

DETERMINING SOLUBILITY FOR FINITELY GENERATED GROUPS OF PL HOMEOMORPHISMS

COLLIN BLEAK, TARA BROUGH, AND SUSAN HERMILLER

ABSTRACT. The set of finitely generated subgroups of the group $PL_+(I)$ of orientation-preserving piecewise-linear homeomorphisms of the unit interval includes many important groups, most notably R. Thompson's group F . Here, we show that every finitely generated subgroup $G < PL_+(I)$ is either soluble, or contains an embedded copy of the finitely generated, non-soluble Brin-Navas group B , affirming a conjecture of the first author from 2009. In the case that G is soluble, we show the derived length of G is bounded above by the number of breakpoints of any finite set of generators. We specify a set of 'computable' subgroups of $PL_+(I)$ (which includes R. Thompson's group F) and give an algorithm which determines whether or not a given finite subset X of such a computable group generates a soluble group. When the group is soluble, the algorithm also determines the derived length of $\langle X \rangle$. Finally, we give a solution of the membership problem for a particular family of finitely generated soluble subgroups of any computable subgroup of $PL_+(I)$.

1. INTRODUCTION

By studying subgroups of the group $PL_+(I)$ of piecewise-linear orientation-preserving homeomorphisms of $I = [0, 1]$ (with finitely many breaks in slope) through the dynamical properties of the action of the subgroup on the unit interval, researchers have obtained considerable information on groups that embed as subgroups of $PL_+(I)$. Brin and Squier [8] show that the group $PL_+(I)$ has no embedded non-abelian free groups. Guba and Sapir [12] show that any non-abelian subgroup of $PL_+(I)$ contained in R. Thompson's group F (which consists of the homeomorphisms for which all breakpoints are 2-adic rational numbers and all slopes are powers of 2) contains an embedded copy of $\mathbb{Z} \wr \mathbb{Z}$. In [6, 4, 3, 5] a theory is built connecting the solubility class of a subgroup $G < PL_+(I)$ with data on how the supports of the elements of G overlap with each other, and how these supports relate to the support of the whole action of G on I . One result in that theory is that the non-soluble non-finitely generated group $W = \bigoplus_{n \in \mathbb{N}} W_n$, where $W_n = (\dots((\mathbb{Z} \wr \mathbb{Z}) \wr \mathbb{Z}) \wr \dots \wr \mathbb{Z}) \wr \mathbb{Z}$ is the iterated wreath product of n copies

2010 *Mathematics Subject Classification.* 20F10; 20F16, 37C25

Keywords: Piecewise linear homeomorphism, Thompson's group, soluble, membership problem.

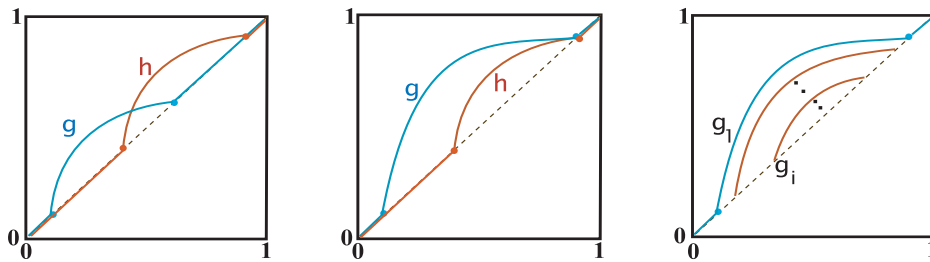


FIGURE 1. Transition chain, one-sided overlap, and tower of infinite height

of \mathbb{Z} , satisfies the properties that W contains an embedded copy of every soluble subgroup of $PL_+(I)$, and any non-soluble subgroup of $PL_+(I)$ contains an embedded copy of W (Corollary 1.2 and Theorem 1.1, respectively, of [5]). In the finitely generated case, it has been conjectured [5] that a stronger result holds, namely that any finitely generated non-soluble subgroup of $PL_+(I)$ contains an embedded copy of the Brin-Navas group B , the finitely generated non-soluble group introduced by Brin [7, Section 5] (as G_1) and Navas [13, Example 6.3] (as Γ) that is presented by

$$B = \langle \{w_i \mid i \in \mathbb{Z}\}, s \mid w_i^s = w_{i+1}, [w_i^{w_k^m}, w_j] = 1 \ (i < k, j < k, m \in \mathbb{Z} \setminus \{0\}) \rangle,$$

and generated by s and w_0 . In this paper we verify this conjecture.

To be more specific, we say that G admits a transition chain if there are two elements $g, h \in G$ with components of support $(a, b), (c, d)$, respectively, such that $a < c < b < d$. If G does not admit such a chain, then we say that G is chainless. In a similar fashion, the group G admits a one-sided overlap if there are two elements $g, h \in G$ with components of support $(a, b), (a, c)$ or $(a, b), (c, b)$, respectively, such that $a < c < b$. Finally, G admits a tower of infinite height if there is an infinite sequence $\{g_i\}_{i \in \mathbb{N}}$ of elements of G with components of support A_i , respectively, such that $A_{i+1} \subsetneq A_i$ for all i . We illustrate these three properties in the special case of elements with a single component of support in Figure 1.

In [4, Theorem 1.1 and Lemma 1.4] (restated below in Theorem 2.3 and Lemma 2.4(2)) the first author shows that a soluble subgroup of $PL_+(I)$ must be chainless and does not admit a tower of infinite height. In our main theorem (Theorem 3.1) in Section 3 below, we show that in the finitely generated case the converse of each of these also holds. In Lemma 2.2 we show that for a subgroup $G \leq PL_+(I)$ generated by a finite set $X \subset PL_+(I)$, the number of G -orbits of the set of breakpoints of elements of G is bounded above by the cardinality of the set of breakpoints of the elements of X (where a point $x \in (0, 1)$ is a breakpoint of $g \in PL_+(I)$ if g changes slope at x); this is applied in Theorem 3.1 to obtain a bound on the derived length in the soluble case.

Theorem 3.1. *Let $G < PL_+(I)$ be generated by a finite set X . The following are equivalent.*

- (1) G is not soluble.
- (2) G admits a transition chain.
- (3) G admits a one-sided overlap.
- (4) G admits a tower of infinite height.

Moreover, if G is soluble, the derived length of G is less than or equal to the cardinality of the set \mathcal{B}_X of breakpoints of elements of X .

Our first application of Theorem 3.1 is a verification of the conjecture discussed in the first paragraph of this introduction. Note that since a soluble group cannot contain a non-soluble subgroup, a subgroup $G \leq PL_+(I)$ containing a copy of the Brin-Navas group B is also non-soluble. Theorem 1.4 of [5] states that if G admits a transition chain, then G admits an embedded copy of B , and so we obtain the following corollary resolving the conjecture.

Corollary 1.1. *Let G be a finitely generated subgroup of $PL_+(I)$. Then G is non-soluble if and only if the Brin-Navas group embeds in G .*

Thus B not only contains every soluble subgroup of $PL_+(I)$, but B is also contained in every finitely generated non-soluble subgroup of $PL_+(I)$.

In Section 4 we give further applications of Theorem 3.1 to several algorithmic questions for subgroups of $PL_+(I)$. We restrict our consideration to *computable* subgroups $G < PL_+(I)$, in which several basic operations can be implemented (see p. 12 for the full definition). Examples of requirements for a group $C \leq PL_+(I)$ to be computable are that the breakpoints and endpoints of components of support of elements can be computed and compared, and that given a finite collection of slopes of affine components of graphs of functions in C , a computer can determine whether the multiplicative subgroup of \mathbb{R}_+^* generated by these slopes is discrete (i.e., has a lower bound on the distance from the identity 1 for all non-identity elements). In particular, we note that the word problem is solvable in finitely generated computable subgroups of $PL_+(I)$.

For a finitely generated group G with solvable word problem, it is natural to ask whether G has solvable *membership decision problem (MDP)* for a fixed subgroup H of G , or solvable *uniform subgroup membership problem (USMP)*. Given a group G with a finite generating set Y and a subgroup H of G , the MDP for the subgroup H asks whether there is an algorithm that, upon input of any word w over $Y^{\pm 1}$, can determine whether or not w lies in the subgroup H . The USMP for G asks whether there is an algorithm that, upon input of any word w over $Y^{\pm 1}$ together with a finite list X of words over $Y^{\pm 1}$, can determine whether or not w lies in the subgroup $\langle X \rangle$.

The computable subgroups of $PL_+(I)$ include R. Thompson's group F . Many algorithmic properties are known for F , including linear time solutions

of the word problem [9, 12, 14] and conjugacy problem [12, 2]. Golan [10] has shown that the generation problem (determining whether a finite input set X of elements of F generates all of F) is solvable for F , and Guba and Sapir [12] solve the membership decision problem for subgroups H of F that are the split group (defined in Section 2.3) of a finitely generated subgroup of F . However, solvability of the MDP for many subgroups, and solvability of the uniform subgroup membership problem, are open questions for R. Thompson's group F [11],[1, Question 2.11], and more generally for computable subgroups of $PL_+(I)$.

A potential approach to building an algorithm to solve the USMP for a computable subgroup G of $PL_+(I)$ is to have the first step of the algorithm determine whether the input set X generates a soluble subgroup; the algorithm then branches into separate subalgorithms in the soluble and non-soluble cases. That is, this first step requires a solution of the *soluble subgroup recognition problem (SSRP)*; given a group G with a finite generating set Y , the SSRP asks whether there is an algorithm that, upon input of a finite set X of words over $Y^{\pm 1}$, can determine whether or not the subgroup $\langle X \rangle$ of G generated by X is a soluble group.

Our next application of Theorem 3.1 is a solution of the SSRP for computable subgroups of $PL_+(I)$, including R. Thompson's group F . (We note that for all of the algorithms in this paper, we allow elements of G to be input using formats other than a list of words over a finite generating set of G ; for example, if all breakpoints and slopes of a homeomorphism are rational numbers, the element can be input as an integer list of numerators and denominators of the breakpoints and slopes.)

