer [2] established the existence of embeddings from Q to V and from V to Q . As a result, Lehnert's conjecture is equivalent Conjecture 1.1. We refer the reader to the excellent introductions of [2] and [3] for a more extensive discussion of these and related questions.

Here we show that many groups defined by finite similarity structures are contained in $co\mathcal{CF}$. The precise statement is as follows.

Main Theorem. *Let X be a compact ultrametric space endowed with a finite similarity structure Sim_X . Assume that there are only finitely many Sim_X -classes of balls.*

For any finitely generated subgroup G of $\Gamma(\text{Sim}_X)$ and finite subset S of G that generates G as a monoid, the co-word problem $\text{CoWP}_S(G) = \{w \in S^ \mid w \neq 1_G\}$ is a context-free language.*

The groups defined by finite similarity structures (or \mathcal{FSS} groups) were first studied by Hughes [6], who showed that all \mathcal{FSS} groups act properly on $\text{CAT}(0)$ cubical complexes and (therefore) have the Haagerup property. Farley and Hughes [4] proved that a class of \mathcal{FSS} groups have type \mathcal{F}_∞ . All of the latter groups satisfy the hypotheses of the main theorem, so all are also $co\mathcal{CF}$ groups. (We note that the main theorem also covers V as a special case.)

The class of \mathcal{FSS} groups is not well-understood, but we can specify a certain subclass that shows promise as a source of counterexamples to Conjecture 1.1. These are the Nekrashevych-Röver examples from [4] and [6]. The results of [4] show that most of these examples are not isomorphic to V (nor to the n -ary versions of V), and it is not difficult

to show that they do not contain V as a subgroup of finite index. It seems to be unknown whether there are any embeddings of these groups into V . Our main theorem therefore leaves Conjecture 1.1 open.

(Note that the Nekrashevych-Röver examples considered in [4] and [6] are not as general as the classes of groups from [11] and [9]; the finiteness of the similarity structures proves to be a somewhat restrictive hypothesis.)

The proof of the main theorem closely follows the work of Lehnert and Schweitzer [7]. We identify two main ingredients of their proof:

- (1) All of the groups satisfying the hypothesis of the main theorem admit *test partitions* (Definition 3.1). That is, there is a finite partition of the compact ultrametric space X into balls, such that every non-trivial word in the generators of G has a cyclic shift that moves at least one of the balls off of itself, and
- (2) for each pair of distinct balls (B_1, B_2) , where B_1 and B_2 are from the test partition, there is a “ (B_1, B_2) -witness automaton”, which is a pushdown automaton that can witness an element $g \in G$ moving part of B_1 into B_2 .

The main theorem follows very easily from (1) and (2). The proofs that (1) and (2) hold are complicated somewhat by the generality of our assumptions, but are already implicit in [7]. Most of the work goes into building the witness automata. We describe a stack language \mathcal{L} that the witness automata use to describe, store, and manipulate metric balls in X . One slight novelty (not present or necessary in [7]) is that the witness automata write functions from the similarity structure on their stacks and make partial computations using these functions.

We briefly describe the structure of the paper. Section 2 contains a summary of the relevant background, including string rewriting systems, pushdown automata, \mathcal{FSS} groups, and standing assumptions. Section 3 contains a proof that the groups G admit test partitions, as described above. Section 4 describes the stack language for the witness automata, and Section 5 gives the construction of the witness automata. Section 6 collects the ingredients of the previous sections into a proof of the main theorem.

2. BACKGROUND

2.1. String Rewriting Systems.

Definition 2.1. A *rewrite system* is a directed graph Γ . We write $a \rightarrow b$ if a and b are vertices of Γ and there is a directed edge from a to b . We write $a \dot{\rightarrow} b$ if there is a directed path from a to b . The rewrite system Γ is called *locally confluent* if whenever $a \rightarrow b$ and $a \rightarrow c$, there is some $d \in \Gamma^0$ such that $c \dot{\rightarrow} d$ and $b \dot{\rightarrow} d$. The rewrite system is *confluent* if whenever $a \dot{\rightarrow} b$ and $a \dot{\rightarrow} c$, there is some $d \in \Gamma^0$ such that $c \dot{\rightarrow} d$ and $b \dot{\rightarrow} d$. The rewrite system Γ is *terminating* if there is no infinite directed path in Γ . If a rewrite system is both terminating and confluent, then we say that it is *complete*. A vertex of Γ is called *reduced* if it is not the initial vertex of any directed edge in Γ .

Theorem 2.2. [10] *Every terminating, locally confluent rewrite system is complete.* □

Remark 2.3. The relation \rightarrow generates an equivalence relation on the vertices of Γ . It is not difficult to see that each equivalence class in this equivalence relation contains a unique reduced element in the event that Γ is complete.

Definition 2.4. Let Σ be a finite set, called an *alphabet*. Let \mathcal{L} be a subset of the free monoid Σ^* . Let \mathcal{R} be a collection of relations (or *rewriting rules*) of the form $w_1 \rightarrow w_2$,

where $w_1, w_2 \in \Sigma^*$. (Thus, the w_i are positive words in the alphabet Σ , either of which may be empty. The w_i are not required to be in \mathcal{L} .)

We define a *string rewriting system* as follows: The vertices are words from \mathcal{L} . For $u, v \in \mathcal{L}$, there is a directed edge $u \rightarrow v$ whenever there are words u', u'' such that $u = u'w_1u''$ and $v = u'w_2u''$, for some $w_1 \rightarrow w_2 \in \mathcal{R}$.

2.2. Pushdown Automata.

Definition 2.5. Let S and Σ be finite sets. The set S is the *input alphabet* and Σ is the *stack alphabet*. The stack alphabet contains a special symbol, $\#$, called the *initial stack symbol*.

A (*generalized*) *pushdown automaton* (or *PDA*) over S and Σ is a finite labelled directed graph Γ endowed with an *initial state* $v_0 \in \Gamma^0$ and a (possibly empty) collection of *terminal states* $T \subseteq \Gamma^0$. Each directed edge is labelled by a triple $(s, w', w'') \in (S \cup \{\epsilon\}) \times \Sigma^* \times \Sigma^*$, where ϵ denotes an empty string.

Each PDA accepts languages either by terminal state, or by empty stack, and this information must be specified as part of the automaton's definition. See Definition 2.7.

Definition 2.6. Let Γ be a pushdown automaton. We describe a class of directed paths in Γ , called the *valid paths*, by induction on length. The path of length 0 starting at the initial vertex $v_0 \in \Gamma^0$ is valid; its *stack value* is $\# \in \Sigma^*$. Let $e_1 \dots e_n$ ($n \geq 0$) be a valid path in Γ , where e_1 is the edge that is crossed first. Let e_{n+1} be an edge whose initial vertex is the terminal vertex of e_n ; we suppose that the label of e_{n+1} is (s, w', w'') . The path $e_1 e_2 \dots e_n e_{n+1}$ is also valid, provided that the stack value of $e_1 \dots e_n$ has w' as a prefix; that is, if the stack value of $e_1 \dots e_n$ has the form $w'\hat{w} \in \Sigma^*$. The stack value of $e_1 \dots e_{n+1}$ is then $w''\hat{w}$. We let $\text{val}(p)$ denote the stack value of a valid path p .

