TWISTED CONJUGACY IN HOUGHTON’S GROUPS

CHARLES COX

Abstract. For a fixed $n \geq 2$, Houghton’s group $H_n$ consists of bijections of $X_n = \{1, \ldots, n\} \times \mathbb{N}$ that are ‘eventually translations’ of each copy of $\mathbb{N}$. These groups were shown to have solvable conjugacy problem in [ABM13]. In general this does not imply that all finite extensions and finite index subgroups have solvable conjugacy problem. In this note we show that, for $H_n$, it does.

1. Introduction

Given a presentation $\langle S \mid R \rangle = G$, a ‘word’ in $G$ is an ordered combination $a_1 \ldots a_f$ where $f \in \mathbb{N}$ and each $a_i$ is from $S \cup S^{-1}$. Dehn’s problems and their generalisations involve seemingly obvious questions about finite presentations. The main problems that we will consider are as follows.

- the word problem for $G$, denoted WP($G$): is there an algorithm which, given two words $a, b \in G$, decides (positively or negatively) whether $a =_G b$, i.e. do these words represent the same element of the group? This is clearly equivalent to asking whether $ab^{-1} =_G 1$. There exist finitely presented groups where this problem is undecidable (see [Nov58] or [Boo59]).
- the conjugacy problem for $G$, denoted CP($G$): is there an algorithm which, given two words $a, b \in G$, decides (positively or negatively) if they are conjugate by some $x \in G$? CP($G$) implies WP($G$) as $1_G$ has its own conjugacy class. It is known to be strictly weaker than WP($G$) since there exist groups where WP($G$) is solvable but CP($G$) is not (e.g see [Mil71]).
- the $\phi$-twisted conjugacy problem for $G$, denoted TCP$_\phi(G)$: is there an algorithm which, given two words $a, b \in G$, decides (positively or negatively) if they are $\phi$-twisted conjugate by some $x \in G$ (i.e. whether $(x\phi)^{-1}ax = b$ for some $x \in G$)? Notice that TCP$_{1\text{Id}}(G)$ is CP($G$).
- the (uniform) twisted conjugacy problem for $G$, denoted TCP($G$): is there an algorithm which, given a $\phi \in \text{Aut}(G)$ and two words $a, b \in G$, decides (positively or negatively) if they are $\phi$-twisted conjugate by some $x \in G$? There exist groups such that CP($G$) is solvable but TCP($G$) is not (e.g see [BMV10]).

Should any of these problems be solved for one finite presentation, then they may be solved for any other finite presentation of that group. We therefore say that such problems are solvable if an algorithm exists for one such presentation. These problems may also be considered for any group that is recursively presented.

If CP($G$) is solvable, then we do not have that finite index subgroups of $G$ or finite extensions of $G$ have solvable conjugacy problem, even if these are of degree 2. Explicit examples can be found for both cases (see [CM77] or [GK75]). Thus it is natural to ask, if CP($G$) is solvable, whether the conjugacy problem holds for finite extensions and finite index subgroups of $G$. In [BMV10] Thm 3.1, an extension
E of G having solvable conjugacy problem is linked with orbit decidability (see Section 5.1). Their theorem requires solving TCP(G). Therefore, in order to solve the conjugacy problem for all finite extensions of any Houghton group (denoted $H_n$ with $n \in \mathbb{N}$), we will solve TCP($H_n$) and show that the orbit decidability condition holds for any finite extension.

**Theorem 1.** Let $n \geq 2$. Then TCP($H_n$) is solvable.

**Theorem 2.** Let $n \geq 2$. Then all finite extensions of $H_n$ have solvable conjugacy problem.

Since the finite index subgroups of $H_n$ are of a simple form, we also obtain

**Theorem 3.** Let $n \geq 2$. Then all finite index subgroups of $H_n$ have solvable conjugacy problem.

We organise the results as follows. Section 2 introduces Houghton’s groups and reformulates twisted conjugacy in $H_n$ into conjugacy within Aut($H_n$) $\cong S_n \times H_n$. We then solve this reformulated problem using the ideas for conjugacy in [ABM13], namely: the sets used to describe orbits (Section 3), their algorithm for finding a conjugator in FSym (Section 4.1), their machinery developed to generalise the conjugator to being in FSym (Section 4.2), and their ideas for producing a bound for the translation lengths of a conjugator (Sections 4.3 and 4.4). We then prove Theorems 2 and 3 (in Sections 5.1 and 5.2 respectively).

**Acknowledgements.** The author thanks Yago Antolín, José Burillo, and Armando Martino, whose results and methods in [ABM13] are drawn upon extensively. The author additionally thanks Armando Martino, his supervisor, for his encouragement and the many helpful discussions which have made this paper possible.

2. Background

As with the authors of [ABM13], the author does not know of a class that contains Houghton groups and where the conjugacy problem has been solved.

2.1. Houghton’s groups. For a fixed $n \in \mathbb{N}$, let $X_n := \{1, 2, \ldots, n\} \times \mathbb{N}$. Arrange these $n$ copies of the positive integers so that the first point of each copy of $\mathbb{N}$ can be joined to make a regular $n$-gon (shown for $n = 5$ in Figure 1 below). Each copy of $\mathbb{N}$ is often referred to as a branch or ray and will let $(i, m)$ denote the $m^{th}$ point on the $i^{th}$ branch.

**Notation.** For a set $X$, the set of all bijections on $X$ form the group which we denote Sym($X$). Those bijections which have finite support (i.e. move finitely many points) form a normal subgroup which we will denote FSym($X$). If there is no ambiguity for $X$, then we will write just Sym or FSym respectively.

Note that, if $X$ is countable, then FSym($X$) is countably generated and Sym($X$) is uncountably generated.

**Definition 2.1.** The $n^{th}$ Houghton group, denoted $H_n$, consists of all elements $g \in$ Sym($X_n$) for which there exists a $z \in \mathbb{N}$ and $\{t_1(g), \ldots, t_n(g)\} \in \mathbb{Z}^n$ such that

$$
(i, m)g = (i, m + t_i(g)) \text{ for all } m \geq z.
$$

(1)
The vector \( t(g) := (t_1(g), \ldots, t_n(g)) \) represents the 'eventual translation length' for each \( g \) in \( H_n \). As \( g \) acts bijectively on \( X_n \), the translation lengths satisfy (2).

\[
\sum_{i=1}^{n} t_i(g) = 0
\]

Since the \( z \) from (1) may be arbitrarily large and \( t_i(g) \) may be 0 for all \( i \in \{1, \ldots, n\} \), we have that \( \text{FSym}(X_n) \leq H_n \). This means that (for all \( n \geq 2 \)) we have the short exact sequence

\[
1 \to \text{FSym}(X_n) \to H_n \xrightarrow{\pi} \mathbb{Z}^{n-1} \to 1
\]

where \( (g)\pi := (t_1(g), \ldots, t_{n-1}(g)) \).

Figure 1. The set \( X_5 \) and action of the standard generators \( g_2, g_4 \in H_5 \)

These groups were introduced in [Hou78] for \( n \geq 3 \). The standard generating set that we will use when \( n \geq 3 \) is \( \{g_2, g_3, \ldots, g_n\} \) where for each \( i \),

\[
(j, m)g_i = \begin{cases} 
(1, m + 1) & \text{if } j = 1 \\
(1, 1) & \text{if } j = i, m = 1 \\
(i, m - 1) & \text{if } j = i, m > 1 \\
(j, m) & \text{otherwise.}
\end{cases}
\]

Notice that for each \( i \), we have \( t_1(g_i) = 1 \) and \( t_j(g_i) = -\delta_{i,j} \) for \( j \in \{2, \ldots, n\} \). Figure 1 shows a geometric visualisation of two such \( g_i \).

There are two observations to see that \( \{g_i \mid i = 2, \ldots, n\} \) generates \( H_n \) for any \( n \geq 3 \). First, any valid translation lengths (those satisfying (2)) can be obtained by these generators. Second, the commutator (defined as \( [g, h] := g^{-1}h^{-1}gh \) for \( g, h \) in \( G \)) of any two distinct elements \( g_i, g_j \in H_n \) is a 2-cycle, and so conjugation of this 2-cycle by an appropriate choice of elements from \( \{g_i \mid i = 2, \ldots, n\} \) will produce a 2-cycle with support equal to any two points of \( X_n \). This is enough to construct any element that is 'eventually a translation', i.e. any element that satisfies condition
and so is enough to generate all of $H_n$. An explicit presentation for $H_3$ can be found in [Joh99] and this was generalised in [Lee12] for all $n \geq 3$.

We now produce generating sets for $H_1$ and $H_2$. If $g \in H_1$, then the translation lengths of $g$ must sum to 0 (by condition (2)). Thus $t_1(g) = 0$ for all $g \in H_1$ and all elements of $H_1$ eventually act trivially (i.e. are in $\text{FSym}(\mathbb{N})$). For $H_2$, our only issue is that the standard generating set, $\{g_2\}$, will generate permutations with any valid translation length but will not include any bijections with finite support. Including the transposition $(1, 1), (2, 1)$ remedies this issue by a conjugation argument similar to that for $H_n$ above. These definitions of $H_1$ and $H_2$ agree with the result for $H_n$ in [Bro87], that (for $n \geq 3$) each $H_n$ is FP$_{n-1}$ but not FP$_n$ i.e. $H_1$ is not finitely generated, $H_2$ is not finitely related, and that for $n \geq 3$, $H_n$ is finitely presented. Since $H_1 \cong \text{FSym}$ and Aut(\text{FSym}) $\cong \text{Sym}$ (see, for example, [DM96] or [Sco87]) we will work with $H_n$ where $n \geq 2$.