Theorem 4.4. *Let C be a computable subgroup of $PL_+(I)$. The soluble subgroup recognition problem is solvable for C ; that is, there is an algorithm which, upon input of a finite subset X of C , can determine whether or not the subgroup $\langle X \rangle$ generated by X is a soluble group. Moreover, in the case that the group $\langle X \rangle$ is soluble, the algorithm also determines its derived length.*

The proof of Theorem 4.4 uses the concept of controllers for chainless groups developed by the first author in [3]; we discuss background on this topic in Section 2.4. The key information that enables the algorithm in Theorem 4.4 to halt after finitely many steps in the non-soluble case is the bound on derived length computed from the finite list X , obtained in Theorem 3.1.

The algorithm of Theorem 4.4 also leads to a proof that the membership decision problem is solvable for some finitely generated soluble subgroups of computable subgroups of $PL_+(I)$. A set $X \subset PL_+(I)$ is a *set of one-bump functions with fundamental domains* if X satisfies the following properties:

- (Z0) Each element h of X admits exactly one component of support, which we denote by A_h . (That is, the graph of h has “one bump”.)

- (Z1) No pair of elements of X forms a transition chain or a one-sided overlap.
- (Z2) If $h, h' \in X$ and $h \neq h'$, then $A_h \neq A_{h'}$.
- (Z3) For each $h \in X$, there is an $r_h \in A_h$ such that for every $h' \in X$ with $A_{h'} \subsetneq A_h$, the containment $A_{h'} \subseteq (r_h, r_h \cdot h)$ also holds. (That is, $(r_h, r_h \cdot h)$ is a fundamental domain for the conjugation action by powers of h .)

In Lemma 4.3 we show that a group $H = \langle X \rangle$ generated by a finite set X of one-bump functions with fundamental domains is a soluble group, contained in the smallest class of groups that includes the trivial group and is closed under wreath products with \mathbb{Z} and finite direct sums.

Corollary 4.7. *Let C be a computable subgroup of $PL_+(I)$. Let H be a subgroup of C generated by a finite set of one-bump functions with fundamental domains. Then the membership decision problem is solvable for H ; that is, there is an algorithm which, upon input of an element w of C , can determine whether $w \in H$.*

In Corollary 4.6 we show that there is an algorithm to find a finite generating set consisting of one-bump functions with fundamental domains for any finitely generated split soluble subgroup of a computable group. Combining these algorithms shows that membership in such subgroups is uniformly solvable.

Corollary 4.8 *Let C be a computable subgroup of $PL_+(I)$. There is an algorithm which, upon input of an element w of C and a finite subset X of C generating a split soluble group, can determine whether $w \in \langle X \rangle$.*

Before proceeding to the proofs of these results in Sections 3 and 4, we begin in Section 2 with background and notation for the group $PL_+(I)$.

ACKNOWLEDGMENTS

The third author was partially supported by grants from the Simons Foundation (#245625) and the National Science Foundation (DMS-1313559).

The authors thank Gili Golan for helpful discussions.

2. THE GROUP $PL_+(I)$

Here we give the basic definitions we require for discussing the group $PL_+(I)$ and its elements. The results discussed in this section are first introduced in either [6], or later in [4, 3, 5].

2.1. Right actions, supports, slopes, and breakpoints. Throughout this paper we will use right action notation. In particular, if $x \in [0, 1]$ and $g \in PL_+(I)$, we write xg for the image of x under the map g . As is somewhat

traditional (but not universal) for right actions, for elements $g, h \in PL_+(I)$, and $S \subset [0, 1]$ we set

$$\begin{aligned} Sg &:= \{sg \mid s \in S\} \\ \text{Supp}(g) &:= \{x \in [0, 1] \mid xg \neq x\} \\ g^h &:= h^{-1}gh \\ [g, h] &:= g^{-1}h^{-1}gh \end{aligned}$$

for the *image of S under the action of g* , the *support of g* , the *conjugate of g by h* , and the *commutator of g and h* , respectively. With this notation in place we have a standard lemma from permutation group theory, restated for elements of the group $PL_+(I)$.

Lemma 2.1. *Let $g, h \in PL_+(I)$. Then $\text{Supp}(g^h) = \text{Supp}(g)h$.*

For a subgroup $G \leq PL_+(I)$, the associated *slope group of G* , denoted Π_G , is the multiplicative subgroup of the positive real numbers generated by the slopes of affine components of elements of G .

For $g \in PL_+(I)$ and $x \in (0, 1)$, we say that x is a *breakpoint* of g whenever xg' does not exist (here, we are using g' to denote the derivative of g). For a set $X \subset PL_+(I)$ we denote by \mathcal{B}_X the set of breakpoints of the elements in X . That is

$$\mathcal{B}_X := \{x \in (0, 1) \mid \exists g \in X, xg' \text{ does not exist}\}.$$

A key technical point used in the proof of Theorem 3.1 is the following lemma, well known to researchers in PL functions on the line.

Lemma 2.2. *If $G < PL_+(I)$ has a finite generating set X , then the number of orbits (under the action of G) of the set of all breakpoints of G is bounded above by the (finite) cardinality of the set of breakpoints of X .*

2.2. Orbitals, towers, and transition chains. We extend the definition of support to groups, so for a group $G \leq PL_+(I)$, we set

$$\text{Supp}(G) := \bigcup_{g \in G} \text{Supp}(g),$$

noting that if $x \in \text{Supp}(G)$ then there is some $g \in G$ such that $xg \neq x$. As $\text{Supp}(G)$ is an open set, it can be written as a disjoint union of open intervals, each one of which is called an *orbital of G* . That is, an orbital of G is a connected component of the support of the action of G on $[0, 1]$. Note that if $A \neq B$ are two orbitals of G , then there is no element of G which can move a point in A to a point in B , which partly motivates our language (as it means that each G -orbit is contained in some orbital of G). For an element $g \in PL_+(I)$, a subset $A = (a, b) \subset [0, 1]$ is an orbital of $\langle g \rangle$ if and only if A is a component of $\text{Supp}(g)$; in this case we say A is an *orbital of g* .

A *signed orbital* is a pair $((a, b), g)$ consisting of an open interval $(a, b) \subset [0, 1]$ and an element $g \in PL_+(I)$ such that (a, b) is an orbital of g ; here (a, b) is the orbital and g is the *signature*.

A *tower* is a set \mathcal{T} of signed orbitals satisfying the property that whenever $((a, b), g)$ and $((c, d), h)$ are in \mathcal{T} , then

- (1) $(a, b) \subseteq (c, d)$ or $(c, d) \subseteq (a, b)$, and
- (2) $(a, b) = (c, d)$ implies $g = h$.

For a tower \mathcal{T} , the cardinality $|\mathcal{T}|$ is called the *height of \mathcal{T}* . Given a group $G \leq PL_+(I)$ and a tower \mathcal{T} , we say that \mathcal{T} is *associated* with G , or that G *admits* the tower \mathcal{T} , if all the signatures of the signed orbitals in \mathcal{T} are elements of G . If $G \leq PL_+(I)$ and A is an open subinterval of the unit interval I , the *orbital depth* of A in G is the supremum of the heights of finite towers associated with G in which the smallest orbital has the form (A, g) for some $g \in G$. If $G \leq PL_+(I)$, we set the *depth of G* to be the supremum of the heights of the towers in the full set of towers associated with G .

The graph on the right in Figure 1 depicts a tower of infinite height. The ordering of indices, which appears inverted, favours the perspective of “depth” over “height.” Reasons for this will become apparent in our construction proving Theorem 4.4.

The main theorem of [4] is the following.

Theorem 2.3. [4, Theorem 1.1] *Let $G \leq PL_+(I)$ and $n \in \mathbb{N}$. The group G is soluble with derived length n if and only if the depth of G is n .*

For chainless groups, towers have further special properties. A tower \mathcal{T} is *exemplary* if whenever $(A, g), (B, h) \in \mathcal{T}$ with $A \subsetneq B = (a, b)$ then

- (1) the orbitals of g are disjoint from the ends of the orbital B , and
- (2) no orbital of g in B shares an end with B .

Another way to put this is that there is an $\epsilon > 0$ so that for any orbital C of g we have $C \cap B \neq \emptyset$ implies $C \subset (a + \epsilon, b - \epsilon)$.

We say that $G \leq PL_+(I)$ admits a *complex overlap* if G admits either a transition chain or a one-sided overlap. The following lemma connects the existence of a complex overlap for G , the depth of G , and the types of towers that can occur. Much of this lemma is a restatement of lemmas in [4, 5]; we add a proof that existence of a transition chain implies existence of a one-sided overlap.

Lemma 2.4. *Let G be a subgroup of $PL_+(I)$.*

- (1) *The group G admits a one-sided overlap if and only if G admits a transition chain.*
- (2) [4, Lemma 1.4] *If G admits a transition chain, then G admits an infinite tower.*

- (3) [5, Lemma 2.7(2)] *If G is chainless, then every tower associated with G is exemplary.*

Proof. It is shown in [5, Lemma 2.7(3)] that if a group $G \leq PL_+(I)$ admits a one-sided overlap, then G admits a transition chain.

Suppose that G admits a transition chain $\{((a, b), f), ((c, d), g)\}$, where $a < c < b < d$. If a is not in the support of g , then f^g admits an orbital (a, e) with $e \neq b$, and hence the pair of signed orbitals $\{((a, b), f), ((a, e), f^g)\}$ represents a one-sided overlap for G . Similarly, if d is not in the support of f , then there is an $e \neq d$ such that the pair $\{((c, d), g), ((e, d), g^f)\}$ represents a one-sided overlap for G .

We extend this endpoint-support argument to the left and the right until we run out of orbitals of f or of g . Eventually we must fail to have an end of one of these signed orbitals in the support of an orbital of the other element, and therefore we can find a one-sided overlap where the signatures are either the elements f and f^g or the elements g and g^f . □

In particular, subgroups of $PL_+(I)$ that admit a complex overlap have infinite depth, and so by Theorem 2.3 they are not soluble.

2.3. The split group and one-bump factors. Let $G \leq PL_+(I)$. Given an element $f \in G$, let A_1, \dots, A_k be the orbitals of f . For all $1 \leq i \leq k$, let f_i be the element of $PL_+(I)$ defined by $f_i|_{A_i} = f|_{A_i}$, and $x f_i = x$ for all $x \in I \setminus A_i$. Each function f_i has precisely one component of support, the functions f_i commute with each other, and $f = f_1 \cdots f_k$. We call these functions f_i the *one-bump factors* of f , and we refer to the signed orbitals (A_i, f_i) as the *factor signed orbitals* associated to f .