The *label* of a valid path $e_1 \dots e_n$ is $s_n \dots s_1$, where s_i is the first coordinate of the label for e_i (an element of S , or the empty string). The label of a valid path p will be denoted $\ell(p)$.

Definition 2.7. Let Γ be a PDA. The *language* \mathcal{L}_Γ *accepted by* Γ is either

- (1) $\mathcal{L}_\Gamma = \{w \in S^* \mid w = \ell(p) \text{ for some valid path } p \text{ with } \text{val}(p) = \epsilon\}$, if Γ accepts by empty stack, or
- (2) $\mathcal{L}_\Gamma = \{w \in S^* \mid w = \ell(p) \text{ for some valid path } p \text{ whose terminal vertex is in } T\}$, if Γ accepts by terminal state.

Definition 2.8. A subset of the free monoid S^* is called a (*non-deterministic*) *context-free language* if it is \mathcal{L}_Γ , for some pushdown automaton Γ .

Remark 2.9. The class of languages that are accepted by empty stack (in the above sense) is the same as the class of languages that are accepted by terminal state. That is, given an automaton Γ' that accepts a language \mathcal{L} by empty stack, there is another automaton Γ'' that accepts \mathcal{L} by terminal state (and conversely).

Remark 2.10. All of the automata considered in this paper will accept by empty stack.

The functioning of an automaton Γ can be described in plain language as follows. We begin with a word $s_n \dots s_1 \in S^*$ written on an input tape, and the word $\# \in \Sigma^*$ written on the memory tape (or stack). We imagine the stack as a sequence of boxes extending indefinitely to our left, all empty except for the rightmost one, which has $\#$ written in it. Our automaton reads the input tape from right to left. It can read and write on the stack

only from the left (i.e., from the leftmost nonempty box). Beginning in the initial state $v_0 \in \Gamma_0$, it can follow any directed edge e it chooses, provided that it meets the proper prerequisites: if the label of e is (s, w', w'') , then s must be the rightmost remaining symbol on the input tape, and the word $w' \in \Sigma^*$ must be a prefix of the word written on the stack. If these conditions are met, then it can cross the edge e into the next state, simultaneously erasing the letter s from the input tape, erasing w' from the left end of the stack, and then writing w'' on the left end of the stack. The original input word is accepted if the automaton can reach a state with nothing left on its input tape, and nothing on its stack (not even the symbol $\#$).

We note that a label such as $(\epsilon, \epsilon, w'')$ describes an empty set of prerequisites. Such an arrow may always be crossed, without reading the input tape or the stack, no matter whether one or the other is empty.

2.3. Review of ultrametric spaces and finite similarity structures. We now give a quick review (without proofs) of finite similarity structures on compact ultrametric spaces, as defined in [6]. Most of this subsection is taken from [4].

Definition 2.11. An *ultrametric space* is a metric space (X, d) such that

$$d(x, y) \leq \max\{d(x, z), d(z, y)\},$$

for all $x, y, z \in X$.

Lemma 2.12. *Let X be an ultrametric space.*

- (1) *Let $N_\epsilon(x)$ be an open metric ball in X . If $y \in N_\epsilon(x)$, then $N_\epsilon(x) = N_\epsilon(y)$.*
- (2) *If B_1 and B_2 are open metric balls in X , then either the balls are disjoint, or one is contained in the other.*
- (3) *If X is compact, then each open ball B is contained in at most finitely many distinct open balls of X , and these form an increasing sequence:*

$$B = B_1 \subsetneq B_2 \subsetneq \dots \subsetneq B_n = X.$$

- (4) *If X is compact and x is not an isolated point, then each open ball $N_\epsilon(x)$ is partitioned by its maximal proper open subballs, which are finite in number.*

□

Convention 2.13. Throughout this paper, “ball” will always mean “open ball”.

Definition 2.14. Let $f : X \rightarrow Y$ be a function between metric spaces. We say that f is a *similarity* if there is a constant $C > 0$ such that $d_Y(f(x_1), f(x_2)) = Cd_X(x_1, x_2)$, for all x_1 and x_2 in X .

Definition 2.15. A *finite similarity structure for X* is a function Sim_X that assigns to each ordered pair B_1, B_2 of balls in X a (possibly empty) set $\text{Sim}_X(B_1, B_2)$ of surjective similarities $B_1 \rightarrow B_2$ such that whenever B_1, B_2, B_3 are balls in X , the following properties hold:

- (1) (Finiteness) $\text{Sim}_X(B_1, B_2)$ is a finite set.
- (2) (Identities) $\text{id}_{B_1} \in \text{Sim}_X(B_1, B_1)$.
- (3) (Inverses) If $h \in \text{Sim}_X(B_1, B_2)$, then $h^{-1} \in \text{Sim}_X(B_2, B_1)$.
- (4) (Compositions) If $h_1 \in \text{Sim}_X(B_1, B_2)$ and $h_2 \in \text{Sim}_X(B_2, B_3)$, then $h_2 h_1 \in \text{Sim}_X(B_1, B_3)$.

(5) (Restrictions) If $h \in \text{Sim}_X(B_1, B_2)$ and $B_3 \subseteq B_1$, then

$$h|_{B_3} \in \text{Sim}_X(B_3, h(B_3)).$$

Definition 2.16. A homeomorphism $h: X \rightarrow X$ is *locally determined by* Sim_X provided that for every $x \in X$, there exists a ball B' in X such that $x \in B'$, $h(B')$ is a ball in X , and $h|_{B'} \in \text{Sim}(B', h(B'))$.

Definition 2.17. The *finite similarity structure (FSS) group* $\Gamma(\text{Sim}_X)$ is the set of all homeomorphisms $h: X \rightarrow X$ such that h is locally determined by Sim_X .

Remark 2.18. The fact that $\Gamma(\text{Sim}_X)$ is a group under composition is due to Hughes [6].

Definition 2.19. ([6], Definition 3.6) If $\gamma \in \Gamma(\text{Sim}_X)$, then we can choose a partition of X by balls B such that, for each B , $\gamma(B)$ is a ball and $\gamma|_B \in \text{Sim}_X(B, \gamma(B))$. Each element of this partition is called a *region* for g .

2.4. Standing Assumptions. In this section, we set conventions that hold for the rest of the paper.

Definition 2.20. We say that two balls B_1 and B_2 are in the same *Sim_X-class* if the set $\text{Sim}_X(B_1, B_2)$ is non-empty.

Convention 2.21. We assume that X is a compact ultrametric space with finite similarity structure Sim_X . We assume that there are only finitely many Sim_X -classes of balls, represented by

$$\tilde{B}_1, \dots, \tilde{B}_k.$$

We let $[B]$ denote the Sim_X -class of a ball B , and let $X = \tilde{B}_1$.

Each ball $B \subseteq X$ is related to exactly one of the \tilde{B}_i . We choose (and fix) an element $f_B \in \text{Sim}_X(\tilde{B}_i, B)$. We choose $f_{\tilde{B}_i} = id_{\tilde{B}_i}$.

Each ball \tilde{B}_i has a finite collection of maximal proper subballs, denoted

$$\tilde{B}_{i1}, \dots, \tilde{B}_{i\ell_i}.$$

This numbering (of the balls \tilde{B}_i and their maximal proper subballs) is fixed throughout the rest of the argument. We let $\ell = \max\{\ell_1, \dots, \ell_k\}$.