**Remark 2.2.** A group is FP$_n$ if and only if a finite index subgroup is FP$_n$. [Bro94 VIII.5.5.1]. This means that all finite extensions of $H_n$ are FP$_{n-1}$ but not FP$_n$.

### 2.2. A reformulation of TCP($H_n$)

We require knowledge of the automorphims of $H_n$ (where $n \geq 2$). From [BCMR14], we have that Aut($H_n$) $\cong S_n \rtimes H_n$ where Inn($H_n$) $\cong H_n$ because $H_n$ is centreless and the action of $S_n$ refers to isometric permutations of the rays. This is because all automorphisms of $H_n$ are obtained by conjugation by an element of $S_n \rtimes H_n$. We shall write elements of $S_n \rtimes H_n$ as $\tau g$ where we first apply the permutation of the branches (denoted by $\tau$) and then apply the permutation that can be considered as in $H_n$ (denoted $g$). We now show that a short computation relates twisted conjugacy for $H_n$ to conjugacy in $S_n \rtimes H_n$.

Starting with the equation for twisted conjugacy, we have

$$(x^{-1})\phi a' x = b' \Rightarrow (\sigma c)^{-1} x^{-1} \sigma c a' x = b' \Rightarrow x^{-1} (\sigma c a') x = \sigma c b'$$

and so twisted conjugacy is equivalent to searching for a conjoagator $x \in H_n$ for $\sigma c a', \sigma c b' \in S_n \rtimes H_n$. We will therefore produce an algorithm to search for an $x \in H_n$ conjugating $\sigma a, \sigma b \in S_n \rtimes H_n$ in order to solve the twisted conjugacy problem for $H_n$.

### 3. Computations in $S_n \rtimes H_n$

#### 3.1. The word problem for $S_n \rtimes H_n$

The generating set in the following lemma will be the one we will use for our problem which is equivalent to TCP($H_n$).

**Lemma 3.1.** For all $n \geq 2$, $S_n \rtimes H_n$ is at most 3 generated.

**Proof.** If $n \geq 3$, use 2 elements to generate all permutations of the rays and then use $g_2$ that was a standard generator for $H_n$. Conjugating $g_2$ by the appropriate permutation of the rays produces the set $\{g_i \mid i = 2, \ldots, n\}$ that can then be used to generate all permutations in $H_n$. For $H_2 \rtimes S_2$ we note that $H_2$ is 2 generated. □

One can then use the solution of the word problem in [ABM13] to obtain a quadratic solution to WP($S_n \rtimes H_n$). We also have

**Lemma 3.2.** Let $\tau g \in S_n \rtimes H_n$. Then we can write explicitly the action of $\tau g$ on $(i, m)$ for all points $(i, m) \in X_n$. 

3.2. The orbits of $S_n \ltimes H_n$. Any element of $S_n \ltimes H_n$ eventually acts in a simple way, i.e. for any $\tau g \in S_n \ltimes H_n$, there exists a $z \in \mathbb{N}$ such that

$$(i, m)\tau g = (i\tau, m + t_{i\tau}(g))$$

for all $m > z$.

Because of this, we will describe an infinite subset of a given infinite orbit. This follows the strategy of [ABM13] for $H_n$. For this the following idea is useful.

**Definition 3.3.** Two sets are *almost equal* if their symmetric difference is finite.

Since $\tau$ acts on all or no points of a branch, we will always consider it as a permutation of $S_n$ acting on the set of branches of $X_n$. Thus we have

**Notation.** For $\tau g \in S_n \ltimes H_n$, we consider $\text{supp}(\tau)$ as a subset of $\{1, \ldots, n\}$.

It will be useful to describe the branches that each disjoint cycle of $\sigma$ acts on.

**Definition 3.4.** Let $\tau g \in S_n \ltimes H_n$. Then a *class* of $\tau g$, denoted $[i]_{\tau}$, is the disjoint cycle of $\tau$ which contains $i$ (where $i \in \{1, \ldots, n\}$). Additionally, we define the *size* of a class $[i]_{\tau}$ to be the cardinality of the set $[i]_{\tau}$, and denote this by $|[i]_{\tau}|$.

These form a partition of the branches of $X_n$. Since our aim will be to conjugate $\sigma a$ to $\sigma b$, we have that the classes of these elements will be the same. Thus the following should not be ambiguous.

**Notation.** We shall often write $[i]$ to mean $[i]_{\sigma}$.

Before defining the sets that will be almost equal to any given orbit of an element of $S_n \ltimes H_n$, we provide three examples, each with the same permutation of the rays (swapping the third and fourth rays).

![Figure 2. Examples of orbits in Aut($H_5$).](image)

Both $\tau a$ and $\tau c$ contain an infinite orbit on the branches on which $\tau$ acts, whilst $\tau b$ contains countably many 2-cycles on these branches. In order to understand why the orbits are finite or infinite, and to motivate the next definition, consider the translation lengths for the branches within the support of $\tau$. We note that

$$t_3(a) + t_4(a) > 0, t_3(b) + t_4(b) = 0 \text{ and } t_3(c) + t_4(c) > 0.$$ 

**Definition 3.5.** For any $\tau g \in S_n \ltimes H_n$, class $[i]_{\tau}$, and $j \in [i]_{\tau} = \{i_1, i_2, \ldots, i_q\}$, let

$$t_j(\tau g) := \sum_{k=1}^{q} t_{i_k}(g)$$
Clearly one has that \( t_i(\tau g) = t_i(\tau g) \) when \( i_k \) and \( i_k' \) belong to the same class. If \( t_j(\tau g) = 0 \), then we will have (for a suitably large \( m_1 \) and any \( i_k \in [i] \)) that \((i_k, m_1)(\tau g)[i] = (i_k, m_1)\). In fact, \( t_j(\tau g) \) measures the distance travelled along a branch after applying our element \([i] \) times. Thus, as \( t_j(g) \) did for \( H_n \), the constants \( t_j(\tau g) \) define the number of distinct infinite orbits on the branches \([i] \).

This can be seen in Figure (2) by the fact that \( \sigma a \) and \( \sigma c \) both have two infinite orbits on the third and fourth branches despite the fact that \( t_3(a) \neq t_3(c) \) and \( t_4(a) \neq t_4(c) \). This is because \( t_3(\sigma a) = t_3(\sigma c) \).

Remark 3.6. For \( H_n \), the map \( g \mapsto t(g) \) was a homomorphism. The corresponding map \( i_k(\tau g) \) for \( \tau g \in S_n \times H_n \) has been defined to record the number of infinite orbits found ‘far out’ on a branch \( i \) but has not been defined to be a homomorphism: composition of elements of \( S_n \times H_n \) with respect to the map have not been defined.

We will now produce a family of sets \( X_{i_1,d}(g) \) so that for any given infinite orbit (of an element of \( S_n \times H_n \)), there exist classes \([i], [j] \) and constants \( d_1, d_2 \in \mathbb{N} \) so that \( X_{i_1,d_1}(g) \cup X_{i_1,d_2} \cup X_{i_2,d_3} \) is almost equal to that infinite orbit. The following holds for \( t_i(\tau g) \geq 0 \), but by swapping \( \tau g \) for \( \tau g^{-1} \), similar arguments hold for \( t_i(\tau g) \leq 0 \).

We consider where a point \((i_1, m_1)\) (for \( m_1 \) suitably large) may be sent within the branches \([i_1] \) by a given \( \tau g \in S_n \times H_n \). Notice that \([i_1] \) is the smallest natural number \( k \) such that \((i_1, m_1)(\tau g)^k \) lies on branch \( i_1 \). Moreover, as \( k \) ranges from 1 to \([|i_1]| \), we will have that \((i_1, m_1)(\tau g)^k \) visits one point from each of the branches of \([i_1]\). Thus if \( t_{i_1}(\tau g) = 0 \), we have countably many cycles of length \([|i_1]| \). Otherwise, the set \( \{(i_1, m_1)(\tau g)^k \mid k \equiv 0 \mod |t_i(\tau g)| \text{ and } k \in \mathbb{N} \} \) will be almost equal to \( X_{i_1,d}(g) \) for some \( 0 \leq d < |t_{i_1}(\tau g)| \) where \( d \equiv m_1 \mod |t_i(\tau g)| \). Continuing in this way for each \( i_1 \in [i_1] \) gives a description of a set almost equal to \( \{(i_1, m_1)(\tau g)^k \mid k \in \mathbb{N} \} \).

Definition 3.7. Let \( \tau g \in S_n \times H_n \), \([i] \tau = \{i_1, \ldots, i_q\} \) be a class of \( \tau \), and \( i \in I \).

Then \( X_{i_1,m_1}(g) \cup X_{i_2,m_1+t_2(g)}(g) \cup \cdots \cup X_{i_q,m_1+t_q(g)}(g) \cup X_{\tau g}(g) \cup \cdots \cup X_{\tau g}(g) \) is almost equal to an infinite orbit of \( \tau g \) and will be denoted by \( X_{[i],d}(g) \).

Thus for any infinite orbit of an element \( \tau g \in S_n \times H_n \), we may choose classes \([i] \) and \([j] \) and numbers \( d_1, d_2 \in \mathbb{N} \) such that \( X_{[i],d}(g) \) is a set almost equal to that orbit.

\[(5) \quad X_{[i],d_1}(g) \cup X_{[j],d_2}(g) \]

Remark 3.8. If \( \tau \) acts trivially (so that \( \tau g \) can be considered as in \( H_n \)), we have that \( X_{[i],d}(g) = X_{i_1,m_1}(g), t_i(\tau g) = t_i(g) \) and these sets are those seen in [ABM13].