The *split group* $S(G)$ associated to the group $G \leq PL_+(I)$, introduced in [3], is the group generated by the one-bump factors of all of the elements of G . If $G = S(G)$, then we say that the group G is *split*. Note that G is a subgroup of $S(G)$, and whenever G is a subgroup of another group H , then $S(G) \leq S(H)$. It is immediate from the chain rule that the slope group of $S(G)$ is also the slope group of G ; that is, $\Pi_{S(G)} = \Pi_G$. In [10, Lemma 5.5], Golan shows that $S(S(G)) = S(G)$ for any subgroup G of Thompson's group F ; her proof also extends to any subgroup G of $PL_+(I)$.

We record the following result of the first author for use in Section 4.

Theorem 2.5. [3, Cor. 4.6] *Suppose that G is a subgroup of $PL_+(I)$. The derived length of G equals the derived length of $S(G)$.*

2.4. Balance and controllers for chainless groups. Suppose $a < d \in [0, 1]$. If $g \in PL_+(I)$ has an orbital of the form (a, b) or (c, d) where $a < b \leq d$ and $a \leq c < d$ then we say g *realises an end of (a, d)* . If there are both b and c so that $a < b \leq c < d$ and g has orbitals (a, b) and (c, d) then we say

g realises both ends of (a, d) . Finally, if g has orbital (a, d) , then we say g realises (a, d) (and in this last case we also say that g realises both ends of (a, d)).

Theorem 2.6 (Brin’s Ubiquity Theorem [6]). *If a group $H \leq PL_+(I)$ contains an element that realises exactly one end of an orbital of H , then H contains a subgroup isomorphic to R. Thompson’s group F .*

Let H be a subgroup of $PL_+(I)$ with orbital A . If whenever $h \in H$ realises one end of A , then h also realises the other, we say A is *balanced for H* . The group H is *balanced* if for every $G \leq H$ and every orbital A of G , the orbital A is balanced for G .

The following lemma will be used in the proof of Theorem 4.4.

Lemma 2.7. *If $H \leq PL_+(I)$ is chainless then H is balanced.*

Proof. Suppose to the contrary that H is chainless but not balanced. Then there is a subgroup G of H , an orbital (a, b) of G , an element g of G , and an orbital (c, d) of g such that $a \leq c$, $d \leq b$, and exactly one of the equations $a = c$ or $b = d$ holds. Suppose that $a = c$ (the proof in the other case is similar). Then $d \in (a, b)$ so there is another element h of G with $d \in \text{Supp}(h)$; let B be the orbital of h containing d . Then the signed orbitals $((c, d), g)$ and (B, h) form a complex overlap, and so Lemma 2.4(1) shows that G , and hence H , is not chainless, giving the required contradiction. \square

Suppose that $H \leq PL_+(I)$ is a chainless group with a single orbital A . Since H is balanced, A cannot satisfy Brin’s Ubiquity condition. Following the discussion of [3, Section 3], a *controller of H over A* is an element $c \in H$ realising A and satisfying the property that $H = \langle c, \mathring{H}_A \rangle$, where \mathring{H}_A is the subgroup of H consisting of all elements $g \in H$ for which there is a neighbourhood in A of the ends of A upon which g acts as the identity. Lemma 2.7 together with [3, Lemma 3.12], show that H must have a (not necessarily unique) controller c over A . Moreover, given $h \in H$, there is a unique integer k and $\mathring{h} \in \mathring{H}_A$ so that $h = c^k \mathring{h}$. In the algorithm in the proof of Theorem 4.4, we construct elements that are controllers for groups H and orbitals A arising in Step 3a of that algorithm.

3. TRANSITION CHAINS IN FINITELY GENERATED NON-SOLUBLE SUBGROUPS

In this section we prove Theorem 3.1. Applying results established in Section 2.2, it remains to show that every finitely generated nonsoluble subgroup of $PL_+(I)$ admits a transition chain (that is, the implication (1) \Rightarrow (2)). The motivating idea for our proof is that if specific orbits of breakpoints of the generators intersect the topological support of deep enough derived subgroups, then the breakpoint orbits must interleave in such a way as to give rise to a transition chain. Thus, we can argue that when passing into

derived subgroups, after at most $|\mathcal{B}_X|$ such passages, we run out of orbits (and so the group is soluble as all remaining elements must be trivial).

Theorem 3.1. *Let $G < PL_+(I)$ be generated by a finite set X . The following are equivalent.*

- (1) G is not soluble.
- (2) G admits a transition chain.
- (3) G admits a one-sided overlap.
- (4) G admits a tower of infinite height.

Moreover, if G is soluble, the derived length of G is less than or equal to the cardinality of the set \mathcal{B}_X of breakpoints of elements of X .

Proof. Let $G = \langle X \rangle$ be a finitely generated subgroup of $PL_+(I)$. Lemma 2.4 shows that (3) \Leftrightarrow (2) \Rightarrow (4), and the implication (4) \Rightarrow (1) follows immediately from Theorem 2.3. It remains to show (1) \Rightarrow (2) and the claim on derived length.

Suppose that G is chainless. Let $n \in \mathbb{N}$ so that $n - 1$ is equal to the number of G -orbits of the set of breakpoints of G (a finite number by Lemma 2.2). Suppose that G is either non-soluble or of derived length z for some $z \geq n$.

We note in passing that we may assume $n > 1$, since if $n = 1$ then G has no breakpoints, and so $G = \{1\}$ and hence the derived length of G is 0, which does not exceed the number 0 of G -orbits in the set \mathcal{B}_G of breakpoints of elements of G .

Theorem 2.3 implies that G admits a tower

$$\mathcal{T} = \{(A_1, g_1), (A_2, g_2), \dots, (A_n, g_n)\}$$

of height n , which by Lemma 2.4(3) is exemplary, and hence we may assume that the signed orbitals of \mathcal{T} are indexed in such a fashion that for all indices $i < n$ we have $\overline{A}_{i+1} \subset A_i$. (In the case that G is non-soluble, G admits towers of arbitrary height by Theorem 2.3.)

The endpoints of A_i might not be breakpoints of g_i ; that is, g_i may have a disjoint orbital with the same endpoint and the same slope for g_i in a neighbourhood of that endpoint. To take this into account, we widen the interval that we consider, as follows. There is a maximal $k_i \in \mathbb{N}$ such that there is an ordered tuple X_i of signed orbitals

$$X_i = ((A_{i1}, g_i), (A_{i2}, g_i), \dots, (A_{ik_i}, g_i)),$$

where we write $A_{ij} = (a_{ij}, b_{ij})$, satisfying the properties that for each index $j < k_i$ we have $b_{ij} = a_{i(j+1)}$ and there is an index m_i with $A_i = A_{im_i}$.

Since G is chainless, Lemma 2.4(1) shows that G does not admit complex overlaps. From the fact that each A_{ij} shares an end with each of its ‘neighbours’ in X_i , and $\overline{A}_{im_i} = \overline{A}_i \subset A_{i-1}$ for $i > 1$, we deduce that for $1 < i < n$ and $1 \leq j \leq k_i$ we have $\overline{A}_{ij} \subset A_{i-1}$ since otherwise G would admit a complex overlap.

Now, for each index $1 \leq i < n$, let $c_i = a_{(i+1)1}$ and $d_i = b_{(i+1)k_{i+1}}$, and let $a_i = a_{im_i}$ and $b_i = b_{im_i}$. Furthermore, set c_n to be some breakpoint of g_n in A_n (such must exist since g_n cannot be affine over A_n). We do not define d_n . If $1 \leq i < n$ it is now the case (by the maximality of k_{i+1}) that c_i and d_i are breakpoints of the element g_{i+1} , and that $\overline{(c_i, d_i)} \subset A_i$.

As n is larger than the number of orbits of breakpoints of G under the action of G , there are indices $r < s$ so that c_r and c_s are in the same G -orbit. Hence there is an element $g \in G$ such that $c_r \cdot g = c_s \in A_s$. In particular, by Lemma 2.1 and the nonexistence of complex overlaps we see that for each index j the interval $A_{(r+1)j} \cdot g$ is an orbital of g_{r+1}^g with closure properly contained in A_s away from the ends of A_s . In particular, we have $\overline{(c_r, d_r)} \cdot g \subset A_s$. The above implies the following chain of relationships.

$$c_r \leq a_{r+1} \leq a_s < c_r \cdot g = c_s < d_r \cdot g < b_s \leq b_{r+1} \leq d_r.$$

This means that g moves c_r to the right across a_s , while also moving d_r to the left across b_s . However, any given orbital of g has all of its points moved in the same direction by g , so g must have at least two distinct orbitals, one orbital (x, y) containing a_s and another containing b_s . Consequently, we have that $x < a_s < y < b_s$ and so $\{((x, y), g), ((a_s, b_s), g_s)\}$ is a transition chain for G . Since G is chainless, this gives a contradiction, so we can conclude that G is indeed soluble with derived length z less than or equal to the number of G -orbits of the set of breakpoints of G . Lemma 2.2 completes the proof. \square

Note that the hypothesis that G is finitely generated is not required for the equivalence of (2) and (3), and that these two conditions could be replaced by the single condition: “ G admits a complex overlap”.

4. AN ALGORITHM TO DETECT SOLUBILITY

The goal of this section is to use Theorem 3.1 and the concept of controllers from Section 2.4 to construct the algorithm to solve the soluble subgroup recognition problem for the proof of Theorem 4.4.

Let $C \leq PL_+(I)$. In order to input a finite list of elements of C into our procedure, we need to be able to write these elements with some sort of data structure; for example, if C is R. Thompson’s group F , we may input an element as a list of numerators and denominators of the breakpoints and slopes of the homeomorphism, since all of these are rational numbers, but we may instead input the element as a word over a finite generating set for F , as a tree pair diagram, or as any other construct that encodes this information. For whatever structure is used, there are several pieces of information we need to be able to calculate from this data, which we list in the following processes. Some of these processes are required to hold for the (potentially larger) split group $S(C)$ (see Section 2.3 for this construction).