We will for the most part freely recycle the subscripts k and ℓ . However, for the reader's convenience, we note ahead of time that we will use k and ℓ with the above meaning in Definitions 4.1, 4.7, 4.9, and 5.1.

Convention 2.22. We will let G denote a finitely generated subgroup of $\Gamma(\text{Sim}_X)$ (see Definition 2.17). We choose a finite set $S \subseteq G$ that generates G as a monoid, i.e., each element $g \in G$ can be expressed in the form $g = s_1 \dots s_n$, where $s_i \in S$, $n \geq 0$, and only positive powers of the s_i are used. We choose (and fix) regions for each $s \in S$.

3. TEST PARTITIONS

Definition 3.1. Let \mathcal{P} be a finite partition of X . We say that \mathcal{P} is a *test partition* if, for any word $s_1 \dots s_n$ in the generators S , whenever

$$s_j \dots s_n s_1 \dots s_{j-1}(P) = P,$$

for all $j \in \{1, \dots, n\}$ and $P \in \mathcal{P}$, then $s_1 \dots s_n = 1_G$.

Lemma 3.2. *If X is a compact ultrametric space and $\epsilon > 0$, then $\{N_\epsilon(x) \mid x \in X\}$ is a finite partition of X by open balls.*

Proof. This follows easily from Lemma 2.12(1). \square

Definition 3.3. Let $\epsilon_1 > 0$ be chosen so that

$$\epsilon_1 < \min\{\text{diam}(B) \mid B \text{ is a defining ball in } S\}.$$

We let $\mathcal{P}_{big} = \{N_{\epsilon_1}(x) \mid x \in X\}$. This is the *big partition*.

Note that, for each $s \in S$ and $P \in \mathcal{P}_{big}$, P is contained in a unique region of s .

Lemma 3.4. *Let B be a compact ultrametric space, and let Γ be a finite group of isometries of B . There is an $\epsilon > 0$ such that if $\gamma \in \Gamma$ acts trivially on $\{N_\epsilon(x) \mid x \in B\}$, then $\gamma = 1_\Gamma$.*

Proof. For each nontrivial $\gamma \in \Gamma$, there is $x_\gamma \in X$ such that $\gamma(x_\gamma) \neq x_\gamma$. We choose $\epsilon_\gamma > 0$ satisfying

$$N_{\epsilon_\gamma}(x_\gamma) \cap N_{\epsilon_\gamma}(\gamma(x_\gamma)) = \emptyset.$$

We set $\epsilon = \min\{\epsilon_\gamma \mid \gamma \in \Gamma - \{1_\Gamma\}\}$. Now suppose that $\gamma \neq 1_\Gamma$ and γ acts trivially on $\{N_\epsilon(x) \mid x \in X\}$. Thus $\gamma(N_\epsilon(x_\gamma)) = N_\epsilon(x_\gamma)$, so $N_\epsilon(\gamma(x_\gamma)) \cap N_\epsilon(x_\gamma) \neq \emptyset$, but

$$N_\epsilon(\gamma(x_\gamma)) \cap N_\epsilon(x_\gamma) \subseteq N_{\epsilon_\gamma}(x_\gamma) \cap N_{\epsilon_\gamma}(\gamma(x_\gamma)) = \emptyset,$$

a contradiction. \square

Definition 3.5. Write $\mathcal{P}_{big} = \{B_1, \dots, B_\ell\}$. For each B_i ($i \in \{1, \dots, \ell\}$), we can choose $\hat{\epsilon}_i$ to meet the conditions satisfied by ϵ in the previous lemma, for $\Gamma = \text{Sim}_X(B_i, B_i)$. Let $\epsilon_2 = \min\{\hat{\epsilon}_1, \dots, \hat{\epsilon}_\ell\}$. Let $\mathcal{P}_{small} = \{N_{\epsilon_2}(x) \mid x \in X\}$. This is the *small partition*.

Proposition 3.6. *The small partition \mathcal{P}_{small} is a test partition.*

Proof. Let $s_1 \dots s_n$ be a word in the generators S ; we assume $s_1 \dots s_n \neq 1$. We suppose, for a contradiction, that for all $P \in \mathcal{P}_{small}$,

$$s_j \dots s_n s_1 \dots s_{j-1}(P) = P,$$

for all $j \in \{1, \dots, n\}$. Since $s_1 \dots s_n \neq 1$, we can find $x \in X$ such that $s_1 \dots s_n(x) \neq x$.

Sublemma 3.7. *Fix $s_1 \dots s_n \in S^+$. For each $x \in X$, there is an open ball B , with $x \in B$, such that:*

- (1) $s_j \dots s_n(B)$ lies in a region of s_{j-1} , for $j = 2, \dots, n+1$, and
- (2) $s_j \dots s_n(B) \in \mathcal{P}_{big}$ for at least one $j \in \{1, \dots, n\}$.

Proof. We first prove that, for any $x \in X$, there is a ball neighborhood B of x satisfying (1). We choose and fix $x \in X$.

Consider the elements $s_1, s_2, \dots, s_n \in G$. We first observe that there is a constant $C \geq 1$ such that if any ball B' lies inside a region for s_j (for any $j \in \{1, \dots, n\}$), then s_j stretches B' by a factor of no more than C . Next, observe that there is a constant D such that any ball of diameter less than or equal to D lies inside of a region for s_j , for all $j \in \{1, \dots, n\}$. It follows easily that any ball of diameter less than D/C^{n-1} satisfies (1); we can clearly choose some such ball, B_1 to be a neighborhood of x . We note that if a ball satisfies (1), then so does every subball.

Let

$$B_1 \subsetneq B_2 \subsetneq B_3 \subsetneq \dots \subsetneq B_m = X$$

be the collection of all balls containing B_1 . (Thus, each B_i is a maximal proper subball inside B_{i+1} , for $i = 1, \dots, m-1$.) There is a maximal ball B_α , $\alpha \in \{1, \dots, m\}$, such that B_α satisfies (1).

If $\alpha = m$, then the entire composition $s_1 \dots s_n \in \text{Sim}_X(X, X)$. We then take $P \in \mathcal{P}_{big}$ such that $s_1 \dots s_n(x) \in P$. The required B is $(s_1 \dots s_n)^{-1}(P)$.

Now assume that $\alpha < m$. There is some $j \in \{1, \dots, n\}$ such that $s_{j+1} \dots s_n(B_{\alpha+1})$ is a ball and

$$(s_{j+1} \dots s_n)|_{B_{\alpha+1}} \in \text{Sim}_X(B_{\alpha+1} s_{j+1} \dots s_n(B_{\alpha+1})),$$

but $(s_{j+1} \dots s_n)(B_{\alpha+1})$ properly contains a region for s_j ; let $\hat{B}_1, \dots, \hat{B}_\beta$ be the regions of s_j that are contained in $(s_{j+1} \dots s_n)(B_{\alpha+1})$. We must have $\hat{B}_\delta \subseteq (s_{j+1} \dots s_n)(B_\alpha)$ for some δ (by maximality of $(s_{j+1} \dots s_n)(B_\alpha)$ in $(s_{j+1} \dots s_n)(B_{\alpha+1})$); the reverse containment $(s_{j+1} \dots s_n)(B_\alpha) \subseteq \hat{B}_\delta$ follows, since $(s_{j+1} \dots s_n)(B_\alpha)$ is contained in a region for s_j by our assumptions.