3.3. Identities arising from the equation for conjugacy. Our aim is to produce an algorithm to search for an \( x \in H_n \) that conjugates \( \sigma a, \sigma b \in S_n \times H_n \). The fact that these elements have the same permutation of the rays \( \sigma \) follows from Section 2.2. In this section we will manipulate the equation we have for conjugacy to produce conditions that a conjugator \( x \in H_n \) must satisfy. We will use these conditions in Section 4.3.

First, some simple computations to rewrite \( t_i(\sigma^{-1}x^{-1}\sigma) \) are needed. Consider a class of rays \([i] = \{i_1, \ldots, i_q\} \) where \( i_q \sigma = i_1 \) and \( i_1 \sigma = i_{1+1} \) for all \( 1 \leq s < q \). We
may then compute \( t_i(\sigma^{-1}x\sigma) \) by considering the image of \((i_1, m_1)\) under \(\sigma^{-1}x\sigma\) for a suitably large number \(m_1\).

\[
(i_1, m_1)\sigma^{-1}x\sigma = (i_q, m_1)x\sigma = (i_q, m_1 + t_q(x))\sigma = (i_1, m_1 + t_q(x))
\]

Similarly, for \(1 < s \leq q\),

\[
(i_s, m_1)\sigma^{-1}x\sigma = (i_{s-1}, m_1)x\sigma = (i_{s-1}, m_1 + t_{i_{s-1}}(x))\sigma = (i_s, m_1 + t_{i_{s-1}}(x))
\]

and so we have \( t_i(x^\sigma) = t_{i\sigma^{-1}}(x) \) for any class \([i]_\sigma\).

**Lemma 3.9.** For any permutation of the branches \(\sigma\) and any \(x \in H_n\), we have \( t_i(\sigma^{-1}x^{-1}\sigma) = -t_{i\sigma^{-1}}(x) \) for all \(i \in \{1, \ldots, n\} \).

**Proof.** We note that through simple computation and the calculations above,

\[
t_i(\sigma^{-1}x^{-1}\sigma) = t_i((\sigma^{-1}x\sigma)^{-1}) = -t_i(x^\sigma) = -t_{i\sigma^{-1}}(x)
\]

\[\square\]

**Lemma 3.10.** Let \(sa, sb \in S_n \rtimes H_n\). For there to exist a conjugator \(x \in H_n\) for these elements, a necessary condition for all \([i]_\sigma\) classes is that \( t_i(sa) = t_i(sb) \).

**Proof.** Assume there exists a conjugator \(x \in H_n\). Also let \([i] = \{i_1, i_2, \ldots, i_q\}\) be any class of \(\sigma\). Then, from the equation for conjugacy,

\[
(\sigma^{-1}x^{-1}\sigma)ax = b
\]

\[
\Rightarrow t_i(\sigma^{-1}x^{-1}\sigma) + t_i(a) + t_i(x) = t_i(b)
\]

\[
\Rightarrow -t_{i\sigma^{-1}}(x) + t_i(a) + t_i(x) = t_i(b)
\]

\[
\Rightarrow -\sum \ t_i(x) + \sum t_i(a) + \sum t_i(x) = \sum t_i(b)
\]

\[
\Rightarrow \sum \ t_i(a) = \sum t_i(b)
\]

\[\square\]

It is intuitively clear that choosing \(t_i(x)\) for one \(i \in [i]\) will determine \(t_j(x)\) for all other \(j \in [i]\). A formal proof follows naturally from Lemma 3.10.

**Lemma 3.11.** Let \(x \in H_n\) conjugate some \(sa, sb \in S_n \rtimes H_n\). Then, for each \([i] = \{i_1, \ldots, i_q\}\), there is an explicit formula to define \(t_i(x)\) for \(1 \leq s \leq q\) that depends on only one \(i_k \in [i]\). Moreover, any conjugator in \(H_n\) for \(sa, sb\) satisfies this formula.

**Proof.** From (6) in the proof of Lemma 3.10

\[
t_i(x) - t_{i\sigma^{-1}}(x) = t_i(b) - t_i(a).
\]

Hence for all \(2 \leq s \leq q\) we have

\[
t_i(x) - t_{i_s}(x) = t_i(b) - t_i(a) \quad \text{and} \quad t_i(x) - t_{i_{s-1}}(x) = t_i(b) - t_i(a)
\]

\[
\Rightarrow t_i(x) - t_{i_{s-1}}(x) = (t_i(b) - t_i(a)) + (t_q(b) - t_q(a)) \quad \text{by setting} \ s = q
\]

\[
\Rightarrow t_i(x) - t_{i_s}(x) = (t_i(b) - t_i(a)) + \sum_{j=s+1}^{q} (t_j(b) - t_j(a)) \quad \text{for} \ 2 \leq s < q
\]
Therefore, 
\[
    t_i(x) = \begin{cases} 
        t_{i_1}(x) + (t_{i_1}(a) - t_{i_1}(b)) & \text{if } s = q \\
        t_{i_1}(x) + (t_{i_1}(a) - t_{i_1}(b)) + \sum_{j=s+1}^{q} (t_{i_j}(a) - t_{i_j}(b)) & \text{if } 2 \leq s < q 
    \end{cases}
\]

Note that the choice of \(i_1\) was arbitrary. Since we have placed no conditions on the conjugator, the final claim holds. \(\square\)

**Remark 3.12.** We will focus on one \(i_1 \in [i_1]_\sigma\) for each class \([i]_\sigma\). Since the choice of \(i_1\) is arbitrary, let it be the smallest ray of the class, i.e. \(i_1 := \inf [i_1]_\sigma\) for each class of \(\sigma\).

The following lemma explicitly computes the sum of the translation lengths in one class for a conjugator in \(H_n\). This will be used to prove Proposition 4.12.

**Lemma 3.13.** Let \(\sigma a, \sigma b \in S_n \ltimes H_n\). If they are conjugate by \(x \in H_n\), then for any class of rays \([i]\) and any choice of \(i_1 \in [i]\) we have that \(\sum_{i \in [i]} t_i(x) = ||[i]|| t_{i_1}(x) + A_{i_1}\).

Moreover, \(A_{i_1}\) is a computable constant (dependent on the given \(\sigma a, \sigma b\) and on the chosen \(i_1\)).

**Proof.** Using the formula from Lemma 3.11 to rewrite \(t_i(x)\) for \(2 \leq s \leq q\),

\[
    (7) \quad \sum_{j=2}^{q-1} t_{i_j}(x) = \sum_{s=2}^{q-1} \left[ t_{i_1}(x) + (t_{i_1}(a) - t_{i_1}(b)) + \sum_{j=s+1}^{q} (t_{i_j}(a) - t_{i_j}(b)) \right]
    \]

\[
    = (q-2)[t_{i_1}(x) + (t_{i_1}(a) - t_{i_1}(b))] + \sum_{s=2}^{q-1} \sum_{j=s+1}^{q} [t_{i_j}(a) - t_{i_j}(b)]
    \]

Now,

\[
    q \sum_{s=2}^{q-1} \sum_{j=s+1}^{q} [(t_{i_j}(a) - t_{i_j}(b))] = (q-2)(t_{i_q}(a) - t_{i_q}(b)) + (q-3)(t_{i_{q-1}}(a) - t_{i_{q-1}}(b)) + \ldots
    \]

\[
    \ldots + 2(t_{i_k}(a) - t_{i_k}(b)) + (t_{i_1}(a) - t_{i_1}(b))
    \]

\[
    = \sum_{k=3}^{q} (k-2)(t_{i_k}(a) - t_{i_k}(b))
    \]

and so including \(t_i(x)\) and \(t_{i_1}(x)\) in the sum labelled (7)

\[
    \sum_{i \in [i]} t_i(x) = t_{i_1}(x) + (q-1)(t_{i_1}(x) + (t_{i_1}(a) - t_{i_1}(b)) + \sum_{k=3}^{q} (k-2)(t_{i_k}(a) - t_{i_k}(b))
    \]

\[
    = q t_{i_1}(x) + (q-1)(t_{i_1}(a) - t_{i_1}(b)) + \sum_{k=3}^{q} (k-2)(t_{i_k}(a) - t_{i_k}(b))
    \]

where \(q = ||[i]||\) and \(t_{i_1}(x)\) is the only term depending on \(x\). \(\square\)

**Definition 3.14.** In light of the lemma above,

\[
    A_{i_1} := A_{i_1}(a, b, [i_1]_\sigma) = (q-1)(t_{i_1}(a) - t_{i_1}(b)) + \sum_{s=3}^{q} (s-2)(t_{i_s}(a) - t_{i_s}(b))
    \]
3.4. Results regarding centralisers in $\text{Aut}(H_n)$. We briefly explain some results relating to $C_{H_n}(\tau g) = \{ x \in H_n \mid \tau gx = x\tau g \}$ where $\tau g \in S_n \ltimes H_n$. We do this because of the following lemma.

**Lemma 3.15.** Let $x \in G$ be a conjugator of $a$ and $b$ in $G$. Then, $x' \in G$ is also a conjugator of $a$ and $b$ if, and only if, $x' = xc$ for some $c \in C_G(a)$.

Centralisers for $H_n$ are described in [SAG12], and so we will describe $C_{H_n}(\sigma a)$, where $\sigma a \in S_n \ltimes H_n$. Our aim will be to describe only the translation lengths of the elements in the centraliser. Notice that the elements of $C_{H_n}(\sigma a)$ conjugate $\sigma a$ to $\sigma a$, and so the equations derived in Section 3 hold, meaning that $t_k(x) = t_j(x)$ whenever $k \in \{ j \}$.