Processes:

- (1) Given $g, h \in S(C)$ determine gh and g^{-1} .
- (2) Given $g \in C$, determine its set of breakpoints \mathcal{B}_g .
- (3) Given $g \in S(C)$ and a breakpoint or orbital endpoint x of $S(C)$, compute $x \cdot g$.
- (4) Given two points $a, b \in [0, 1]$ that occur either as breakpoints or as orbital endpoints of elements of $S(C)$, determine whether $a < b$, $a = b$, or $a > b$.
- (5) Given $g \in S(C)$, produce the finite tuple $X_g = [A_1, A_2, \dots, A_{k_g}]$ of all orbitals of g , where each A_i is stored as the ordered pair $(a_i, b_i) = (\inf(A_i), \sup(A_i))$, and $a_1 < a_2 < \dots < a_{k_g}$.
- (6) Given a signed orbital (A, g) associated with $S(C)$, output the factor signed orbital (A, h) satisfying $h|_A = g|_A$ and $\text{Supp}(h) = A$.
- (7) Given a signed orbital $((a, b), g)$ of $S(C)$, determine the slopes $m_{ga} := ag'_+$ and $m_{gb} := bg'_-$ of the affine components of the graph of g over (a, b) near a and b respectively.
- (8) Given two elements m_1, m_2 of the slope group $\Pi_C = \Pi_{S(C)}$ of C , compute $m_1 m_2$ and m_1^{-1} , and determine whether or not $m_1 < m_2$, $m_1 = m_2$, or $m_1 > m_2$.
- (9) Given a finite set $Z = \{m_1, m_2, \dots, m_k\}$ of positive numbers in $\Pi_C = \Pi_{S(C)}$, determine if the multiplicative group $\Pi_Z := \langle m_1, m_2, \dots, m_k \rangle \leq \mathbb{R}_+^*$ is discrete. If this group Π_Z is discrete, further determine integers p_1, p_2, \dots, p_k so that $\Pi_{Z,s} := m_1^{p_1} m_2^{p_2} \dots m_k^{p_k}$ is the least value in Π_Z greater than one.

We say that a subgroup $C \leq PL_+(I)$ is a *computable group* if the elements of C have representatives for which this list of processes can be carried out by a computer.

For a subgroup C of $PL_+(I)$, if the sets of breakpoints, orbital endpoints and slopes of affine components of graphs of elements of C are sufficiently specialised sets of values, then these processes can be performed. We observe that all of the processes above can be carried out for elements in R. Thompson's group F by a modern computer. In particular, since breakpoints of elements of F all lie in the rationals, orbital endpoints for elements of F are also rational, and so the breakpoints and orbital endpoints of elements of $S(F)$ are rational as well; hence the processes involving $S(F)$ can be performed. For the most complex process, namely Process 9, one uses a generalised Euclidean Algorithm on the log base two values of the sets of slopes to determine the integers p_i in this case. Hence F is computable.

We also note that any subgroup of a computable group is computable. For the algorithm we provide below, we actually work in subgroups of the split group $S(G)$ of our original computable group G . As a consequence the following corollary and lemma will be applied several times. Corollary 4.1 follows immediately from Theorems 2.5 and 3.1, and the fact that whenever

H_1 is a subgroup of a group H_2 , the derived length of H_1 is at most the derived length of H_2 .

Corollary 4.1. *Let G be a subgroup of $PL_+(I)$, and let H be a subgroup of the split group $S(G)$ containing G .*

- (1) *The derived length of G equals the derived length of H .*
- (2) *If H admits a complex overlap, then G also admits a complex overlap and G is not soluble.*

We will also apply the following lemma in the proof of Theorem 4.4.

Lemma 4.2. *Let $G \leq PL_+(I)$. Suppose that $(A_1, g_1), \dots, (A_q, g_q)$ are factor signed orbitals associated to elements of G .*

- (1) *If $A_1 = \dots = A_q$ and if p_1, \dots, p_q are any integers, then the one-bump factors of $g_1^{p_1} \dots g_q^{p_q}$ are also one-bump factors of an element of G .*
- (2) *If $\overline{A_1} \subset A_2$, then the conjugate $g_1^{g_2}$ is also a one-bump factor of an element of G .*

Proof. Let $\hat{g}_1, \dots, \hat{g}_q$ be elements of G such that (A_i, g_i) is an associated factor signed orbital of \hat{g}_i for each i .

First suppose that $A_1 = \dots = A_q$ and $p_1, \dots, p_q \in \mathbb{Z}$, and let $c := g_1^{p_1} \dots g_q^{p_q}$. Let $\hat{c} := \hat{g}_1^{p_1} \dots \hat{g}_q^{p_q}$. Then since $g_i|_{A_1} = \hat{g}_i|_{A_1}$ for all i , and each $g_i|_{A_i}$ is a homeomorphism of the interval A_i , we have $c|_{A_1} = \hat{c}|_{A_1}$. Hence the one-bump factors of c are exactly the one-bump factors of \hat{c} whose support is contained in the interval A_1 .

Next suppose that $\overline{A_1} \subset A_2$. By Lemma 2.1, the support of the conjugate $g_1^{g_2}$ is the interval $A_1 \cdot g_2$. Since g_2 acts as a homeomorphism of the interval A_2 and fixes the rest of $I \setminus A_2$, then $A_1 \cdot g_2 \subseteq A_2$. Similarly the conjugation action of \hat{g}_2 on \hat{g}_1 takes the signed orbital (A_1, \hat{g}_1) to the signed orbital $(A_1 \cdot \hat{g}_2, \hat{g}_1^{\hat{g}_2})$. Since $\hat{g}_2|_{A_2} = g_2|_{A_2}$, then $A_1 \cdot \hat{g}_2 = A_1 \cdot g_2$ and on this interval the functions $\hat{g}_1^{\hat{g}_2}$ and $g_1^{g_2}$ agree. Thus $g_1^{g_2}$ is a one-bump factor of $\hat{g}_1^{\hat{g}_2}$. \square

While Corollary 4.1 and Lemma 4.2 are used toward determining when the input group G is not soluble, the following lemma will be used toward determining when G is soluble.

Lemma 4.3. *Suppose that $H < PL_+(I)$ is generated by a finite set Z of one-bump functions with fundamental domains, and let S_Z be the set of signed orbitals associated to the elements of Z . Then H is a soluble group, and the derived length of H is the largest height of a tower of signed orbitals contained in the set S_Z .*

Proof. Let n be the largest height of a tower of signed orbitals that are contained in S_Z ; we proceed by induction on n .

If $n=0$, then S_Z and hence Z is empty, and H is the trivial group, which is soluble of derived length 0. If $n = 1$, then Properties Z1-Z2 of the definition

of a set of one-bump functions with fundamental domains (p. 4) imply that the supports of the elements of the generating set Z are pairwise disjoint, and so the elements of Z commute. Therefore H is abelian, and so H is soluble with derived length 1.

Now suppose that $n > 1$ and the result is true for finite sets satisfying Properties Z0-Z3 with maximum associated tower height at most $n - 1$. For each element $h \in Z$, let A_h denote the support $\text{Supp}(h)$ (in the notation of Property Z0) and let $\text{orbDepth}(A_h)$ denote the maximum height of a tower built from elements of S_Z such that A_h is the smallest orbital (that is, A_h is contained in the supports of all of the other signed orbitals in the tower). Let $Y := \{h \in Z \mid \text{orbDepth}(A_h) = 1\}$, and for each $h \in Y$, let $P_h := \{h' \in Z \mid A_{h'} \subsetneq A_h\}$. Property Z2 implies the set Y does not contain two elements with the same support. Then Property Z1 implies that the elements of Y have disjoint support, and so H is the direct product of the subgroups $\langle h, P_h \rangle$ for $h \in Y$.

Note that each subset P_h of Z satisfies Properties Z0-Z3, and its associated signed orbitals have maximal tower height $n - 1$, so by induction the group $\langle P_h \rangle$ is a soluble group for each $h \in Y$, and the derived length of $\langle P_h \rangle$ is the maximal height of a tower that can be built from signed orbitals associated to elements of P_h .

Now Property Z3 implies that for distinct integers j the groups $\langle P_h \rangle^{h^j}$ have disjoint support, and so the subgroup of $\langle h, P_h \rangle$ generated by these subgroups is the direct product $\bigoplus_{j \in \mathbb{Z}} \langle P_h \rangle^{h^j}$. The conjugation action of the group $\mathbb{Z} = \langle h \rangle$ in this direct product permutes the summands. Hence the group $\langle h, P_h \rangle$ is a wreath product $\langle h, P_h \rangle = \langle P_h \rangle \wr \mathbb{Z}$. Then $\langle h, P_h \rangle$ is again soluble. Moreover, the derived length of $\langle h, P_h \rangle$ is one more than the derived length of $\langle P_h \rangle$; that is, it is the maximum height of a tower associated to elements of the set $\{h\} \cup P_h$. Since at least one $\langle P_h \rangle$ has derived length $n - 1$, this implies that some $\langle P_h, h \rangle$ has derived length n .

Putting these results together, we have $H = \bigoplus_{h \in Y} (\langle P_h \rangle \wr \mathbb{Z})$ is a soluble group, with derived length n . \square

We are now in position to prove Theorem 4.4.

Theorem 4.4. *Let C be a computable subgroup of $PL_+(I)$. The soluble subgroup recognition problem is solvable for C ; that is, there is an algorithm which, upon input of a finite subset X of C , can determine whether or not the subgroup $\langle X \rangle$ generated by X is a soluble group. Moreover, in the case that the group $\langle X \rangle$ is soluble, the algorithm also determines its derived length.*

Proof. Suppose C is a computable subgroup of $PL_+(I)$, and f_1, f_2, \dots, f_m are elements of C input to the algorithm, where m is a positive integer. Let $G := \langle f_1, f_2, \dots, f_m \rangle$. Then $G \leq C$; hence G is also computable.