Now note that \hat{B}_δ is partitioned by elements of \mathcal{P}_{big} ; there is some $P \subseteq \hat{B}_\delta$ such that $s_{j+1} \dots s_n(x) \in P$. We have that the map $s_{j+1} \dots s_n : B_\alpha \rightarrow \hat{B}_\delta$ is a map from the similarity structure. The required ball B is $(s_{j+1} \dots s_n)^{-1}(P)$. \square

Apply the sub lemma to x : there is B (an open ball) with the given properties. Let j be such that

$$(s_j \dots s_n)|_B \in \text{Sim}_X(B, s_j \dots s_n(B)),$$

where $s_j \dots s_n(B) \in \mathcal{P}_{big}$.

Since $s_j \dots s_n(B)$ is invariant under every cyclic permutation of $s_1 \dots s_n$ by our assumption,

$$s_j \dots s_n s_1 \dots s_{j-1}(s_j \dots s_n(B)) = s_j \dots s_n(B),$$

so $s_1 \dots s_n(B) = B$.

Our assumptions imply that $(s_j \dots s_n)|_B : B \rightarrow s_j \dots s_n(B) = P$ and $(s_1 \dots s_{j-1})|_P : P \rightarrow B$ are both in Sim_X , and both are bijections.

Consider $(s_j \dots s_n s_1 \dots s_{j-1})|_P \in \text{Sim}_X(P, P)$. It must be that $(s_j \dots s_n s_1 \dots s_{j-1})_P \neq 1_P$; if $(s_j \dots s_n s_1 \dots s_{j-1})_P = 1$, then

$$s_j \dots s_n s_1 \dots s_{j-1}(s_j \dots s_n)(x) = s_j \dots s_n(x),$$

which implies that $s_1 \dots s_n(x) = x$, a contradiction.

Now, since $(s_j \dots s_n s_1 \dots s_{j-1})_P \neq 1_P$, it moves some element of \mathcal{P}_{small} . \square

4. A LANGUAGE FOR Sim_X

In this section, we introduce languages \mathcal{L}_{red} and \mathcal{L} . The language \mathcal{L} will serve as the stack language for the witness automata of Section 5. The language \mathcal{L}_{red} consists of the reduced elements of \mathcal{L} ; it is useful because there is a one-to-one correspondence between elements of \mathcal{L}_{red} and metric balls in X .

4.1. The languages \mathcal{L}_{red} and \mathcal{L} .

Definition 4.1. We define a language \mathcal{L}_{red} as follows. The alphabet Σ for \mathcal{L}_{red} consists of the symbols:

- (1) $\#$, the initial stack symbol;
- (2) $A_{1, \emptyset}$;
- (3) $A_{i, n}$, $i \in \{1, \dots, k\}$, $n \in \{1, \dots, \ell\}$.

(We refer the reader to Convention 2.21 for the meanings of k and ℓ .) The language \mathcal{L}_{red} consists of all words of the form

$$A_{1,\emptyset}A_{i_1,n_1}A_{i_2,n_2}\cdots A_{i_m,n_m}\#,$$

where $m \geq 0$ and $[\tilde{B}_{i_{s-1}n_s}] = [\tilde{B}_{i_s}]$ for $s = 1, \dots, m$. (Here, and in what follows, we make the convention that $i_0 = 1$; i.e., that $X = \tilde{B}_{i_0}$.)

The language \mathcal{L} also uses symbols of the form $[f]$, where $f \in \text{Sim}_X(\tilde{B}_i, \tilde{B}_i)$. The general element of \mathcal{L} takes the form

$$A_{1,\emptyset}w_0A_{i_1,n_1}w_1A_{i_2,n_2}w_2\cdots A_{i_{m-1},n_{m-1}}w_{m-1}A_{i_m,n_m}w_m\#,$$

where each w_j ($j \in \{0, 1, \dots, m\}$) is a word in the symbols $\{[f] \mid f \in \text{Sim}_X(\tilde{B}_{i_j}, \tilde{B}_{i_j})\}$, and some or all of the w_j might be empty.

Remark 4.2. The letter $A_{i,n}$ signifies a ball of similarity class $[\tilde{B}_i]$; the n signifies that it is the n th maximal proper subball of the ball before it in the sequence. The letter $A_{1,\emptyset}$ signifies the top ball, X . Refer to Convention 2.21 for the significance of i and n .

The condition $[\tilde{B}_{i_{s-1}n_s}] = [\tilde{B}_{i_s}]$ for $s = 1, \dots, m$ is designed to insure that each ball has the correct type; i.e., that the sequence encodes consistent information about the similarity types of subballs.

Definition 4.3. Let \mathcal{B}_X denote the collection of all metric balls in X . We define an *evaluation map* $E : \mathcal{L} \rightarrow \mathcal{B}_X$ by sending

$$w = A_{1,\emptyset}w_0A_{i_1,n_1}w_1A_{i_2,n_2}w_2\cdots A_{i_{m-1},n_{m-1}}w_{m-1}A_{i_m,n_m}w_m\#,$$

to

$$E(w) = \left(f_{\tilde{B}_{1n_1}} \circ f_{w_0} \circ f_{\tilde{B}_{i_1n_2}} \circ f_{w_1} \circ \cdots \circ f_{w_{m-1}} \circ f_{\tilde{B}_{i_{m-1}n_m}} \right) (\tilde{B}_{i_m}),$$

where $f_{w_i} = f_{j_1} \circ f_{j_2} \circ \cdots \circ f_{j_\alpha}$ if $w_i = [f_{j_1}][f_{j_2}]\cdots[f_{j_\alpha}]$.

Definition 4.4. Let $w, w' \in \mathcal{L}$. We say that w' is a *prefix* of w if w' with the initial stack symbol $\#$ omitted is a prefix of w in the usual sense; that is $w = w'u$, for some string $u \in \Sigma^*$.

Proposition 4.5. *The function $E : \mathcal{L}_{red} \rightarrow \mathcal{B}_X$ is a bijection.*

Moreover, a word $w' \in \mathcal{L}_{red}$ is a proper prefix of $w \in \mathcal{L}_{red}$ if and only if $E(w)$ is a proper subball of $E(w')$, and w' is a maximal proper prefix of w if and only if $E(w)$ is a maximal proper subball of $E(w')$.

Proof. We first prove surjectivity. Let B be a ball in X . We let

$$B = B_m \subseteq B_{m-1} \subseteq B_{m-2} \subseteq \cdots \subseteq B_0 = X$$

be the collection of all balls in X that contain B_m . (Thus, B_i is a maximal proper subball in B_{i-1} for $i = 1, \dots, m$.)