**Definition 3.16.** Given $\tau g \in S_n \ltimes H_n$, we define an equivalence relation $\sim_g$ on $\{ [\ell] \mid \ell = 1, \ldots, n \}$, as the one generated by setting $[i] \sim_g [j]$ if and only if there is an orbit of $\tau g$ almost equal to $X_{[i], d_i}(g) \cup X_{[j], d_j}(g)$ for some $d_i, d_j \in \mathbb{N}$.

**Proposition 3.17.** Should an element of $C_{H_n}(\sigma a)$ act on almost all of a branch $i \in I$, then it must act on almost all of every branch $j$ such that $[j] \sim_a [i]$.

**Proof.** Using the equation

$$g\sigma a = \sigma ag$$

satisfied by any $g \in C_{H_n}(\sigma a)$, we have that how $g$ acts on one point of an orbit defines how it acts on the entire orbit. Thus setting $t_i(g) = \delta$ for some $\delta \in \mathbb{N}$ and $i$ such that $t_i(\sigma a) < 0$ defines how $g$ acts on the $|t_i(\sigma a)|$ infinite orbits of the branch $i$. The branches $k$ that these orbits have infinite support on satisfy $|k| \sim_a |i|$ and therefore define $t_k(g)$ for these branches. Continuing in this way defines $t_j(g)$ for all $j$ such that $[j] \sim_a [i]$ or produces a contradiction either by producing a $g \not\in H_n$ or by setting two distinct points to have the same image or preimage under $g$. Thus we will have either produced an element $g \in C_{H_n}(\sigma a)$, or shown that no element of the centraliser satisfies $t_i(g) = \delta$. Note that we may choose any $i \in I$ since we may run the algorithm for $\sigma a$ or $(\sigma a)^{-1}$ as $C_{H_n}(\sigma a) = C_{H_n}((\sigma a)^{-1})$. We may also pick points from $S = \{ (i, m) \mid i \in \{ 1, \ldots, n \} \}$ and $l(\sigma a) \leq m < l(\sigma a) + |t(\sigma a)|$ to define the action of $g$ suitably far out since defining $g'$ to act on points further out will imply that $g'$ acts on $S$ in the same way as $g$. \hfill $\square$

We may therefore compute elements of the centraliser which act only on branches in $I$. Also, if $h \in C_{H_n}(\sigma a)$ acts on almost all of a branch $i \in I$, then it must act on almost all of every branch $j$ such that $[j] \sim_a [i]$. If $g, g' \in C_{H_n}(\sigma a)$ act with non-zero translation length only on the branches $\{ j \mid [j] \sim_a [i] \}$, then we have that $\text{supp}(g)$ is almost equal to $\text{supp}(g')$. Furthermore, if $g' \in C_{H_n}(\sigma a)$ acts by a different, non-zero translation length to $g \in C_{H_n}(\sigma a)$, then we have, for $t_k(g), t_k(g') \neq 0$, that

$$\frac{t_j(g)}{t_k(g)} = \frac{t_j(g')}{t_k(g')}$$

implying that all translation lengths of centralisers acting only on $\{ j \mid [j] \sim_a [i] \}$ can be generated by one element.

We now use an adaptation of the element defined in [ABM13] Lem. 4.6.

**Definition 3.18.** Let $\tau g \in S_n \ltimes H_n$ and let $g_1 g_2 \ldots g_m$ be an expression of $\tau g$ as a (possibly infinite) product of disjoint cycles. Choose an $i \in I$ and compute $\{ j \mid [j] \sim_r [i] \}$. We then define $g_{[i]}$ as the product of all cycles among the $g_1 g_2 \ldots g_m$. 

which are infinite and whose support is almost equal to \( X_{[j],d}(g) \cup X_{[j']},s(g) \) for any \([j],[j']\) related to \([i]\).

Let \( \omega \in \mathbb{N} \) be the order of \( \sigma \), the isometric permutation of the rays for the elements \( \sigma a \) and \( \sigma b \) which we wish to conjugate.

**Lemma 3.19.** Let \( \sigma a, \sigma b \in S_n \ltimes H_n \) be the elements we wish to conjugate. If \( i \in I \), then \((a_{[i]})^{\omega}\) is in \( H_n \) and is in the centraliser of \( \sigma a \).

**Proof.** We will show that \( a_{[i]} \in S_n \ltimes H_n \) and commutes with \( \sigma a \) since then \((a_{[i]})^{k}\) is in the centraliser of \( S_n \ltimes H_n \) for all \( k \in \mathbb{N} \). This will prove the claim since then \((a_{[i]})^{\omega}\) is in \( H_n \) (by our definition of \( \omega \)).

By considering \( a_{[i]} \) and \( \sigma a \) in \( \text{Sym}(X_n) \), the commutativity follows since we are choosing disjoint cycles from \( \sigma a \). For any branch \( k \in I^c \), we have that \( g_{[i]} \) acts trivially on almost all of the branch \( k \). For branches \( j \) such that \([j] \sim_{a} [i] \), we have that \( X_{[j],d_1} \cup X_{[j'],d_2} \) is almost equal to an orbit of \( \sigma a \) for suitably chosen \( d_1, d_2 \in \mathbb{N} \) and a class \([j'] \sim_{a} [i] \). We also have that any infinite orbit of \( \sigma a \) containing a set almost equal to \( X_{[j],d} \) (for any \( d \in \mathbb{N} \)) will be an orbit of \( a_{[i]} \). Continuing in this way, if \( a_{[i]} \) has an orbit almost equal to \( X_{[i'],d_1} \cup X_{[j'],d_2} \) for any \( i', j' \in \{1, \ldots, n\} \) and some \( d_1', d_2' \in \mathbb{N} \), then it has orbits containing sets almost equal to \( X_{[i'],d} \) and \( X_{[j'],d} \) for every \( d \in \mathbb{N} \). Thus \( a_{[i]} \) is in \( S_n \ltimes H_n \) and so \((a_{[i]})^{\omega}\) is in \( H_n \).

Since \((a_{[i]})^{\omega}\) is an element acting on almost all of the branches \([j] | [j] \sim_{a} [i]\) and no other branches, checking

\[ t_i(g) = 1, \ldots, |\omega \cdot t_i(\sigma a)| \]

will compute the required generator (since \( t_i(a_{[i]}^{\omega}) \leq \omega \cdot t_i(a_{[i]}) \) = \( \omega \cdot t_i(\sigma a) \) is computable and finite).

**Definition 3.20.** For any \( i \in I \), let \( c_i \in C_{H_n}(\sigma a) \) denote the generator produced by the above algorithm. Note that if \([j] \sim_{a} [i]\) then \( c_j = c_i \).

**Remark 3.21.** The author believes that the generator will always be the root of \((a_{[i]})^{\omega}\). We may think of the root as being an element with translation lengths given by \( t((a_{[i]})^{\omega})/v \), where \( v \) is the greatest common divisor of the entries of \( t((a_{[i]})^{\omega}) \).

**Definition 3.22.** Let \( \tau g \) be an element of \( S_n \ltimes H_n \) written in disjoint cycle notation and fix an \( r \in \mathbb{N} \). Then \((\tau g)_{|r}\) consists of all of the \( r \)-cycles of \( \tau g \).

Notice that for any \( \tau g \in S_n \ltimes H_n \) and any \( r \in \mathbb{N} \), we have that \((\tau g)_{|r}\) is in \( S_n \ltimes H_n \) and that, given \( \tau g \), these are computable for every \( r \in \mathbb{N} \) (by Lemma 3.2). We now deal with the branches \( j \in I^c \). These branches consist almost entirely of \([j]\) cycles. The \( r \)-cycles can be computed for any \( \sigma a \in N_{\text{Sym}}(H_n) \) and \((\sigma a)_{|r}\) describes such an element. Now, we may record numbers \( r_j \) such that there exists a \( j \) with \( t_j(\sigma a) = 0 \) and \([j] = r_j \). Since an element of the centraliser must send \( r \)-cycles to \( r \)-cycles, we have that if there is only one class of branches \([j]\) with \( t_j(\sigma a) = 0 \) and \([j]\) = \( r \), then the centraliser acts as \( \text{FSym} \) on almost all of this branch. Moreover the centraliser acts as \( H_s \) with \( s \) being the number of classes of size \( r_j \) for some \( j \in I^c \). This generalises the \( H_n \) case where the classes were all of size 1, and means that the translation lengths for these branches is either 0 or any vector in \((H_s)_{\pi} \) i.e. any vector in \( \mathbb{Z}^{s-1} \) (from the condition that the components sum to 0).
Remark 3.23. Computing $C_{S_n \rtimes H_n}(\sigma a)$ would require additional work, since for example, if $t_i(a) = t_j(a)$, then we the element of $S_n \rtimes H_n$ which swaps almost all of these branches may be in the centraliser.

4. Conjugacy in $S_n \rtimes H_n$

In this section we will construct an algorithm which, given $\sigma a, \sigma b \in S_n \rtimes H_n$, either outputs an $x \in H_n$ such that $(\sigma a)^x = (\sigma b)$, or halts without outputting such an $x$ if one does not exist.

From Section 3.2 those rays where $t_i(\tau g) = 0$ produce orbits of a different structure to branches where $t_i(\tau g) \neq 0$. Since these will require different arguments, we have the following.

Definition 4.1. Let $\tau g \in S_n \rtimes H_n$. Then $I(\tau g) := \{ i \mid t_i(\tau g) \neq 0 \}$.