Before giving the technical details of the algorithm, we begin with an overview of our procedure. In the algorithm below, we build the tree of towers (up to conjugation equivalence of towers) for the group $S(G)$. We apply a breadth-first-search to the tree of nested orbitals of these towers (successively moving left to right through all orbitals at the least depth before moving on to orbitals with greater depth), looking for complex overlaps.

In the steps of this algorithm we maintain a finite set SO of signed orbitals of the split group $S(G)$. For any collection S' of signed orbitals, there is an associated *signature group*, denoted by $\text{sigGrp}(S')$, which is the group generated by the signatures of the orbitals in S' . At every step the set SO and its associated signature group will satisfy the following properties:

- SO.1* Each element of SO has a signature that is a one-bump factor of an element of G , and hence $\text{sigGrp}(SO)$ is a subgroup of $S(G)$.
- SO.2* The signature group $\text{sigGrp}(SO)$ contains G .

We will also maintain two disjoint sets \mathcal{S} and \mathcal{U} of orbitals (representing the “seen” and “unseen” orbitals, respectively), whose union is the collection of unsigned orbitals associated to the signed orbitals in SO . Each orbital O will arise from Process 5, and so will be stored by the algorithm in the format $(\text{inf}(O), \text{sup}(O))$; that is, by storing the endpoints of the interval O . The orbitals in \mathcal{S} will satisfy the properties

- S.1* No pair of signed orbitals in SO whose (unsigned) orbitals lie in \mathcal{S} forms a complex overlap.
- S.2* For every orbital A of \mathcal{S} , there is exactly one signed orbital in SO with A as its support; we denote this signed orbital $\sigma_A = (A, h_A)$.
- S.3* For every orbital A of \mathcal{S} , there is a point $r_A \in A$ such that for every $A' \in \mathcal{S}$ with $A' \subsetneq A$, the containment $A' \subseteq (r_A, r_A \cdot h_A)$ also holds.

Note that these properties imply that the set $Z := \{h_A \mid A \in \mathcal{S}\}$ of signatures associated to the orbitals in \mathcal{S} satisfies conditions Z0-Z3 of the definition of a set of one-bump functions with fundamental domains, but properties *S.1-S.3* also include a partial extension of Z0-Z3 to SO .

We further partition \mathcal{U} into sets *Top* and *Lower*, to keep track of the order in which orbitals will be processed. Some of the orbitals O in \mathcal{U} and all of the orbitals in \mathcal{S} will be assigned an “orbital depth value” $\text{orbDepth}(O)$, which is a lower bound on the numerical value of the orbital depth of O in the group $\text{sigGrp}(SO)$; in particular, $\text{orbDepth}(O) = n$ will mean that the algorithm has found an exemplary tower of height n associated with $\text{sigGrp}(SO)$ with a signed orbital of the form $(O, g) \in SO$ at the bottom.

As our computation proceeds, new (signed or unsigned) orbitals will be added to SO and \mathcal{U} , and in other steps element orbitals will move from \mathcal{U} to \mathcal{S} or will be removed from SO or \mathcal{U} . Our calculation will terminate either when the algorithm detects either a complex overlap in the group $\text{sigGrp}(SO)$ or an orbital that is ‘too deep’, or else (soon) after all orbitals

have been removed from \mathcal{U} , so that $\mathcal{U} = \emptyset$. We will process the set \mathcal{U} carefully, keeping track of the height of towers that have been found, so that we will be guaranteed that the algorithm will stop if it finds no complex overlaps.

Throughout the description of the algorithm we also include proofs that the sets SO and \mathcal{S} have the properties $SO.1, SO.2$, and $\mathcal{S}.1, \mathcal{S}.2, \mathcal{S}.3$ respectively, as well as other commentary adding information about the steps along the way. In order to distinguish between steps of the algorithm and explanations of its validity, we number and indent the steps of the algorithm. The remaining bulk of the proof that the algorithm is valid is provided after all of the steps have been described.

Start of algorithm

Step 0 (*Setting up the algorithm*):

0.1 Let SO , \mathcal{U} , \mathcal{S} , Top , and $Lower$ be empty sets. Let $\text{maxDepth} := 0$ and $\text{counter} := 0$.

(Note that \mathcal{S} satisfies properties $\mathcal{S}.1$, $\mathcal{S}.2$, and $\mathcal{S}.3$ here.)

0.2 For each input element f_i : Apply Process 5 to compute the tuple $X_{f_i} = [A_{i1}, A_{i2}, \dots, A_{ik_i}]$ of orbitals of f_i . Next use Process 6 to compute the corresponding k_i signed orbitals (A_{ij}, \tilde{f}_{ij}) associated to the split group $S(G)$, where the \tilde{f}_{ij} are the one-bump functions associated to f_i (that is, \tilde{f}_{ij} equals f_i over A_{ij}). Add the pairs (A_{ij}, \tilde{f}_{ij}) to the set SO and add the orbital parts A_{ij} to \mathcal{U} .

Note that since the set SO contains the set of factor signed orbitals of the generating set $X := \{f_1, \dots, f_m\}$ of G , this set SO satisfies properties $SO.1$ and $SO.2$.

0.3 Compute the value $n := |\mathcal{B}_X|$ (the total number of breakpoints of elements of X) using Processes 2 and 4.

Step 1 (*Building Top and Lower from \mathcal{U}*):

1.1 Check whether $\mathcal{U} = \emptyset$. If so, then terminate the algorithm and output “The group G is soluble with derived length maxDepth ”.

1.2 Determine, using Process 4, whether or not $\mathcal{U} \cup \mathcal{S}$, and therefore SO , contains a complex overlap. If so, then terminate the algorithm and output “The group G is not soluble.”

1.3 For all A, B in \mathcal{U} : Using Process 4, determine whether $\bar{A} \subset B$, and if so add A to $Lower$.

1.4 Let $Top :=$ the complement of $Lower$ in \mathcal{U} .
Let $\text{counter} := \text{counter} + 1$.

The variable `counter` is used to record, for use in Step 2.1, whether Step 1 has been performed more than once; after Steps 1-3 are done, the algorithm can loop back to Step 1 again.

Step 2 (*Processing the orbitals in Top to detect excessive depth*):

- 2.1 If `counter` = 1, then for all orbitals $A \in Top$, assign the value $\text{orbDepth}(A) := 1$. Otherwise, if `counter` > 1, then for each orbital $A \in Top$, assign the value $\text{orbDepth}(A) := 1 +$ the number of orbitals in \mathcal{S} that contain A . (This requires Process 4.)
- 2.2 Compute $\text{maxDepth} := \max(\{\text{orbDepth}(A) \mid A \in Top\} \cup \{\text{maxDepth}\})$. If $\text{maxDepth} > n$, terminate the algorithm and output “The group G is not soluble.”

Note that since this maximum is taken with the old value of maxDepth included, successive occurrences of Step 2.2 cannot decrease the value of maxDepth .

Step 3 (*Processing the leftmost element of Top*):

- 3.1 Among the orbitals in Top with the smallest value of orbDepth , find the leftmost orbital (via Process 4), which we denote (a, b) throughout this step.
Let $Y = \{g_1, g_2, \dots, g_q\}$ be the set of signatures associated to the signed orbitals of SO whose orbital is (a, b) .

(The current occurrence of Step 3.1 is the start of the next step in our breadth-first-search.)

Step 3a (*Building a local controller c over (a, b)*):

- 3.2 For $1 \leq i \leq q$: Compute the slopes $m_{g_i a} := ag'_+$ and $m_{g_i b} := bg'_-$ (Process 7). Let $M_{aY} := \{m_{ga} \mid g \in Y\}$ and $M_{bY} := \{m_{gb} \mid g \in Y\}$.
- 3.3 Determine whether each of the groups $\Pi_{M_{aY}} = \langle M_{aY} \rangle$ and $\Pi_{M_{bY}} = \langle M_{bY} \rangle$ is discrete (Process 9). If either of these groups is not discrete, terminate the algorithm and output “The group G is not soluble.”
- 3.4 Using Process 9 again, compute integers p_1, p_2, \dots, p_q such that $m_{g_1 a}^{p_1} m_{g_2 a}^{p_2} \cdots m_{g_q a}^{p_q}$ is the least real number $\Pi_{M_{aY}, s}$ greater than 1 in the group $\Pi_{M_{aY}}$. Compute the element $c := g_1^{p_1} g_2^{p_2} \cdots g_q^{p_q}$. (Process 1).

Note that by construction, c has an orbital with a at its left endpoint.

- 3.5 Determine whether the orbital of c with left endpoint a and the orbital (a, b) of Top form a complex overlap (Processes 5 and 4). If so, then terminate the algorithm and output “The group G is not soluble.”

- 3.6 Compute the slope $m_{cb} := bc'_-$ of c at the right endpoint b of the orbital (a, b) using Process 7. Also compute the least real number $\Pi_{M_{bY}, s}$ greater than 1 in the group $\Pi_{M_{bY}}$ (Processes 9 and 8). If $m_{cb} \neq \Pi_{M_{bY}, s}$, then terminate the algorithm and output “The group G is not soluble.”
- 3.7 For $1 \leq i \leq q$: By iterating over successively larger positive and negative integers k , computing c^k (Process 1) and its slope $a(c^k)'_+$ to the right of a (Process 7), and comparing this slope to $m_{g_i a}$ (Process 8), find the unique integer l_{ia} such that the slope $m_{g_i a}$ is equal to the slope $a(c^{l_{ia}})'_+$. If $b(c^{l_{ia}})'_-$ is not equal to $m_{g_i b}$, then terminate the algorithm and output “The group G is not soluble.”

(Note that the justification for the outputs of Steps 3.3, 3.6, and 3.7 is given below after the completion of the algorithm.)

Step 3b (*Altering the orbital data sets*):

- 3.8 Add the signed orbital $((a, b), c)$ to SO .

Since all of the factors in the formula defining c in Step 3.4 realise the orbital (a, b) , Lemma 4.2(1) says that c is a one-bump factor of an element of G . Hence Step 3.8 preserves properties $SO.1$ and $SO.2$ of the set SO .