In the diagram

$$\begin{array}{ccccccc} \tilde{B}_{i_m} & & \tilde{B}_{i_{m-1}} & & \tilde{B}_{i_{m-2}} & & \cdots & & \tilde{B}_0 \\ f_{B_m} \downarrow & & f_{B_{m-1}} \downarrow & & f_{B_{m-2}} \downarrow & & & & f_X \downarrow \\ B_m & \longrightarrow & B_{m-1} & \longrightarrow & B_{m-2} & \longrightarrow & \cdots & \longrightarrow & X, \end{array}$$

the balls \tilde{B}_{i_j} and the maps f_{B_j} are the ones given in Convention 2.21; the unlabeled arrows are inclusions. Note, in particular, that the maps f_{B_j} are bijections taken from the Sim_X -structure, and that f_X is the identity map. If we follow the arrows from \tilde{B}_{i_j} to $\tilde{B}_{i_{j-1}}$, the corresponding composition is a member of Sim_X that carries the ball \tilde{B}_{i_j} to a maximal proper subball of $\tilde{B}_{i_{j-1}}$. Supposing that the number of the latter maximal proper subball is n_j (see Convention 2.21), we obtain a diagram

$$\begin{array}{ccccccc} \tilde{B}_{i_m} & \xrightarrow{I_m} & \tilde{B}_{i_{m-1}} & \xrightarrow{I_{m-1}} & \tilde{B}_{i_{m-2}} & \xrightarrow{I_{m-2}} & \dots \xrightarrow{I_1} \tilde{B}_0 \\ f_{B_m} \downarrow & & f_{B_{m-1}} \downarrow & & f_{B_{m-2}} \downarrow & & f_X \downarrow \\ B_m & \longrightarrow & B_{m-1} & \longrightarrow & B_{m-2} & \longrightarrow & \dots \longrightarrow X, \end{array}$$

where $I_j = f_{\tilde{B}_{i_{j-1}n_j}}$, for $j = 1, \dots, m$. This diagram commutes “up to images”: that is, if we start at a given node in the diagram, then the image of that first node in any other node is independent of path. (The diagram is not guaranteed to commute in the usual sense.) Set $w = A_{1,\emptyset}A_{i_1,n_1} \dots A_{i_m,n_m}$. We note that

$$\begin{aligned} E(w) &= (I_1 \circ I_2 \circ \dots \circ I_m)(\tilde{B}_{i_m}) \\ &= (f_X \circ I_1 \circ I_2 \circ \dots \circ I_m)(\tilde{B}_{i_m}) \\ &= f_{B_m}(\tilde{B}_{i_m}) \\ &= B_m, \end{aligned}$$

where the first equality is the definition of $E(w)$, the second follows since $f_X = \text{id}_X$, the third follows from the commutativity of the diagram up to images, and the fourth follows from surjectivity of f_{B_m} . This proves that $E : \mathcal{L}_{red} \rightarrow \mathcal{B}_X$ is surjective.

Before proving injectivity of E , we note that, for a given

$$w = A_{1,\emptyset}A_{i_1,n_1} \dots A_{i_m,n_m}$$

and associated

$$E(w) = \left(f_{\tilde{B}_{i_0n_1}} \circ f_{\tilde{B}_{i_1n_2}} \circ \dots \circ f_{\tilde{B}_{i_{m-1}n_m}} \right) (\tilde{B}_{i_m}),$$

each of the functions $f_{\tilde{B}_{i_{s-1}n_s}} : \tilde{B}_{i_s} \rightarrow \tilde{B}_{i_{s-1}n_s}$ maps its domain onto a proper subball of its codomain. As a result, a word w' is a proper prefix of w if and only if $E(w)$ is a proper subball of $E(w')$, and w' is a maximal proper prefix of w if and only if $E(w)$ is a maximal proper subball of $E(w')$.

Suppose now that $E(w_1) = E(w_2)$, for some $w_1, w_2 \in \mathcal{L}_{red}$, $w_1 \neq w_2$. By the above discussion, we can assume that neither w_1 nor w_2 is a prefix of the other. Let w_3 be the largest common prefix of w_1 and w_2 . Let $E(w_3) = B$. Since $w_1 = w_3w'$ and $w_2 = w_3w''$ for non-trivial strings w' and w'' with different initial symbols, $E(w_1)$ and $E(w_2)$ are disjoint proper subballs of $E(w_3)$. \square

Definition 4.6. Let B be a ball in X . The *address* of B is the inverse image of B under the evaluation map $E : \mathcal{L}_{red} \rightarrow \mathcal{B}_X$, but with the initial stack symbol omitted. We write $\text{addr}(B)$.

4.2. A string rewriting system based on \mathcal{L} . In this subsection, we describe a string rewriting system with underlying vertex set \mathcal{L} . The witness automata of Section 5 will use this rewrite system to perform partial calculations in Sim_X on their stacks.

Definition 4.7. Define

$$[\cdot] : \bigcup_{(B_1, B_2)} \text{Sim}_X(B_1, B_2) \rightarrow \{[f] \mid f \in \text{Sim}_X(\tilde{B}_j, \tilde{B}_j), j \in \{1, \dots, k\}\}$$

by the rule $[h] = [f_{B_2}^{-1} h f_{B_1}]$, for $h \in \text{Sim}_X(B_1, B_2)$. (We recall that k is the number of Sim_X -classes of balls in X ; see Convention 2.21.) The union is over all pairs of balls $B_1, B_2 \subseteq X$.

If $[h] = [f]$, where $f \in \text{Sim}_X(\tilde{B}_j, \tilde{B}_j)$ for some $j \in \{1, \dots, k\}$, then f is the *standard representative* of h , and $[f]$ is the *standard form* for $[h]$.

Remark 4.8. If $f \in \text{Sim}_X(\tilde{B}_j, \tilde{B}_j)$ for some $j \in \{1, \dots, k\}$, then we sometimes confuse $[f]$ with f itself; this is justified by our choices in Convention 2.21.

Definition 4.9. Define a string rewriting system $(\mathcal{L}, \rightarrow)$ as follows. The vertices are elements of the language \mathcal{L} . There are four families of rewriting rules:

(1) (Restriction)

$$[f]A_{i_s, n_s} \rightarrow A_{i_s, f(n_s)}[f]_{|\tilde{B}_{i_{s-1} n_s}},$$

where $[f]$ is a standard form; i.e., $f \in \text{Sim}_X(\tilde{B}_{i_{s-1}}, \tilde{B}_{i_{s-1}})$;

(2) (Group multiplication)

$$[f_1][f_2] \rightarrow [f_1 \circ f_2],$$

where $f_1, f_2 \in \text{Sim}_X(\tilde{B}_j, \tilde{B}_j)$, for some $j \in \{1, \dots, k\}$;

(3) (Absorption)

$$[f]\# \rightarrow \#,$$

for arbitrary $[f]$;

(4) (Identities)

$$[id_{\tilde{B}_j}] \rightarrow \emptyset,$$

for $j = 1, \dots, k$.

Remark 4.10. We note that the total number of the above rules is finite, since there are only finitely many Sim_X -classes of balls.

Proposition 4.11. *The string rewriting system $(\mathcal{L}, \rightarrow)$ is locally confluent and terminating. Each reduced element of \mathcal{L} is in \mathcal{L}_{red} . The function E is constant on equivalence classes modulo \rightarrow .*

Proof. It is clear that each reduced element of \mathcal{L} is in \mathcal{L}_{red} , and that $(\mathcal{L}, \rightarrow)$ is terminating. Local confluence of $(\mathcal{L}, \rightarrow)$ is clear, except for one case, which we will now consider.