Note. We shall often write $I$ to mean $I(\sigma a)$. Note that $I(\sigma a) = I(\sigma b)$ if $\sigma a \sim_x \sigma b$ for some $x \in H_n$ (by Lemma 3.10). Also let $I^c := \{ 1, \ldots, n \} \setminus I$.

We will be working with many results that require an arbitrary choice of $i_1 \in [i]_\sigma$. It will be satisfactory to choose each $i_1$ to be infinitum of the set $[i]_\sigma$. Unless stated, any choice of $i_1$ by the algorithm is made in this manner. An example of such a choice can be found in Lemma 3.13.

4.1. An algorithm for finding a conjugator in $FSym$. Many of the arguments and ideas of this section draw their ideas from [ABM13, Section 3].

By definition, any conjugator of $\sigma a, \sigma b \in S_n \rtimes H_n$ will send the support of $\sigma a$ to the support of $\sigma b$. If we wish to find a conjugator in $FSym$, this means that the symmetric difference of these sets must be finite whilst $\text{supp}(\sigma a) \cap \text{supp}(\sigma b) =: N$ can be infinite. Our method will be to produce a partial bijection from $\text{supp}(\sigma a) \setminus N$ to $\text{supp}(\sigma b) \setminus N$. Our first lemma shows that any such partial bijection can be extended to a bijection on $(\text{supp}(\sigma a) \setminus N) \cup (\text{supp}(\sigma b) \setminus N)$. By producing a finite set in which the support of our conjugator is contained, our algorithm will simply try all possible bijections on this set, either finding a partial bijection from $\text{supp}(\sigma a)$ to $\text{supp}(\sigma b)$ or conclude that no such partial bijection exists. Such an algorithm is therefore sufficient to find a conjugator in $FSym$. Notice that we may not always be able to produce a bound for any conjugator in $FSym$ since, for example, it may act as any permutation on the fixed points of $\sigma a$ which can be a countably infinite set. Thus we produce a set $Z'$ such that, if a conjugator $x \in FSym$ exists, then there is another conjugator $x' \in FSym$ with $\text{supp}(x') \subseteq Z'$.

Lemma 4.2. Let $\sigma a, \sigma b \in S_n \rtimes H_n$. If $\sigma a \sim_x \sigma b$ by some $x \in FSym$, then

\begin{equation}
|\text{supp}(\sigma a) \setminus N| = |\text{supp}(\sigma b) \setminus N| \text{ where } N = \text{supp}(\sigma a) \cap \text{supp}(\sigma b)
\end{equation}

Proof. Note that if $\text{supp}(x) \subseteq A$, then $x$ restricts to a bijection on $A$. Clearly if $\text{supp}(\sigma a) = \text{supp}(\sigma b)$, then the result holds. If they are not equal then $\text{supp}(x) \neq \emptyset$ and $x$ restricts as a bijection from $\text{supp}(\sigma a)$ to $\text{supp}(\sigma b)$. Thus $x$ restricts to a bijection on $\text{supp}(\sigma a) \cup \text{supp}(\sigma b) \cup \text{supp}(x)$.

Let $B_{\sigma a} := \text{supp}(\sigma a) \setminus N$ and $B_{\sigma b} := \text{supp}(\sigma b) \setminus N$ where $N$ is as above. Additionally, let $B_x := \text{supp}(x) \setminus (B_{\sigma a} \cup B_{\sigma b})$. These sets are coloured in on the venn diagram below.
Now, as $\sigma a$ and $\sigma b$ are conjugate by an $x \in \text{FSym}$, $|\text{supp}(x)| < \infty$ and $\text{supp}(\sigma a)$ is almost equal to $\text{supp}(\sigma b)$, making both $B_{\sigma a}$ and $B_{\sigma b}$ finite. Thus $x$ sends $B_{\sigma a} \sqcup B_x$ bijectively to $B_{\sigma b} \sqcup B_x$ meaning $|B_{\sigma a} \sqcup B_x| = |B_{\sigma b} \sqcup B_x|$ and so $|B_{\sigma a}| = |B_{\sigma b}|$. □

Our work must now be to produce a set which bounds $\text{supp}(x')$ for a conjugator $x' \in \text{FSym}$. The following necessary condition is useful for this. Let $x$ be our conjugator. Since $x \in \text{FSym}$ we that $t_i(x) = 0$ for all $i$, and so

$$(\sigma^{-1}x\sigma)ax = b$$

$\Rightarrow t_i(\sigma^{-1}x^{-1}\sigma) + t_i(a) + t_i(x) = t_i(b)$$

$\Rightarrow t_i(x^{-1}) + t_i(a) + t_i(x) = t_i(b)$$

which implies that $t_i(a) = t_i(b)$ and so $t_i(\sigma a) = t_i(\sigma b)$ for all $i \in \{1, \ldots, n\}$. From Lemma 3.2 there exists a computable number $z \in \mathbb{N}$ (dependent only on $\sigma a$ and $\sigma b$) such that (9) holds. This will be the ‘eventual’ action of any element of $S_n \ltimes H_n$ (so that almost all points of $X_n$ are acted on in this way).

$$(i, m)\sigma a = (i\sigma, m + t_i(\sigma a)) \text{ for all } i \in \{1, \ldots, n\} \text{ and all } m > z$$

(9)

$$(i, m)\sigma b = (i\sigma, m + t_i(\sigma b)) \text{ for all } i \in \{1, \ldots, n\} \text{ and all } m > z$$

Recall that, for any $\tau g \in S_n \ltimes H_n$, the element $(\tau g)|_r$ acts as $\tau g$ on all orbits of length $r$ and trivially elsewhere.

**Lemma 4.3.** Let $\sigma a, \sigma b \in S_n \ltimes H_n$ and fix an $r \in \mathbb{N}$. If $(\sigma a)|_r \sim_x (\sigma b)|_r$ by some $x \in \text{FSym}$, then there is an $x' \in \text{FSym}$ which conjugates $(\sigma a)|_r$ to $(\sigma b)|_r$ with $\text{supp}(x') \subseteq \{(i, m) \mid i = 1, \ldots, n, m \leq z\} =: Z$, where $z$ is the number seen in (9).

**Proof.** Note that the action of a conjugator $x$ on a point $(i, m)$ defines the action of $x$ on all points within the orbit $\{(i, m)(\sigma a)^k \mid k \in \mathbb{Z}\}$. Thus to simplify our computations, we will only describe how a conjugator acts on one point of a given orbit. This can be done since an orbit either lies entirely in $Z$ or $X_n \setminus Z$.

Let $x$ be as stated above (some conjugator of $(\sigma a)|_r$ and $(\sigma b)|_r$). We will aim to produce the conjugator $x'$. First, if $x$ acts bijectively on $X_n \setminus Z$ with some permutation $\delta$ then, since $\sigma a$ and $\sigma b$ agree on these points, another conjugator will be given by $x(\delta^{-1})$. It may also be that $x$ sends $(i, m) \in Z$ to some $(i', m') \in X_n \setminus Z$. 
Since $|x| < \infty$ (as $x \in \text{FSym}$) we have that $(i, m)x^p$ lies in $Z$ (for some minimal $p \in \mathbb{N}$). Since $\{(i, m)x^k \mid k \in \mathbb{N}\}$ are all points lying in the support of some $r$-cycle of $\sigma a$, we may define $x'$ to send $(i, m)$ to $(i, m)x^p$.

Next we show that the set $\{(i, m) \mid i = 1, \ldots, n \text{ and } m \leq z + |t_i(\sigma a)|\}$ is a suitable candidate for $Z'$, a set which, should a conjugator exists, includes $\text{supp}(x)$ for some conjugator $x \in \text{FSym}$.

**Lemma 4.4.** Let $\sigma a, \sigma b \in S_n \ltimes H_n$. If these are conjugate by an element of $\text{FSym}$, then there exists a conjugator $x \in \text{FSym}$ such that

$$\text{supp}(x) \subseteq \{(i, m) \in X_n \mid i \in \{1, \ldots, n\} \text{ and } m \leq z + |t_i(\sigma a)|\} =: Z'.$$

**Proof.** First we deal with the rays $j \in I^c$. From Section 3.2 these rays contain countably many $|[j]|$ cycles. If $\sigma a$ and $\sigma b$ do not agree on almost all of these, $x$ cannot conjugate $\sigma a$ and $\sigma b$ since $\text{supp}(x)$ is finite. By applying the method of the proof above, we may assume that our conjugator acts trivially on all points $(j, m)$ where $j \in I^c$ and $m > z$.

Now we focus on the branches $i \in I$. The number $z$ in (9) also gives us that $(i, m)(\sigma a)^{[i]} = (i, m + t_i(\sigma a))$ for all $i \in I$ and all $m \geq z$. The following argument holds for $t_i(\sigma a) > 0$, but a similar argument (replacing $a, b$ with their inverses) holds for when $t_i(\sigma a) < 0$. Since $x \in \text{FSym}$, there exists a $z'$, which without loss of generality may be assumed to be minimal, such that

$$(i, m)x = (i, m) \text{ for all } m \geq z'.$$

We shall now bound $z'$ by $z + t_i(\sigma a)$. We argue by contradiction and suppose that $z' > z + t_i(\sigma a)$. For all $m \geq 0$ we have

$$(i, z' + m + 1)(x^{-1}\sigma ax)^{−[i]} = (i, z' + m)(\sigma b)^{−[i]} = (i, z' + m - t_i(\sigma b)) = (i, z' + m - t_i(\sigma a))$$

since $z' + m \geq z' > z$. Similarly,

$$(i, z' + m + 1)(x^{-1}\sigma ax)^{−[i]} = (i, z' + m + 1)(x^{-1}(\sigma a)^{−[i]}x$$

$$= (i, z' + m)(\sigma a)^{−[i]}x = (i, z' + m - t_i(\sigma a))x$$

which contradicts the minimality of $z'$ in (10). □

**Proposition 4.5.** Given $\sigma a, \sigma b \in S_n \ltimes H_n$, there is an algorithm that decides (positively or negatively) whether there exists a conjugator $x \in \text{FSym}$ for these elements and produces such an $x$ if it exists.