- 3.9 For $1 \leq i \leq q$: Calculate the element $h_i := g_i c^{-l_{ia}}$ of $S(G)$ (via Process 1). Use Process 5 to build the orbital tuple

$$X_{h_i} = [B_{i1}, B_{i2}, \dots, B_{ik_i}]$$

for h_i . For each orbital B_{ij} use Process 6 to produce the factor signed orbitals (B_{ij}, \tilde{h}_{ij}) , where \tilde{h}_{ij} is the one-bump function which agrees with h_i over B_{ij} . Add the signed orbitals (B_{ij}, \tilde{h}_{ij}) to SO , and add the orbitals B_{ij} to \mathcal{U} .

Note that we have $\overline{B_{ij}} \subset (a, b)$ for all i, j . Each of the factors in the product $g_i c^{-l_{ia}}$ defining h_i in Step 3.8 realise the orbital (a, b) , and so Lemma 4.2(1) says that each of the one-bump factors \tilde{h}_{ij} of h_i is also a one-bump factor of an element of G . Hence Step 3.9 also preserves properties $SO.1$ and $SO.2$ of the set SO .

(Note that in both Steps 3.8 and 3.9, we may not be adding new signed orbitals to SO each time; it may be the case, for example, that the signed orbital already lies in SO due to other generators of G .)

- 3.10 For $1 \leq i \leq q$: Determine whether $g_i = c$ (using a combination of Processes 2, 4, 7, and 8). If not, then remove the signed orbital $((a, b), g_i)$ from SO .

Since $g_i = h_i c^{l_{ia}}$ and $h_i, c \in \text{sigGrp}(SO)$ after Step 3.10 is applied, the group $\text{sigGrp}(SO)$ is not altered in this step. Hence $SO.1$ and $SO.2$ are again preserved.

Note that the signed orbital $((a, b), c)$ is now the only element of SO with support (a, b) .

Step 3c (*Checking for complex overlaps beneath c*):

- 3.11 Determine, using Procedure 4, whether or not SO contains a complex overlap. If so, then terminate the algorithm and output “The group G is not soluble.”
- 3.12 Compute the set $Proj_{(a,b)}$ of all of the signed orbitals in SO with support in (a, b) which do not realise the orbital (a, b) (Process 4). If $Proj_{(a,b)} = \emptyset$, then go to Step 1.

Note that since SO has no complex overlaps (after Step 3.11), each orbital $((r, s), g)$ in $Proj_{(a,b)}$ satisfies $a < r < s < b$. Since c has (a, b) as its only orbital and the slope ac'_+ is greater than 1, we also have $r < r \cdot c$.

- 3.13 Determine whether the orbitals $((r, s), g)$ and $((r \cdot c, s \cdot c), g^c)$ form a complex overlap; to do this, it suffices to check, using Processes 3 and 4, whether $r \cdot c < s$. If so, terminate the algorithm and output “The group G is not soluble.”

Note that if the algorithm continues after Step 3.13, then since $x \cdot c > x$ for all $x \in (a, b)$, we have that each $((r, s), g) \in Proj_{(a,b)}$ satisfies $(r, s) \subseteq (r, r \cdot c)$.

- 3.14 For each pair of elements $\rho = ((r, s), g), \sigma = ((u, v), h)$ of $Proj_{(a,b)}$: By iteratively computing $u \cdot c^i$ for positive and negative integers i (Processes 1 and 3) and comparing with r (Process 4), compute the unique integer $k_{\rho\sigma}$ such that $r \leq u \cdot c^{k_{\rho\sigma}} < r \cdot c$. Construct the signed orbitals $\tau'_{\rho\sigma} := ((u, v) \cdot c^{k_{\rho\sigma}-1}, h^{c^{k_{\rho\sigma}-1}})$ and $\tau_{\rho\sigma} := ((u, v) \cdot c^{k_{\rho\sigma}}, h^{c^{k_{\rho\sigma}}})$ (and store them for use in later steps). Use Process 4 to determine whether $\tau'_{\rho\sigma}$ or $\tau_{\rho\sigma}$ yield a complex overlap with ρ . If so, terminate the algorithm and output “The group G is not soluble.”

Note that continuation of the algorithm after Step 3.14 implies that one of the following must hold:

- (i) $(u, v) \cdot c^{k_{\rho\sigma}} = (r, s)$ (ii) $\overline{(u, v)} \cdot c^{k_{\rho\sigma}} \subset (r, s)$
- (iii) $(u, v) \cdot c^{k_{\rho\sigma}} \subset (s, r \cdot c)$ (iv) $(u, v) \cdot c^{k_{\rho\sigma}} \supset \overline{(r \cdot c, s \cdot c)}$.

In case (iv) we have $\overline{(r, s)} \cdot c^{-k_{\rho\sigma}+1} = \overline{(r, s)} \cdot c^{k_{\sigma\rho}} \subset (u, v)$.

- 3.15 Determine a partial order \prec on $Proj_{(a,b)}$ as follows. For each ordered pair of elements $\rho = ((r, s), g), \sigma = ((u, v), h)$ of $Proj_{(a,b)}$: Determine whether $v \cdot c^{k_{\rho\sigma}} < s$ (and hence whether the unsigned orbital associated to $\tau_{\rho\sigma}$ satisfies $\overline{(u, v)} \cdot c^{k_{\rho\sigma}} \subset (r, s)$), using Process 4 and the stored $\tau_{\rho\sigma}$. If so, we add $\sigma \prec \rho$ to the relation. After this has been completed for all ordered pairs, determine a left-most element $\rho_0 = ((r_0, s_0), g)$ of $Proj_{(a,b)}$ that is maximal with respect to the relation \prec . (The choice of ρ_0 might not be unique,

because two distinct maximal signed orbitals might share the same support.)

To see that the relation \prec is antisymmetric, suppose that ρ and σ are as in Step 3.15 with $\rho \prec \sigma \prec \rho$. Then $(r, s) \cdot c^{k_{\sigma\rho} + k_{\rho\sigma}} \subset (u, v) \cdot c^{k_{\rho\sigma}} \subset (r, s)$, which is impossible since c moves all points to the right on (a, b) and so cannot conjugate an orbital in (a, b) inside itself.

From Step 3.13, we have that the support (r_0, s_0) of ρ_0 satisfies $(r_0, s_0) \subseteq (r_0, r_0 \cdot c)$. From the note after Step 3.14, maximality of ρ_0 implies that every signed orbital $\sigma = ((u, v), h)$ of $Proj_{(a,b)}$ satisfies either

- (i) $(u, v) \cdot c^{k_{\rho_0\sigma}} = (r_0, s_0)$,
- (ii) $(u, v) \cdot c^{k_{\rho_0\sigma}} \subset (r_0, s_0)$, or
- (iii) $(u, v) \cdot c^{k_{\rho_0\sigma}} \subset (s_0, r_0 \cdot c)$.

Therefore $(u, v) \cdot c^{k_{\rho_0\sigma}} \subset (r_0, r_0 \cdot c)$. That is, for every signed orbital $\sigma \in Proj_{(a,b)}$, the support of the signed orbital $\tau_{\rho_0\sigma}$ is contained in the interval $(r_0, r_0 \cdot c)$.

- 3.16 For each element $\sigma = ((u, v), h)$ of $Proj_{(a,b)}$: Add the signed orbital $\tau_{\rho_0\sigma}$ to SO , and add the associated orbital $(u, v) \cdot c^{k_{\rho_0\sigma}}$ to \mathcal{U} .

Now Lemma 4.2(1) says that powers of c are one-bump factors of elements of G . In Step 3.16, since $(u, v) \subseteq (a, b)$, Lemma 4.2(2) says that the signature $h^{c^{k_{\rho_0\sigma}}}$ of $\tau_{\rho_0\sigma}$ is also a one-bump factor of an element of G . Therefore Step 3.16 preserves properties $SO.1$ and $SO.2$ of the set SO .

- 3.17 Determine, using Procedure 4, whether or not SO contains a complex overlap. If so, then terminate the algorithm and output “The group G is not soluble.”

(Step 3.17 is not strictly necessary, since Steps 3.14 and 3.15 guarantee that no new complex overlap is added to SO in 3.16; we include Step 3.17 to highlight the fact that SO does not contain a complex overlap in the following steps.)

- 3.18 For each unsigned orbital $(u, v) \in \mathcal{U}$ such that there is an element $h \in PL_+(I)$ with $((u, v), h) \in Proj_{(a,b)}$ and $(u, v) \cap (r_0, r_0 \cdot c) = \emptyset$: Remove all signed orbitals from SO whose associated unsigned orbital is (u, v) , and remove the unsigned orbital (u, v) from \mathcal{U} .

For any orbital $\sigma = ((u, v), h)$ removed from SO in Step 3.18, the related element $\tau_{\rho_0\sigma}$ with signature $h^{c^{k_{\rho_0\sigma}}}$ (added to SO in Step 3.16) remains in SO . Since $((a, b), c) \in SO$ as well (from Step 3.8), we have $h^{c^{k_{\rho_0\sigma}}}, c \in \text{sigGrp}(SO)$, and so $h \in \text{sigGrp}(SO)$ after Step 3.18 is complete. That is, Step 3.18 does not alter the group $\text{sigGrp}(SO)$, and so step 3.18 preserves properties $SO.1$ and $SO.2$ of the set SO .

- 3.19 Remove the orbital (a, b) from \mathcal{U} and add it to \mathcal{S} .

Since $\mathcal{U} \cup \mathcal{S}$ had no complex overlaps in Step 3.17, and no orbitals were added to this set in the intermediate Step 3.18, then Step 3.19 preserves property $\mathcal{S}.1$. The fact that Step 3.19 preserves property $\mathcal{S}.2$ of the set \mathcal{S} follows from the fact that Step 3.10 has been performed for the orbital (a, b) in the current instance of Step 3, and property $\mathcal{S}.3$ follows from Steps 3.13 through 3.18.

3.20 Proceed again to Step 1.

End of algorithm

It remains to show that this algorithm will terminate on every possible input, and that when it terminates, it outputs the correct answer. We begin with the latter.

In Step 0 of this algorithm, a set SO satisfying properties $SO.1$ and $SO.2$ is computed, and in all subsequent steps in which the set SO is changed, namely Steps 3.8, 3.9, 3.16, and 3.18, these two properties have been shown to be preserved. Therefore the signature group $\text{sigGrp}(SO)$ associated to SO satisfies $G \leq \text{sigGrp}(SO) \leq S(G)$. Now Corollary 4.1(1) says that the derived length of G equals the derived length of $\text{sigGrp}(SO)$ throughout the algorithm.