Suppose that $w \in \mathcal{L}$ contains a substring of the form $[f][g]A_{i_s n_s}$. We can apply two different overlapping rewrite rules to w , one sending $[f][g]A_{i_s n_s}$ to $[f \circ g]A_{i_s n_s}$, and the other sending $[f][g]A_{i_s n_s}$ to $[f]A_{i_s g(n_s)}[g]_{|\tilde{B}_{i_{s-1} n_s}}$. We need to show that $[f \circ g]A_{i_s n_s}$ and $[f]A_{i_s g(n_s)}[g]_{|\tilde{B}_{i_{s-1} n_s}}$ flow to a common string. Note that

$$[f \circ g]A_{i_s n_s} \rightarrow A_{i_s f(g(n_s))}[(f \circ g)_{|\tilde{B}_{i_{s-1} n_s}}],$$

and

$$[f]A_{i_s g(n_s)}[g|_{\tilde{B}_{i_s-1}n_s}] \rightarrow A_{i_s} f(g(n_s))[f|_{\tilde{B}_{i_s-1}g(n_s)}][g|_{\tilde{B}_{i_s-1}n_s}].$$

It therefore suffices to demonstrate that the maps $[(f \circ g)|_{\tilde{B}_{i_s-1}n_s}]$ and $[f|_{\tilde{B}_{i_s-1}g(n_s)}][g|_{\tilde{B}_{i_s-1}n_s}]$ are equal. But this follows from the commutativity of the following diagram:

$$\begin{array}{ccccc} \tilde{B}_{i_s} & \xrightarrow{[g]} & \tilde{B}_{i_s} & \xrightarrow{[f]} & \tilde{B}_{i_s} \\ \downarrow & & \downarrow & & \downarrow \\ \tilde{B}_{i_s-1}n_s & \xrightarrow{g} & g(\tilde{B}_{i_s-1}n_s) & \xrightarrow{f} & (f \circ g)(\tilde{B}_{i_s-1}n_s), \end{array}$$

where the vertical arrows are the canonical identifications from Convention 2.21 (e.g., the first vertical arrow is $f|_{\tilde{B}_{i_s-1}n_s}$). It now follows that $(\mathcal{L}, \rightarrow)$ is locally confluent and terminating.

We now prove that E is constant on the equivalence classes modulo \rightarrow . It is clear that applications of rules (2)-(4) do not change the value of E ; we check that (1) also does not change the value of E . Suppose we are given

$$\tilde{B}_{i_m} \xrightarrow{I_m} \tilde{B}_{i_{m-1}} \xrightarrow{I_{m-1}} \dots \longrightarrow \tilde{B}_{i_1} \xrightarrow{I_1} \tilde{B}_{i_0} = X,$$

where each $I_j = f_{w_{m-1}} \circ f_{\tilde{B}_{i_{m-1}}n_m}$, and $(I_1 \circ \dots \circ I_m)(\tilde{B}_{i_m})$ is therefore $E(w)$, for $w \in \mathcal{L}$ in the form given in Definition 4.1. We pick a particular $I_\alpha = f_{w_{\alpha-1}} \circ f_{\tilde{B}_{i_{\alpha-1}}n_\alpha}$, for some $\alpha \in \{1, \dots, m\}$. We note that the map I_α corresponds to the substring $w_{\alpha-1}A_{i_\alpha n_\alpha}$ of w . We may assume that $w_{\alpha-1}$ has length 1 (after applying rewriting rules of the form (2)); we write $[f]$ in place of $f_{w_{\alpha-1}}$, where $[f]$ is in standard form. The result of applying a rewrite rule of type (1) to $w_{\alpha-1}A_{i_\alpha n_\alpha}$ is the string $A_{i_\alpha} f(n_\alpha)[f|_{\tilde{B}_{i_{\alpha-1}}n_\alpha}]$. The latter string corresponds to the map

$$I'_\alpha = f_{\tilde{B}_{i_{\alpha-1}}f(n_\alpha)} \circ [f|_{\tilde{B}_{i_{\alpha-1}}n_\alpha}],$$

so we must show that $I'_\alpha = I_\alpha$. We consider the commutative diagram

$$\begin{array}{ccccc} \tilde{B}_{i_\alpha} & \longrightarrow & \tilde{B}_{i_{\alpha-1}n_\alpha} & \longrightarrow & \tilde{B}_{i_{\alpha-1}} \\ \downarrow [f] & & \downarrow f|_{\tilde{B}_{i_{\alpha-1}}n_\alpha} & & \downarrow f \\ \tilde{B}_{i_\alpha} & \longrightarrow & \tilde{B}_{i_{\alpha-1}f(n_\alpha)} & \longrightarrow & \tilde{B}_{i_{\alpha-1}}, \end{array}$$

where the leftmost horizontal arrows are the canonical identifications of Convention 2.21 and the rightmost horizontal arrows are inclusions. If we follow the arrows in this rectangle from the upper left corner, down the left side, and across the bottom, the resulting map is I'_α ; if we follow the arrows along the top and right side, the resulting map is I_α . This proves that $I'_\alpha = I_\alpha$, as required. \square

4.3. The Action of Sim_X on \mathcal{L} .

Definition 4.12. Let $f \in \text{Sim}_X(B', B'')$, where B', B'' are arbitrary balls in X . Suppose that $\text{addr}(B') = \hat{w}$ and $\text{addr}(B'') = \tilde{w}$, where

$$\hat{w} = A_{1,\emptyset}A_{i_1 n_1} \dots A_{i_m n_m} \# \quad \text{and} \quad \tilde{w} = A_{1,\emptyset}A_{j_1 \ell_1} \dots A_{j_t \ell_t} \#.$$

Let $w \in \mathcal{L}$. For a word $w \in \mathcal{L}$, define a partial function $\phi_f : \mathcal{L} \rightarrow \mathcal{L}$ by the rule

$$\phi_f(w) = \tilde{w}[f]w'$$

if w has \hat{w} as a prefix, i.e., if $w = \hat{w}w'$ for some string w' . Otherwise, $\phi_f(w)$ is undefined.

Proposition 4.13. *The expression $\phi_f(w)$ is defined if and only if $\text{addr}(B')$ is a prefix of $E(w)$. If $\phi_f(w)$ is defined, then*

$$E(\phi_f(w)) = f(E(w)).$$

Proof. The first statement is straightforward.

Assume first that $w \in \mathcal{L}_{red}$. We have $w = A_{1,\emptyset}A_{i_1n_1} \dots A_{i_mn_m}A_{i_{m+1}n_{m+1}} \dots A_{i_un_u}\#$. If we write I_v in place of $f_{\tilde{B}_{i_{v-1}n_v}}$ for $v \in \{1, \dots, u\}$, then $E(w) = (I_1 \circ \dots \circ I_u)(\tilde{B}_{i_u})$. If

$$B_u \subseteq B_{u-1} \subseteq \dots \subseteq B_1 \subseteq X$$

is the sequence of all balls containing $B_u = E(w)$ (so that each ball is necessarily a maximal proper subball in the next), we have

$$\begin{array}{ccccccc} \tilde{B}_{i_u} & \xrightarrow{I_u} & \tilde{B}_{i_{u-1}} & \xrightarrow{I_{u-1}} & \tilde{B}_{i_{u-2}} & \xrightarrow{I_{u-2}} & \dots & \xrightarrow{I_{m+1}} & \tilde{B}_{i_m} \\ \downarrow & & \downarrow & & \downarrow & & & & \downarrow \\ B_u & \longrightarrow & B_{u-1} & \longrightarrow & B_{u-2} & \longrightarrow & \dots & \longrightarrow & B' \end{array}$$