**Proof.** From the previous lemma, if there is a conjugator in $\text{FSym}$ then we will find a conjugator with support contained in $Z' := \{(i, m) \in X_n \mid i \in \{1, \ldots, n\} \text{ and } m \leq z + |t_i(\sigma a)|\}$, i.e. we may let our conjugator be the identity for all $(i, m) \in X_n \setminus Z'$. Also, the statement from Lemma 4.2 holds, and so (in the terminology of this lemma) there is a bijection between the sets $\text{supp}(\sigma a) \setminus N$ and $\text{supp}(\sigma a) \setminus N$ which can be extended as the identity to $\text{supp}(\sigma a) \cup \text{supp}(\sigma b) = \text{supp}(\sigma a) \cup (\text{supp}(\sigma b) \setminus N)$. Clearly we may then extend this to be the identity on the rest of $X_n$.

We are now ready to produce a conjugator $x \in \text{FSym}$. First, compute $z$ and the set $Z'$. Second, compute all partial maps from $\text{supp}(\sigma a) \setminus N$ to $\text{supp}(\sigma b) \setminus N$ and check whether each of these satisfy the (finite number of) equations given by the equation for conjugacy (since $\text{supp}(x)$ is finite). If such a partial map is not found we have that $\sigma a$ and $\sigma b$ are not conjugate by any element of $\text{FSym}$. If we find such
form a partition of \( \mathbf{X}_n \) by checking if \( x \) decides (positively or negatively) whether there exists a conjugator \( \sigma \). Proposition 4.7.\)

Now, if there is an algorithm that decides whether any two given elements \( g, h \) of \( G \) are conjugate by an element of \( C_0 \) and finds this conjugator, then there is an algorithm that decides whether any two given elements \( g, h \) of \( G \) are conjugate by an element of \( C \) and finds this conjugator.

Condition (11) will be critical to our arguments. Let \( H_n^* \) denote the group consisting of all \( x \) satisfying this condition. Also, \( \omega := |\sigma| \) (the order of the permutation of the rays we denoted by \( \sigma \)), and is included since it is necessary for Lemma 4.13.

\[
x \in H_n \text{ and } t_i(x) \equiv 0 \mod |\omega \cdot t_i(\sigma a)| \text{ for all } i \in I.
\]

Proposition 4.7. Assume there exists an algorithm, which given \( \sigma a, \sigma b \in S_n \ltimes H_n \), decides (positively or negatively) whether there exists a conjugator \( x^* \in H_n^* \). Then there exists an algorithm, which given \( \sigma a, \sigma b \in S_n \ltimes H_n \), decides (positively or negatively) whether there exists an \( x \in H_n \) such that \( x^{-1} \sigma a x = \sigma b \).

Proof. Our aim will be to show that the conditions of Lemma 4.6 are satisfied. In the terminology of this lemma we have that \( C \) is the underlying set of \( H_n \), i.e. those \( x \in H_n \) whose translation lengths sum to 0. We set

\[
C_\omega = \{ x \in H_n \mid t_i(x) \equiv 0 \mod |\omega \cdot t_i(\sigma a)| \text{ for all } i \in I \},
\]

\[
C_\omega = \{ x \in H_n \mid t_i(x) \equiv r_i \mod |\omega \cdot t_i(\sigma a)| \text{ for all } i \in I \}
\]

where \( r_i = (r_1, \ldots, r_n) \) with \( 0 \leq r_i < |\omega \cdot t_i(\sigma a)| \) for all \( i \in I \) and \( r_i = 0 \) otherwise. Thus the \( C_\omega \) form a partition of \( C \). One can decide whether or not a given element \( g \in H_n \) belongs to a given \( C_\omega \) by checking if \( t_i(g) \equiv r_i \mod |\omega \cdot t_i(\sigma a)| \) for all \( i \in I \). Moreover, given a tuple \( s = (s_1, \ldots, s_n) \in \mathbb{Z}^n \) with \( \sum s_i = 0 \), we can easily construct an \( x_s \in H_n \) such that \( t(x_s) = s \) and have that \( C_\omega x_s^{-1} = C_\omega \). We have therefore satisfied all of the required assumptions to apply Lemma 4.6.\)

Thus our problem is now to find an algorithm, which given \( \sigma a, \sigma b \in S_n \ltimes H_n \), decides (positively or negatively) whether there is an \( x \), satisfying condition (11), which conjugates \( \sigma a \) to \( \sigma b \). In order to have a finite number of problems to solve so that we may apply Lemma 4.6, we want to bound the translation lengths of our conjugator \( x \in H_n^* \).

Definition 4.8. An absolute bound \( M := M(\sigma a, \sigma b) \) is a computable number \( M \) (dependent only on \( \sigma a \) and \( \sigma b \)) such that there exists a conjugator \( x \in H_n \) with

\[
\sum_{i=1}^n |t_i(x)| < M.
\]

We will now use Lemma 4.6 and the absolute bound \( M \) to show that the algorithm for finding a conjugator in \( \text{FSym} \) is sufficient to find a conjugator \( x \in H_n \).
Proposition 4.9. If there exists an algorithm that finds a conjugator in FSym for \( \sigma a, \sigma b \in S_n \rtimes H_n \) and an absolute bound \( M(\sigma a, \sigma b) \) for a conjugator \( x \in H_n^* \), then there exists an algorithm that decides (positively or negatively) whether there exists a conjugator in \( H_n^* \).

Proof. We will show that the assumptions of Lemma 4.6 are satisfied. From the absolute bound \( M \), we have that if \( \sigma a, \sigma b \) are conjugate by \( x^* \in H_n^* \), then there is a conjugator such that \( \sum |t_i(x)| < M \) where \( i \in \{1, \ldots, n\} \). This means our algorithm need only search for conjugators satisfying condition (11) and with translation lengths up to this bound. Since we have an algorithm for conjugacy in FSym (our first assumption), set

\[
C_0 := \{ x \in H_n^* \mid t_i(x) = 0 \}
\]

\[
C_p := \{ x \in H_n^* \mid t_i(x) = p(i) \}
\]

where \( p \in P \), which is the set defined as

\[
\{ p \in \mathbb{Z}^n \mid \sum_{i=1}^n p(i) = 0, \sum_{i=1}^n |p(i)| < M, \text{and } p(i) \equiv 0 \mod |\omega t_i(\sigma a)| \text{ for all } i \in I \}.
\]

As \( P \) is finite (\( |P(i)| < M/2 \) for every \( i \)), this gives a partition of \( C := \sqcup_{p \in P} C_p \) where \( C \subseteq H_n^* \) and so Lemma 4.6 may be applied.

Our aim must now be to satisfy the assumptions of Proposition 4.9. This will lead to an algorithm satisfying Proposition 4.7, giving us an algorithm for twisted conjugacy for \( H_n \). The first assumption of Proposition 4.9 (that there exists an algorithm to search for a conjugator in FSym) is satisfied by the arguments of Section 4.1. Our work is therefore to produce an absolute bound \( M(\sigma a, \sigma b) \) for a conjugator satisfying condition (11).

Remark 4.10. We have reduced the problem in this way so to use condition (11) to produce the absolute bound \( M \). Proposition 4.7 means that this bound is then sufficient for searching for a conjugator in \( H_n \).

4.3. A bound for \( \sum_i |t_i(x)| \), where \( i \in I \). Our aim is to produce a bound for \( \sum_i |t_i(x)| \) for all \( i \in \{1, \ldots, n\} \) which is computable from only the elements we wish to conjugate. In this section we show, with certain conditions for \( i, j \in I \), that \( t_i(x) \) and \( t_j(x) \) are a bounded distance from each other (which is computable from \( \sigma a, \sigma b \)). We then use this to produce a (computable) bound \( M' \in \mathbb{N} \) such that \( \sum_i |t_i(x)| < M' \) for \( i \in I \). In Section 4.4 we compute \( \sum_j t_j(x) \) (where \( j \in I' \)) for any conjugator \( x \in H_n \) and show that all of values for \( t(x') \) in \( I' \) are computable for a conjugator \( x' \in H_n \).

Definition 4.11. Let \( x \in H_n^* \) be our conjugator. Then, for each \( i \in I \), the numbers \( l_i \) are defined so as to satisfy \( t_i(x) = l_i |t_i(\sigma a)| \).