In all steps in which a complex overlap is found among the elements of SO , namely Steps 1.2, 3.5, 3.11, 3.13, 3.14, and 3.17, we have that the signature group $\text{sigGrp}(SO)$ admits a complex overlap. Corollary 4.1(2) then shows that G is not soluble, verifying the output of these six steps.

In Step 2.2, if maxDepth is found to be greater than the number n of breakpoints among the finite set of homeomorphisms in the input to the algorithm, then the algorithm has found an orbital $A \in \mathcal{U}$ that is contained in at least n orbitals in \mathcal{S} . Now $\mathcal{U} \cup \mathcal{S}$ is the set of unsigned orbitals associated to the set SO of signed orbitals, and at this step we know that the set SO contains no complex overlaps (from Step 1.2). Thus the n unsigned orbitals in \mathcal{S} together with A arise from $n+1$ signed orbitals in SO that form a tower of height $n+1$, with the orbital associated to A at the “bottom”, and so the orbital depth of A with respect to the signature group $\text{sigGrp}(SO)$ must be at least $\text{orbDepth}(A) > n$. Then Theorem 2.3 implies that the derived length of $\text{sigGrp}(SO)$ is at least $n+1$. Since G and $\text{sigGrp}(SO)$ have the same derived length, then G must have derived length at least $n+1$. However, Theorem 3.1 shows that if G is soluble, then its derived length must be at most n . Hence the “not soluble” output of Step 2.2 is valid.

To show that Step 3.3 is valid, we consider the subgroup $H := \langle g_1, \dots, g_q \rangle$ of $\text{sigGrp}(SO)$, with the single orbital (a, b) . Note that if one of the groups $\Pi_{M_a Y}$ and $\Pi_{M_b Y}$ of Step 3.3 is not a discrete group, then that group is neither the trivial group nor isomorphic to \mathbb{Z} . Lemmas 3.10 and 3.11 of [3] show that in this case the group H is not balanced. Lemma 2.7 then H is not chainless, and so by Theorem 3.1, H is not soluble. Thus the group $\text{sigGrp}(SO)$

contains a nonsoluble subgroup, and so also is nonsoluble. Therefore G is not soluble, as required.

Suppose next that the conditions of Step 3.6 hold, namely that the slope $m_{cb} = bc'_-$ of c in a neighborhood to the left of b satisfies $m_{cb} \neq \Pi_{M_{bY},s}$, where $\Pi_{M_{bY},s}$ is the least number greater than 1 in the discrete group $\Pi_{M_{bY}}$. Since the algorithm did not terminate at Step 3.5, the element c of $\text{sigGrp}(SO)$ cannot have a fixed point in the interval (a, b) , and since its slope ac'_+ on the right at a is greater than 1, we must have $bc'_- > \Pi_{M_{bY},s}$. Let d be the element of $\text{sigGrp}(SO)$ defined by $d := g_1^{\tilde{p}_1} \cdots g_q^{\tilde{p}_q}$. Then the support of d lies in (a, b) , and the slope bd'_- of d from the left in a neighborhood of the endpoint b is the number $\Pi_{M_{bY},s}$. If d has a fixed point in the interval (a, b) , then the group $\text{sigGrp}(SO)$ has a complex overlap. On the other hand, if d does not have a fixed point in (a, b) , then the slope ad'_+ of d from the right at a must be greater than 1, and so is greater than or equal to the slope $\Pi_{M_{aY},s} = ac'_+$ of c at a . Now the element cd^{-1} of $\text{sigGrp}(SO)$ has an orbital of the form (a', b) for some $a < a' < b$, and so $\text{sigGrp}(SO)$ again admits a complex overlap. Theorem 3.1 says that $\text{sigGrp}(SO)$ is not soluble in both cases, and so G also is not soluble. This verifies the output of Step 3.6.

Next suppose that the condition of Step 3.7 holds; that is, suppose that there is an index i such that $l_{ia} \neq l_{ib}$. Let $d := g_i c^{-l_{ia}}$. Then $d \in \text{sigGrp}(SO)$, the support of d is a subset of (a, b) , and d fixes an open neighborhood of a . However, the slope bd'_- of d at b from the left is not 1. Therefore $\text{sigGrp}(SO)$ again admits a complex overlap, and so G is not soluble, so Step 3.7 is also valid.

The last output step left to check is the only step that outputs that the group G is soluble, namely Step 1.1. Suppose that the set \mathcal{U} is found to be empty in an occurrence of Step 1.1. If `counter` = 0, and so the algorithm terminates at the first occurrence of Step 1.1, then G is the trivial group, and the algorithm correctly outputs the value 0 for the derived length. On the other hand, suppose that `counter` > 0, and so this the algorithm terminates at a later occurrence of Step 1.1. From Steps 0.1 and 3.19, we know that the unsigned orbitals in the set \mathcal{S} satisfy properties $\mathcal{S}.1$, $\mathcal{S}.2$, and $\mathcal{S}.3$. Thus the set $Z := \{h \mid (A, h) \in SO\}$ is a finite generating set of $\text{sigGrp}(SO)$ that satisfies properties Z0-Z3 of the definition of a set of one-bump functions with fundamental domains. Lemma 4.3 says that the group $\text{sigGrp}(SO)$ is soluble, with derived length equal to the height of the largest tower that can be formed from orbitals in the set SO . The algorithm adds unsigned orbitals in order of containment (larger intervals before smaller), and so the value of `maxDepth` from the last instance of Step 2.2 will be the derived length of $\text{sigGrp}(SO)$. Again using the fact that G and $\text{sigGrp}(SO)$ have the same derived length, this shows that the output of Step 1.1 is valid.

Finally we turn to the proof that the algorithm will terminate on all possible inputs. In Step 0.2 of the algorithm, the set SO is built from the

one-bump factors of the finite set $\{f_1, \dots, f_m\}$ of input functions, and the finite set \mathcal{U} of associated unsigned orbitals is created. The only step in which the set \mathcal{U} is altered is Step 3; in particular, Steps 3.9, 3.16, and 3.18. Each time Step 3 is performed, an orbital of smallest value of `orbDepth` is removed from \mathcal{U} , as well as possibly some others of greater orbital depth, and a finite (possibly zero) number of orbitals of strictly larger depth are added to \mathcal{U} . After a finite number of iterations of Step 3, then, the least value of `orbDepth` of an element of \mathcal{U} must increase or else \mathcal{U} must become empty. In the case that the group $G = \langle f_1, \dots, f_m \rangle$ is soluble this implies that \mathcal{U} must eventually be empty after a finite number of occurrences of Step 3, causing the algorithm to terminate at Step 1.1. In the case that G is not soluble, this means that either the algorithm must halt in one of Steps 1.2, 3.3, 3.6, 3.7, 3.11, 3.13, 3.14, or 3.17, or else after a finite number of steps the smallest value of `orbDepth` among the elements of $Top \subseteq \mathcal{U}$ is greater than the number n of breakpoints of the f_i input functions, causing the algorithm to terminate at Step 2.2. \square

Remark 4.5. In the proof of Theorem 4.4, every element of $S(G)$ involved in the computations throughout the procedure can be shown to be the one-bump factor of an element of G , using Lemma 4.2. If the algorithm stores the element of G with each of these one-bump factors, then in each application of Processes 1-9, it is possible for the algorithm instead to perform the procedure with the corresponding elements of G , and then apply Process 6, in order to accomplish the process for the element of $S(G)$. At some potential cost in efficiency, then, Theorem 4.4 also holds for groups C admitting Processes 1-9 in which the group $S(C)$ is replaced by C in each of the process statements.

We note that it may be possible to make this algorithm more efficient; in particular, some repeated steps may be streamlined. It is of interest to consider whether a different strategy for choosing the element of Top to consider in the next occurrence of Step 3.1, for example with a depth-first-search instead, would improve efficiency. We also note that the algorithm can be made parallel in various ways, for example by processing all elements of least `orbDepth` value in Top simultaneously, while the sets SO , \mathcal{U} , and \mathcal{S} and the value `maxDepth` are treated as global objects in shared memory.

We also note that if a finite generating set X of a split soluble group G is input to the algorithm in Theorem 4.4, then at the time the algorithm terminates (at an instance of Step 1.1), the set X' of signatures of the elements of SO satisfies the properties that $G = \langle X' \rangle$ (since $SO.1$ - $SO.2$ hold and so $G = S(G) = \text{sigGrp}(SO)$) and X' is a set of one-bump functions with fundamental domains (since $\mathcal{S} = SO$ and properties $SO.1$ and $\mathcal{S}.1$ - $\mathcal{S}.3$ hold). Hence the following is immediate from the algorithm of Theorem 4.4.

Corollary 4.6. *Let C be a computable subgroup of $PL_+(I)$. There is an algorithm which, upon input of any finite generating set of a soluble subgroup*

H of C satisfying $H = S(H)$, outputs a finite set of one-bump functions with fundamental domains that generate H .

Next, we turn to the solution of the membership decision problem for finitely generated soluble subgroups of computable subgroups of $PL_+(I)$ that are generated by a finite set of one-bump functions with fundamental domains.

Corollary 4.7. *Let C be a computable subgroup of $PL_+(I)$. Let H be a subgroup of C generated by a finite set of one-bump functions with fundamental domains. Then the membership decision problem is solvable for H ; that is, there is an algorithm which, upon input of an element w of C , can determine whether $w \in H$.*

Proof. Let Z be the finite set of one-bump functions with fundamental domains generating H . For each $h \in Z$, we replace h by h^{-1} , if necessary, so that we may assume that the slope $m_{ha} = ah'_+$ of h at the left endpoint $a = \inf \text{Supp}(h)$ of its support satisfies $m_{ha} > 1$.