where the bottom horizontal arrows are inclusions, the vertical arrows are the canonical maps, and the diagram commutes up to images. Concatenating diagrams, we have

$$\begin{array}{ccccccccccc} \tilde{B}_{i_u} & \xrightarrow{I_u} & \dots & \xrightarrow{I_{m+1}} & \tilde{B}_{i_m} & \xrightarrow{[f]} & \tilde{B}_{j_t} & \xrightarrow{I'_t} & \dots & \xrightarrow{I'_1} & X \\ \downarrow & & & & \downarrow & & \downarrow & & & & \downarrow f_X = \text{id}_X \\ B_u & \longrightarrow & \dots & \longrightarrow & B' & \xrightarrow{f} & B'' & \longrightarrow & \dots & \longrightarrow & X \end{array}$$

where the left rectangle is the previous diagram, the middle square commutes, and the right rectangle defines $E(\tilde{w})$ (i.e, the bottom horizontal arrows are inclusions, the vertical arrows are the canonical identifications, and the maps I'_β ($\beta \in \{1, \dots, t\}$) are the ones from the definition of $E(\tilde{w})$). In particular, the entire diagram commutes up to images.

Next, we note that if we follow the arrows from \tilde{B}_{i_u} along the top of the diagram, and down the right side, then the image of the corresponding composition is exactly $E(\phi_f(w))$, by definition. The image of \tilde{B}_{i_u} as we trace the left side and bottom of the diagram is $(f \circ f_{B_u})(\tilde{B}_{i_u}) = f(B_u) = f(E(w))$. This proves the Proposition in the case that $w \in \mathcal{L}_{red}$.

Now we assume only that $w \in \mathcal{L}$ and \hat{w} is a prefix of w . We let w_{red} denote the (unique) reduced element in the equivalence class of w modulo \rightarrow . We note that, as we rewrite w , all of the reductions are made to a suffix that does not include any part of the prefix \hat{w} , since \hat{w} contains no symbols of the form $[f]$. It follows, in particular, that $w = \hat{w}w'$ and $w_{red} = \hat{w}w''$, and that w'' is the reduced form of w' . Applying ϕ_f to w and w_{red} , we get

$$\phi_f(w) = \tilde{w}[f]w' \quad \phi_f(w_{red}) = \tilde{w}[f]w''.$$

It follows that

$$\phi_f(w) = \tilde{w}[f]w' \rightarrow \tilde{w}[f]w'' = \phi_f(w_{red}).$$

Using the fact that E is constant on equivalence classes modulo \rightarrow , and the fact that $E(\phi_f(w)) = f(E(w))$ for reduced words w , we see that

$$E(\phi_f(w)) = E(\phi_f(w_{red})) = f(E(w_{red})) = f(E(w)).$$

□

5. WITNESS AUTOMATA

Definition 5.1. Let $B_1, B_2 \subseteq X$ be metric balls. We now define a PDA, called the (B_1, B_2) -witness automaton. There are four states: L (the initial state, or *loading* state), R (the *ready* state), C (the *cleaning* state), and E (the *eject* state). The directed edges are as follows:

- (1) Two types of directed edges lead away from L . The first type is a loop at L having the label $(\epsilon, \epsilon, A_{i,n})$ (i and n range over all possibilities: $i \in \{1, \dots, k\}$ and $n \in \{1, \dots, \ell\}$, where k and ℓ are as in Convention 2.21.) There is just one edge of the second type: it leads to the ready state R . Its label is $(\epsilon, \epsilon, addr(B_1))$.
- (2) Let $s \in S$, and let B be a region for s . By definition, $s|_B = f$, for some $f \in \text{Sim}_X(B, f(B))$. We create a directed edge from R to C with the label

$$(s, addr(B), addr(f(B))[f]);$$

there is one such edge for each $s \in S$ and region B for s .

- (3) The cleaning state C is the initial vertex for three kinds of edges. First, we note that there is obviously a uniform bound K on the lengths of the words in $\{addr(B)\}$, where the $addr(B)$ are the middle coordinates of the labels of edges leading from the ready state R . For each unreduced word $w \in \Sigma^*$ that occurs as a prefix of length less than or equal to $K + 1$ to a word in \mathcal{L} , we add a directed loop at C with label $(\epsilon, w, r(w))$. These are the first type of edges. There are two additional edges: the first is labelled $(\epsilon, \epsilon, \epsilon)$, and leads from C back to R . The second is labelled $(\epsilon, addr(B_2), \epsilon)$, and leads from C to E .
- (4) The edges leading away from state E are all of the same type. They are loops with label (ϵ, A, ϵ) , where A is an arbitrary symbol from the alphabet Σ , including the initial stack symbol, $\#$.

Remark 5.2. With a bit more care, it is possible to specify edges leading away from the loading state L in such a way that it is impossible to arrive in the state R with anything other than a valid word of \mathcal{L}_{red} written on the stack; we will assume that this extra care has been taken, leaving details to the reader.

Proposition 5.3. *For any pair of balls $B_1, B_2 \subseteq X$, the language*

$$\mathcal{L}_{(B_1, B_2)} = \{w \in S^* \mid w(B_1) \cap B_2 \neq \emptyset\}$$

is accepted by the (B_1, B_2) -witness automaton. In particular, $\mathcal{L}_{(B_1, B_2)}$ is (non-deterministic) context-free.

Proof. Let $w \in \mathcal{L}_{(B_1, B_2)}$. We will prove that w is accepted by the (B_1, B_2) -witness automaton.

We regard $w = s_1 \dots s_n$ as an element of G . By continuity of w , there is some ball $B_3 \subseteq B_1$ such that $w(B_3) \subseteq B_2$. We may furthermore assume (as in Sublemma 3.7(1)) that $s_j \dots s_n(B_3)$ lies inside a region for s_{j-1} , for $j = 2, \dots, n + 1$.

Our automaton begins in state L , with $\#$ written on its stack. It begins by writing the address of B_3 on its stack, and (in the process) moving to state R . Since B_3 lies inside a region D_{s_n} for s_n by our assumptions, it follows that the address for D_{s_n} is a prefix of the address for B_3 . Let $f \in \text{Sim}_X(D_{s_n}, f(D_{s_n}))$ satisfy $s_n|_{D_{s_n}} = f$. It follows that we are permitted to follow the directed edge labelled $(s_n, \text{addr}(D_{s_n}), \text{addr}(f(D_{s_n})))[f])$ (and in fact can follow no other) to state C . We note that, after doing so, the stack value of the path is $\phi_f(\text{addr}(B_3))$. It follows, in particular, that

$$E(\phi_f(\text{addr}(B_3))) = f(E(\text{addr}(B_3))) = s_n(E(\text{addr}(B_3))) = s_n(B_3),$$

where the first equality is due to Proposition 4.13, the second is due to the equality $f|_{B_3} = s_n|_{B_3}$, and the third is by the definitions of E and addr . It follows that the stack value is a word in the language \mathcal{L} whose reduced form in \mathcal{L}_{red} is the address of $s_n(B_3)$.