Let \( \sigma a, \sigma b \in S_n \rtimes H_n \) be conjugate by \( x \in H_n^* \) (consisting of those elements \( g \in H_n \) satisfying condition (11), i.e. \( t_i(g) \equiv 0 \mod |\omega t_i(\sigma a)| \) for all \( i \in I \)). Lemma 3.10 tells us that \( t_i(\sigma a) = t_i(\sigma b) \) for every \( [i] \) class of \( \sigma \). Thus \( X_{i_1, d_1}(a) = X_{i_1, d_1}(b) \) and similarly \( X_{[i],d_i}(a) = X_{[i],d_i}(b) \) (where these sets were defined in (3.2) of Section 3.2). For elements of \( S_n \rtimes H_n \) conjugate by \( x \in H_n^* \), we may therefore write \( X_{i_1, d_1} \) and \( X_{[i],d_i} \) for \( X_{i_1, d_1}(a) \) and \( X_{[i],d_i}(a) \) respectively. Several implications now follow from condition (11). First, we clearly have that \( l_i \in \omega \mathbb{Z} \) for each \( i \in I \).
Secondly, \((X_{i_1,d_1})x\) is almost equal to \(X_{i_1,d_1}\) since the set \(X_{i_1,d_1}\) consists of all points \((i_1,m) \in X_n\) where \(m \equiv d_1 \mod |t_i(\sigma a)|\). Finally, we have that \((X_{i_1,d_1})x\) is almost equal to \(X_{i_1,d_1}\) since \(X_{i_1,d_1}\) is the union of sets \(X_{i_k,d_k}\) where for each \(i_k \in [i]\) we have that \((X_{i_k,d_k})x\) is almost equal to \(X_{i_k,d_k}\). Thus for any \(\sigma a, \sigma b \in S_n \ltimes H_n\) that are conjugate by an \(x \in H_n^*\), the infinite orbits of \(\sigma a\) must be almost equal to those of \(\sigma b\). This means that any such \(\sigma a\) and \(\sigma b\) produce the same equivalence relation of Definition 3.10.

The next proposition means that if a conjugator exists, there is a conjugator for which if \([i] \sim_a [j]\), then \(t_i(x)\) and \(t_j(x)\) are computably close.

**Proposition 4.12.** Let \(\sigma a, \sigma b \in S_n \ltimes H_n\) be conjugate by an \(x \in H_n^*\). Furthermore, let \(t_i(x) = t_i(t_i(\sigma a))\) for all \(i \in I\) and recall each \(t_i\) is in \(Z\). Given any \([i], [j]\) where \([i] \sim_a [j]\) (meaning \([i] \sim_b [j]\)), there exists a computable constant \(K\) (depending only on \(\sigma a, \sigma b\)) such that for all \(i_k \in [i]\) and \(j_k \in [j]\) we have \(||i||l_{i_k}| - ||j||l_{j_k}|\| ≤ K\).

To simplify the proof of this proposition, we give an alternative description of the set \(X_{i_1,d_1}\), and introduce notation for a set almost equal to \(X_{i_1,d_1}\). For a fixed \(\tau g\), a class \([i]_{\tau g}\), and some constants \(q_i\),

\[
X_{i_1,d_1} = \{(i,m) \mid i \in [i]_\tau; \text{and for each } i, m \equiv d_i \mod |t_i(\tau g)|\}
\]

\[
X_{i_1,d_1}|_{q_i} := \{(i,m) \mid i \in [i]_{\tau g}; \text{and for each } i, m \equiv d_i \mod |t_i(\tau g)| \text{ and } m \ge q_i\}
\]

**Proof of Proposition 4.12** We follow in spirit the proof of [ABM13 Prop 4.3].

Since \(\sigma a, \sigma b, x\) are known we may compute \(z, z' \in \mathbb{N}\) such that

\[
(k, m) \sigma a = (k \sigma, m + t_k(\sigma a)), k \in [i] \cup [j], \text{ for all } m \ge z
\]

\[
(k, m) \sigma b = (k \sigma, m + t_k(\sigma b)), k \in [i] \cup [j], \text{ for all } m \ge z
\]

\[
(k, m)x = (k, m + t_k(x)), k \in [i] \cup [j], \text{ for all } m \ge z'
\]

and where \(z\) is chosen to be minimal, i.e. the smallest number to satisfy these equations. To simplify what follows, we shall also assume that \(z' \equiv z\) modulo both \(|t_i(\sigma a)|\) and \(|t_j(\sigma a)|\), and that \(z' ≥ z + \sum_i |t_i(x)|\) where \(i \in [i] \cup [j]\). We may do this because we can increase \(z'\) arbitrarily.

Let \(O_a\) be the orbit of \(\sigma a\) almost equal to \(X_{i_1,d_1} \cup X_{j_1,d_1}\). Then \(X_{i_1,d_1} \cap O_a\) contains \(X_{i_1,d_1}|_{z+\epsilon_i}\) where the numbers \(\epsilon_i\) are chosen so that

\[
(i_1, z + \epsilon_i)(\sigma a)^k \in X_{i_1,d_1} \text{ for all } k \in \mathbb{N}_0 \text{ or for all } k \in \mathbb{Z} \setminus \mathbb{N}
\]

and are chosen to be minimal. Note that finding such \(\epsilon_i\) is possible since for each \(i \in [i]\) we have

\[
z + \epsilon_i \equiv d_i \mod |t_i(\sigma a)| \text{ and } 0 \le \epsilon_i < \sum_{k \in [i]} |t_k(a)|.
\]

We also have that \(X_{i_1,d_1} \cup X_{j_1,d_1}\) is almost equal to \(O_b\) (some orbit of \(\sigma b\)) and, from our choice of \(z\), \(X_{i_1,d_1} \cap O_b\) also contains \(X_{i_1,d_1}|_{z+\epsilon_i}\) for the same \(\{\epsilon_i \mid i \in [i]\}\).

Similarly, we have that \(X_{j_1,d_1} \cap O_a\) and \(X_{j_1,d_1} \cap O_b\) contain \(X_{j_1,d_1}|_{z+\delta_j}\) for minimal numbers \(\delta_j\) where \(z + \delta_j \equiv \epsilon_j \mod |t_j(\sigma a)|\) and \(0 \le \delta_j < \sum_{k \in [j]} |t_k(a)|\) for each \(j \in [j]\). Now, for some finite sets denoted by \(S\) and \(T\), we have \(O_a = A \cup B \cup S\) and \(O_b = A \cup B \cup T\).
Our assumptions imply that \( x \) restricts to a bijection from \( \mathcal{O}_a \) to \( \mathcal{O}_b \) and eventually acts as in (12) on the branches \([i] \cup [j]\). Since \( z' \equiv z \mod |t_\alpha(x)| \), \( x \) restricts to a bijection from \( X_{[i],d_1,|z'|+\epsilon} \) to \( X_{[i],d_1,|z'|+\epsilon}x \), and from \( X_{[j],e_1,|z'|+\delta} \) to \( X_{[j],e_1,|z'|+\delta}x \). Notice that we can only say this for \( z' \), not \( z \). Thus \( x \) restricts to a bijection between the finite sets (13) and (14).

Recall that \( x \) acts as a translation with amplitude \( t_{ik} \) for each \( i \in [i] \). Therefore,

\[
\left| (X_{[i],d_1,|z'|+\epsilon} - X_{[i],d_1,|z'|+\epsilon}) \right| = \left| (X_{[i],d_1,|z'+\epsilon|} - (X_{[i],d_1,|z'|+\epsilon})x) \right| + \sum_{i \in [i]} t_i
\]

\[
\left| (X_{[j],e_1,|z'|+\delta} - X_{[j],e_1,|z'|+\delta}) \right| = \left| (X_{[j],e_1,|z'+\delta|} - (X_{[j],e_1,|z'|+\delta})x) \right| + \sum_{j \in [j]} t_j.
\]

Since \( x \) restricts to a bijection between (13) and (14), these sets must have the same cardinality. Using the observations of (15), we have

\[
|T| = |S| + \sum_{i \in [i]} t_i + \sum_{j \in [j]} t_j.
\]

Now we use Lemma 3.13 to write the sums in terms of one \( i_1 \in [i] \) and one \( j_1 \in [j] \).

\[
|T| - |S| = \frac{1}{|t_i(\alpha)|} \sum_{i \in [i]} t_i(x) + \frac{1}{|t_j(\alpha)|} \sum_{j \in [j]} t_j(x)
\]

\[
= \frac{1}{|t_i(\alpha)|} (|[i]|t_{i_1}(x) + A_{i_1}) + \frac{1}{|t_j(\alpha)|} (|[j]|t_{j_1}(x) + A_{j_1})
\]

\[
= \frac{1}{|t_i(\alpha)|} (|[i]|t_{i_1}(x) + A_{i_1}) + \frac{1}{|t_j(\alpha)|} (|[j]|t_{j_1}(x) + A_{j_1}) + A_{j_1}
\]

\[
= ([i]|t_{i_1} + |[j]|t_{j_1}) + \frac{A_{i_1}}{|t_i(\alpha)|} + \frac{A_{j_1}}{|t_j(\alpha)|} + A_{j_1}
\]

By the generalised triangle inequality we have the required inequality.

\[
|[i]|t_{i_1} \leq |[j]|t_{j_1} + |S| + |T| + \frac{|A_{i_1}|}{|t_i(\alpha)|} + \frac{|A_{j_1}|}{|t_j(\alpha)|}
\]

\[
|[j]|t_{j_1} \leq |[i]|t_{i_1} + |S| + |T| + \frac{|A_{i_1}|}{|t_i(\alpha)|} + \frac{|A_{j_1}|}{|t_j(\alpha)|}
\]

Since for any element in \( S_n \times H_n \) there are finitely many many infinite orbits and these are computable, we may do this process for all \([i']\) and \([j']\) where \([i'] \sim_a [j']\). Lemma 3.11 means that we may now produce a bound for any \( i_k \in [i'] \) and \( j_k' \in [j'] \). □

Now, by exploiting Proposition 4.12, we will be able to show that there is a conjugator \( x \in H_n^* \) with \( \sum \limits_{i \in [i]} |t_i(x)| < M' \), where \( M' \) is computable using only the elements we wish to conjugate. We will do this by using a centraliser argument to prove that there is a conjugator \( x \), which for each \( i \in I \), there exists a \([j] \sim_a [i] \) such that \( t_j(x) = 0 \).