We first show that H is equal to the split group $S(H)$. Following the notation of the proof of Lemma 4.3, let n be the largest height of a tower in the set $S_Z = \{(A_h, h) \mid h \in Z\}$ (where $A_h = \text{Supp}(h)$) of signed orbitals associated to the elements of Z . If $n = 0$, then Z is empty and $S(H) = H = 1$ is the trivial group. If $n = 1$ then the elements of Z have disjoint support, H is the free abelian group generated by the elements of Z , and again $S(H) = H$. Now suppose that $n > 1$ and the result holds for finite sets of one-bump functions with fundamental domains with maximum associated tower height at most $n - 1$. Suppose that g' is any one-bump factor of an element $g \in H$. Recall from the proof of Lemma 4.3 that $H = \bigoplus_{h \in Y} (\langle P_h \rangle \wr \mathbb{Z})$ where $Y = \{h' \in Z \mid \text{orbDepth}(A_{h'}) = 1\}$ is the set of elements of minimal orbital depth in Z , and such that for each $h \in Y$ the set $P_h := \{h' \in H \mid A_{h'} \subsetneq A_h\}$ is the set of elements of Z whose support is properly contained in the support A_h of h . Since the support of the group H is $\cup_{h \in Y} A_h$, we have $\text{Supp}(g') \subseteq A_h$ for some $h \in Y$. Now the element $g \in \langle Z \rangle$ is a product of an element \tilde{g} of $\langle h, P_h \rangle = \langle P_h \rangle \wr \langle h \rangle$ with an element of $\langle Z \setminus (\{h\} \cup P_h) \rangle$ whose support does not intersect A_h . Moreover, \tilde{g} is another element of H that has g' as a one-bump factor. We can write $\tilde{g} = \hat{g}h^k$ for some $\hat{g} \in \bigoplus_{j \in \mathbb{Z}} \langle P_h \rangle^{h^j}$ and $k \in \mathbb{Z}$. Moreover, \hat{g} can be written as a product of elements of a finite subset Q of $\bigcup_{j \in \mathbb{Z}} \langle P_h \rangle^{h^j}$ that is a set of one-bump functions with fundamental domains with maximum associated tower height at most $n - 1$. If $k = 0$ then g' is a one-bump factor of $\tilde{g} = \hat{g} \in \langle Q \rangle$, and so g' is an element of the split group $S(\langle Q \rangle)$. By the inductive assumption above, $S(\langle Q \rangle) = \langle Q \rangle$; hence in the $k = 0$ case, $g' \in \langle Q \rangle < H$. On the other hand, if $k \neq 0$, then since the supports of the elements in $\bigcup_{j \in \mathbb{Z}} \langle P_h \rangle^{h^j}$ do not share an endpoint of A_h , the support of \tilde{g} includes intervals with endpoints that are the endpoints of A_h . Since H is soluble (Lemma 4.3), Theorems 3.1 and 2.5 show that $S(H)$ does not admit a complex overlap, and so we have $\text{Supp}(\tilde{g}) = A_h$. In this case \tilde{g}

is already a one-bump function, and so $g' = \tilde{g}$. Thus again we have $g' \in H$. Hence $S(H) = H$, as claimed.

Next we note that upon input of the set Z to the algorithm of Theorem 4.4, no orbitals are added or removed from the set SO after Step 0.2, and the algorithm will terminate at an instance of Step 1.1, with $SO = \mathcal{S} = S_Z$, and output the derived length n of H .

Finally we are ready to give the MDP algorithm. Input the set $Z \cup \{w\}$ to the SSRP algorithm of Theorem 4.4. At step 0.2, the algorithm will place the signed orbitals of the one-bump factors of w into the set SO ; since $S(H) = H$, these factors lie in H iff w lies in H . Proceeding through the algorithm, if at any time the SSRP algorithm outputs “The group G is not soluble” or “The group G is soluble with derived length m ” where m is greater than the derived length n of H , then the present (MDP) algorithm outputs “The element w is not in H ”. For the rest of this proof we assume that the eventual output of the SSRP algorithm (with input $Z \cup \{w\}$) is “The group G is soluble with derived length n ”.

The MDP algorithm uses a slight restriction on Step 3 of the SSRP procedure, to ensure that no signed orbital associated to an element of Z is removed from the set SO . Each time that the SSRP algorithm reaches Step 3.1, since $Z \subseteq SO$, the breadth-first-search structure of the SSRP algorithm, processing intervals of least orbital depth first, guarantees that for all $h' \in Z$ satisfying $A_{h'} \supsetneq (a, b)$, Step 3 has already been performed for the interval $A_{h'}$. Also condition Z2 for the set Z implies that the set Y of elements of SO with support (a, b) contains at most one element of Z . Suppose first that Y does not contain any element of Z ; that is, all elements w' of Y are derived from w via earlier Steps 0.2, 3.9, and 3.16 of the SSRP. Then the group $\langle Z \cup \{w\} \rangle$ contains a subgroup $\langle Y \rangle$ not in $\langle Z \rangle$, and so the MDP algorithm halts and outputs “The element w is not in H ”. Next suppose that $Y = \{h\}$ is a singleton set whose element h is in Z . Then the only substep of Step 3 that has an effect is Step 3.19, moving the orbital $A_h = (a, b)$ from \mathcal{U} to \mathcal{S} ; then the SSRP algorithm returns to Step 1. Finally suppose that Y contains an element h of Z and $|Y| > 1$. If the slope m_{ha} of the element h at a does not equal the slope $\Pi_{M_a Y}$ in the subsequent occurrence of Step 3.4, then the group of slopes at a of signed orbitals with support (a, b) for $\langle Z \cup \{w\} \rangle$ does not equal the same slope group for $\langle Z \rangle$, and so we stop and output “The element w is not in H ”. Otherwise, we can take $c = h$ in this round of Step 3.4. Continuing with Step 3, in Steps 3.9 and 3.10 the orbital associated to each $w' \in Y \setminus \{h\}$ in SO is replaced by signed orbitals of one-bump factors of a product of w' with a power of h , and in Steps 3.16 and 3.18 these orbitals may be replaced again by orbitals associated to conjugation of the signatures by a power of h . Again using $S(H) = H$, we have $w' \in H$ iff these (conjugates of) factors lie in H . Again in Step 3.19 the orbital $A_h = (a, b)$ is moved from \mathcal{U} to \mathcal{S} , and then the SSRP algorithm returns to Step 1.

Continue through the SSRP procedure and repeat the above process for all instances of Step 3. When the SSRP algorithm terminates, the MDP algorithm outputs “The element w is not in H ” unless the SSRP algorithm terminates at an instance of Step 1.1 with $SO = \mathcal{S} = S_Z$, in which case the output is “The element w is in H ”. \square

Finally, combining the algorithms in Corollaries 4.6 and 4.7 shows that membership in finitely generated split soluble subgroups of computable groups is uniformly solvable.

Corollary 4.8. *Let C be a computable subgroup of $PL_+(I)$. There is an algorithm which, upon input of an element w of C and a finite subset X of C generating a split soluble group, can determine whether $w \in \langle X \rangle$.*

REFERENCES

1. *Thompson’s group at 40 years, Preliminary Problem List*, 2004, available at <http://www.aimath.org/WWN/thompsonsgroup/thompsonsgroup.pdf>.
2. James Belk and Francesco Matucci, *Conjugacy and dynamics in Thompson’s groups*, *Geom. Dedicata* **169** (2014), 239–261. MR 3175247
3. Collin Bleak, *An algebraic classification of some solvable groups of homeomorphisms*, *J. Algebra* **319** (2008), no. 4, 1368–1397. MR 2383051 (2008k:20070)
4. ———, *A geometric classification of some solvable groups of homeomorphisms*, *J. Lond. Math. Soc. (2)* **78** (2008), no. 2, 352–372. MR 2439629 (2009g:20069)
5. ———, *A minimal non-solvable group of homeomorphisms*, *Groups Geom. Dyn.* **3** (2009), no. 1, 1–37. MR 2466019 (2010d:20049)
6. Matthew G. Brin, *The ubiquity of Thompson’s group F in groups of piecewise linear homeomorphisms of the unit interval*, *J. London Math. Soc. (2)* **60** (1999), no. 2, 449–460. MR 1724861 (2000i:20061)
7. ———, *Elementary amenable subgroups of R . Thompson’s group F* , *Internat. J. Algebra Comput.* **15** (2005), no. 4, 619–642. MR 2160570 (2007d:20052)
8. Matthew G. Brin and Craig C. Squier, *Groups of piecewise linear homeomorphisms of the real line*, *Invent. Math.* **79** (1985), no. 3, 485–498. MR MR782231 (86h:57033)
9. J.W. Cannon, W.J. Floyd, and W.R. Parry, *Introductory notes on Richard Thompson’s groups*, *Enseign. Math. (2)* **42** (1996), no. 3–4, 215–256. MR MR1426438 (98g:20058)
10. Gili Golan, *The generation problem in Thompson group F* , 2016, arXiv:math.GR/1608.02572.
11. V. S. Guba and M. V. Sapir, *On subgroups of the R . Thompson group F and other diagram groups*, *Mat. Sb.* **190** (1999), no. 8, 3–60. MR MR1725439 (2001m:20045)
12. Victor Guba and Mark Sapir, *Diagram groups*, *Mem. Amer. Math. Soc.* **130** (1997), no. 620, viii+117. MR MR1396957 (98f:20013)
13. Andrés Navas, *Quelques groupes moyennables de difféomorphismes de l’intervalle*, *Bol. Soc. Mat. Mexicana (3)* **10** (2004), no. 2, 219–244 (2005). MR MR2135961 (2006j:43006)
14. V Shpilrain and A. Ushakov, *Thompson’s group and public key cryptography*, ACNS 2005, *Lecture Notes in Comput. Sci.*, vol. 3531, Springer-Verlag, New York, 2005, pp. 151–164.

SCHOOL OF MATHEMATICS AND STATISTICS, UNIVERSITY OF ST. ANDREWS, NORTH
HAUGH ST ANDREWS, FIFE KY16 9SS, SCOTLAND

E-mail address: `cb211@st-andrews.ac.uk`

SCHOOL OF MATHEMATICS AND STATISTICS, UNIVERSITY OF ST. ANDREWS, NORTH
HAUGH ST ANDREWS, FIFE KY16 9SS, SCOTLAND

E-mail address: `tarabrough@gmail.com`

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF NEBRASKA, LINCOLN NE 68588-
0130, USA

E-mail address: `hermiller@unl.edu`