Next, beginning at state C , we repeatedly apply all possible reductions to prefixes of length $K + 1$, using the directed loops at C . The effect of doing this is to gather all letters of the form $[f]$ at the end of the prefix (in the $(K + 1)$ st position at worst; the symbols $[f]$ drop out entirely if the empty stack symbol becomes visible to the automaton). After doing this, we follow the directed edge labelled $(\epsilon, \epsilon, \epsilon)$ back to the ready state R . Note that the stack is “clean” – there are no symbols of the form $[f]$ among the first K symbols on the stack, and, in view of the fact that $s_n(B_3)$ lies inside a region for s_{n-1} , we can (as above) follow a unique directed edge back to the state C .

The process repeats. Eventually the automaton winds up in state C with a word $w \in \mathcal{L}$ on the stack satisfying

$$E(w) = s_1 \dots s_n(B_3),$$

and nothing left on the input tape. We again apply the cleaning procedure as described above, resulting in a word w' which still evaluates to $s_1 \dots s_n(B_3)$, but now has a prefix of length K that is free of the symbols $[f]$. (If the word w' has total length less than K , then w' is entirely free of the symbols $[f]$.) In view of the fact, that $(s_1 \dots s_n(B_3)) \subseteq B_2$ by our assumption, it now follows that the address of B_2 is a prefix of w' . We may therefore follow the arrow labelled $(\epsilon, \text{addr}(B_2), \epsilon)$ to the eject state E , where the automaton can completely unload its stack using the directed loops at E . Since the entire input tape has been read and the stack is now empty, the automaton accepts w .

Now let us suppose that $w = s_1 \dots s_n \notin \mathcal{L}_{(B_1, B_2)}$. We must show that the automaton cannot accept w . The automaton is forced to begin by loading the address of an (unknown) subball B_3 of B_1 on its stack. After doing this, it is in the ready state R . Assuming that the automaton has at least (and, therefore, exactly) one edge to follow from R , it arrives in state C with a word w' written on its stack, such that $E(w')$ is $s_n(B_3)$. We can then assume that the automaton follows the cleaning procedure sketched above. (Not doing so would only make the automaton less likely to accept w .) At this point, the automaton can move back to the ready state, or (if applicable) to the eject state. However, assuming that $n > 1$, moving to the eject state will cause the automaton to fail, since, from E , there is no longer any opportunity to read the input tape. If $n = 1$ (i.e., if $w = s_n$), then $s_n(B_3) \not\subseteq B_2$, so that the address of B_2 is not a prefix of the address for $s_n(B_3)$, and therefore the directed edge from C to E cannot be crossed.

We may therefore assume that the automaton moves back and forth between the ready and cleaning states, ultimately ending in the cleaning state C with a word w' on the stack, satisfying

$$E(w') = s_1 \dots s_n(B_3),$$

and no letters on the input tape. We may assume, moreover, that w' has no symbol of the form $[f]$ among its first K entries. Now, since $s_1 \dots s_n(B_3) \not\subseteq B_2$, the address for B_2 is not a prefix of the address for w' , it is not possible to follow the directed edge into E . The automaton's only move is to follow the arrow labelled $(\epsilon, \epsilon, \epsilon)$ back to R , where it gets stuck. It follows that the automaton cannot accept w . \square

6. PROOF OF THE MAIN THEOREM

Proof of Main Theorem. By Proposition 3.6, there is a finite test partition \mathcal{P} for G . We let $\mathcal{P} = \{B_1, \dots, B_\alpha\}$, where each of the B_i is a metric ball.

Consider the language

$$\hat{\mathcal{L}} = \{w \in S^* \mid w(B_i) \cap B_j \neq \emptyset, i, j \in \{1, \dots, \alpha\}, i \neq j\} = \bigcup_{i \neq j} \mathcal{L}_{(B_i, B_j)}.$$

By Proposition 5.3, and because a finite union of context-free languages is context-free, $\hat{\mathcal{L}}$ is context-free.

For any language \mathcal{L} , we let \mathcal{L}° denote the cyclic shift of \mathcal{L} . That is,

$$\mathcal{L}^\circ = \{w_2 w_1 \in S^* \mid w_1 w_2 \in \mathcal{L}; w_1, w_2 \in S^*\}.$$

A theorem of [5] says that the cyclic shift of a context-free language is context-free. It follows that $\hat{\mathcal{L}}^\circ$ is context-free.

Finally, we claim that $\text{CoWP}_S(G) = \hat{\mathcal{L}}^\circ$. The reverse direction follows from the (obvious) fact that $\hat{\mathcal{L}} \subseteq \text{CoWP}_S(G)$, and from the fact that the co-word problem is closed under the cyclic shift. Now suppose that $w = s_1 \dots s_n \in \text{CoWP}_S(G)$. Since \mathcal{P} is a test partition, we must have a ball $B_i \in \mathcal{P}$ such that $s_\beta \dots s_n s_1 \dots s_{\beta-1}(B_i) \neq B_i$. It follows easily that either $s_\beta \dots s_n s_1 \dots s_{\beta-1} \in \mathcal{L}_{(B_i, B_j)}$ or $s_\beta \dots s_n s_1 \dots s_{\beta-1} \in \mathcal{L}_{(B_j, B_i)}$, for some $i \neq j$. This implies that $w = s_1 \dots s_n \in \hat{\mathcal{L}}^\circ$. \square

REFERENCES

- [1] A. V. Anisimov. The group languages. *Kibernetika (Kiev)*, (4):18–24, 1971.
- [2] Collin Bleak, Francesco Matucci, and Max Neunhöffer. Embeddings into thompson's group v and $coCF$ groups. *arXiv:1312.1855*, 15 pages, 2013.
- [3] Collin Bleak and Olga Salazar-Díaz. Free products in R. Thompson's group V . *Trans. Amer. Math. Soc.*, 365(11):5967–5997, 2013.
- [4] Daniel Farley and Bruce Hughes. Finiteness properties of some groups of local similarities. *arXiv:1206.2692*, 46 pages, 2012.
- [5] Derek F. Holt, Sarah Rees, Claas E. Röver, and Richard M. Thomas. Groups with context-free co-word problem. *J. London Math. Soc. (2)*, 71(3):643–657, 2005.
- [6] Bruce Hughes. Local similarities and the Haagerup property. *Groups Geom. Dyn.*, 3(2):299–315, 2009. With an appendix by Daniel S. Farley.
- [7] J. Lehnert and P. Schweitzer. The co-word problem for the Higman-Thompson group is context-free. *Bull. Lond. Math. Soc.*, 39(2):235–241, 2007.
- [8] David E. Muller and Paul E. Schupp. Groups, the theory of ends, and context-free languages. *J. Comput. System Sci.*, 26(3):295–310, 1983.
- [9] Volodymyr V. Nekrashevych. Cuntz-Pimsner algebras of group actions. *J. Operator Theory*, 52(2):223–249, 2004.
- [10] M. H. A. Newman. On theories with a combinatorial definition of “equivalence.”. *Ann. of Math. (2)*, 43:223–243, 1942.

- [11] Claas E. Röver. Constructing finitely presented simple groups that contain Grigorchuk groups. *J. Algebra*, 220(1):284–313, 1999.
- [12] Michael Shapiro. A note on context-sensitive languages and word problems. *Internat. J. Algebra Comput.*, 4(4):493–497, 1994.

DEPARTMENT OF MATHEMATICS, MIAMI UNIVERSITY, OXFORD, OH 45056,
E-mail address: `farleyds@muohio.edu`