Since it will be used again below, recall that

\[
H_n^* = \{ x \in H_n \mid t_i(x) \equiv 0 \mod |\omega \cdot t_i(\alpha)| \text{ for all } i \in I \}\]
and that \((a_{[i]})^\omega\) is the element in \(C_{H_n}(\sigma a)\) which acts on all rays in classes \([j]\) where \([j] \sim [i]\) (see Definition 3.18).

**Lemma 4.13.** Let \(\sigma a, \sigma b \in S_n \ltimes H_n\) be conjugate by some \(x \in H_n^*\). Then there exists a conjugator \(x' \in H_n^*\) which for each \(i \in I\) satisfies condition (16).

\[(16) \quad \text{There exists a } [j]_\sigma \sim_a [i]_\sigma \text{ such that } t_j(x') = 0 \text{ for some } j' \in [j]_\sigma.\]

**Proof.** Our equivalence relation produces a partition of \([\{j\} \mid j \in I]\). Let \(j^k\) where \(k = 1, \ldots, u\) be a set of representatives for these. Thus for any given \(i \in I\) there is a unique \(k \in \{1, \ldots, u\}\) such that \([j^k] \sim_a [i]\). Now we apply Lemma 3.13 to construct \(a_{[j]}\) for some \(j \in \{j^1, j^2, \ldots, j^u, \ldots, j^u\}\) where \(a_{[j]} = 1\). Replacing the conjugator \(x\) with \(x^* := a_{[j]}^{-1} x\) (well defined since \(l_j \in \mathbb{Z}\)) gives us an element \(x^* \in H_n^*\) (recall that \(l_j \in \omega \mathbb{Z}\)) with \(t_j(x^*) = 0\). Also,

\[(a_{[j]}^{-1} x)^{-1} \sigma a(a_{[j]}^{-1} x) = x^{-1} \sigma a x = \sigma b\]

since \(a_{[j]}\) commutes with \(\sigma a\). Thus \(x^*\) conjugates \(\sigma a\) to \(\sigma b\). Setting

\[x' := \prod_{j \in \Lambda} a_{[j]}^{-1} x\]

produces a conjugator for \(\sigma a\) and \(\sigma b\) satisfying condition (16) for each \(i \in I\).

From Proposition 4.12 and Lemma 4.13 we have the following.

**Proposition 4.14.** Let \(\sigma a, \sigma b \in S_n \ltimes H_n\) be conjugate by some \(x \in H_n^*\). Then there exists a computable bound \(M'\) such that there exists a conjugator \(x' \in H_n^*\) with \(\sum_{i \in I} |t_i(x')| < M'\).

**Proof.** We work for a computable bound for the \(|l_k|\) to produce a bound for the \(|t_i(x')|\). Proposition 4.12 means that there is a computable number \(K\) such that for every \(j^k \in [j]\) and \(i_k \in [i]\) where \([j] \sim_a [i]\), we have

\[||[i]| |l_{i_k}| - |[j]| |l_{j^k}| || \leq K.\]

Using Lemma 4.13 for each class we may choose a \([j]\) and \(j_1 \in [j]\) with \(t_{j_1}(x) = 0\). Thus, for all \(i_k \in [i]\) where \([i] \sim_a [j]\),

\[||[i]| |l_{i_k}| - |[j]| |l_{j_1}| || \leq K \Rightarrow ||[i]| |l_{i_k}| || \leq K \Rightarrow |l_{i_k}| \leq K \Rightarrow \sum_{i_k \in [i]} |l_{i_k}| \leq K.\]

Now, since our equivalence relation partitions the rays into at most \(n\) parts and we may let \(Q := \max \{|t_i(\sigma a)| : i \in I\}\),

\[\sum_{i \in I} |l_k| \leq nK \Rightarrow \sum_{i \in I} |t_i(x)| \leq nKQ.\]

Therefore defining \(M' := nKQ + 1\) provides a suitable bound.

We now bound the remaining rays.
4.4. Computing $t_j(x)$ for all $j \in I^c$. There are two possibilities for the image under $\sigma a$ of the points $(j, m_1)$ where $j \in I^c$ and $m_1$ is sufficiently large: either all of these points are fixed or they are all on a ray on which $\sigma$ acts. In the second case, since $t_j(\sigma a) = 0$ when $j \in I^c$, we have that $(j, m)$ is on an orbit of order $|[j]|$ for all $m \geq m_1$. These points are fixed if and only if $|[j]| = 1$. Note that, since $H^*_n$ poses no restriction on the branches $j \in I^c$, the arguments of this section work as though dealing with $H_n$.

Our strategy will be to show that if a conjugator $x \in H^*_n$ exists, then another conjugator $x' \in H^*_n$ exists where $t_j(x')$ is computable for all $j \in I^c$.

Partitioning $I^c$ into classes of the same size, we have that elements of the centraliser of $\sigma a$ eventually act on each of these partitions disjointly (from Section 3.4). Let $j^1, \ldots, j^v$ be the representatives of these, so that if $j \in I^c$, then $|[j]| = |[j^u]|$ for exactly one $u \in \{1, \ldots, v\}$. Also, let $V = \{j \mid j \in [j^u]\}$ for some $u \in \{1, \ldots, v\}$.

Thus we have, for any $x, x' \in H^*_n$ conjugating $\sigma a$ to $\sigma b$, that

$$
\sum_{k \in \eta(r)} t_k(x) = \sum_{k \in \eta(r)} t_k(x')
$$

where $\eta(r) = \{j^r \in I^c \mid |[j^r]| = r\}$. This means we are free to choose $t_k(x)$ for each class $[k]$ where $k \in I^c \setminus V$. Moreover, defining one suitable combination for $\{t_k(x) \mid k \in \eta(r)\}$ is sufficient for our search for a conjugator.

We may now use the fact that any conjugator must restrict to a bijection between the $r$-cycles of $\sigma a$ and $\sigma b$.

**Definition 4.15.** Let $\tau g \in S_n \ltimes H_n$. For a ray $j \in I^c(\tau g)$, $m_j(g) \in \mathbb{N}$ is the smallest number for the branch $j$ such that for all $1 \leq k \leq |[j]|$, 

$$(j, m)(\tau g)^k = (j \tau^k, m + \sum_{j=1}^{k} t_j \tau^+(g)) \text{ for all } m \geq m_j(g).$$

We will now work with a fixed $r \in \mathbb{N}$ (where $\eta(r) \neq \emptyset$). Let $\gamma$ be the unique element of $V$ such that $|\gamma| = r$, and let

$$t_k(x) := m_k(b) - m_k(a)$$

for all $k \in \eta(r) \setminus \gamma$. Now, to produce a bijection between all $r$-cycles of $\sigma a$ and $\sigma b$, we need only produce a bijection between

$$\{(i, m) \mid i \in \{1, \ldots, n\}, m < m_i(\sigma a)\} \cup \{(i, m) \mid i \in [\gamma], m \geq m_i(\sigma a)\}$$

and the corresponding sets for $\sigma b$.

**Definition 4.16.** Let $\sigma a \in S_n \ltimes H_n$ and $r \in \mathbb{N}$. Then let $s_r(\sigma a)$ denote the number of $r$-cycles within $\{(i, m) \mid i \in \{1, \ldots, n\} \text{ and } m < m_i(a)\}$.

In order to produce a bijection, we now have

$$t_\gamma(x) = m_\gamma(b) - m_\gamma(a) + s_{|\gamma|}(\sigma a) - s_{|\gamma|}(\sigma b)$$

and so we have defined $t_i(x)$ for all $i \in \eta(r)$. Thus if any conjugator $x' \in H_n$ exists, then

$$\sum_{j \in I^c} t_j(x') = \sum_{j \in I^c} t_j(x)$$

by (17).
Notice that the numbers \( \{ m_i(\tau g) \mid i \in I^c \} \) and \( \{ s_r(\tau g) \mid r \in \mathbb{N} \} \) are computable for any \( \tau g \in S_n \ltimes H_n \) and so, if a conjugator exists, then we may compute values for \( \{ t_j(x) \mid j \in I^c \} \) such that \( x \in H_n \) is also a conjugator. Thus, using these values for our algorithm will be sufficient to find a conjugator if one exists.

**Proof of Theorem 1.** We show that the assumptions of Proposition 4.9 are satisfied: that there is an algorithm for searching for a conjugator in \( \text{FSym}(X_n) \) and that we have a partition of a subset of \( H^*_n \) which contains translation lengths so that we may say no conjugator exists if one is not found in this subset. Proposition 4.5 provides an algorithm to search in \( \text{FSym}(X_n) \). Searching for an \( x \in H^*_n \), where \( \sum_{i \in I} |t_i(x)| < M' \) and where the values for \( \{ t_j(x) \mid j \in I^c \} \) are defined as above provide us with a finite subset within which it is sufficient to search. \( \square \)

**Remark 4.17.** Proposition 4.9 does not use the restrictions that the algorithm places on the translation lengths of a conjugator \( x \in H^*_n \). Using these restrictions would provide a more efficient algorithm.

In the next section we use Theorem 1 to show that, for all \( n \geq 2 \), any finite extension of \( H_n \) has solvable conjugacy problem. We then show that, for \( n \geq 2 \), any finite index subgroup of \( H_n \) has solvable conjugacy problem.

## 5. Further results

### 5.1. All finite extensions of \( H_n \) have solvable conjugacy problem.

Our strategy will be to use Theorem 5.3. In this theorem we will set \( F \) to be a finite group and \( D \) to be a Houghton group so that \( E \) realises any finite extension of any \( H_n \). Note that for a group \( G \) and any \( g \in G \), we have that \( \phi_g \) is the automorphism of \( G \) defined by \( (h)\phi_g := g^{-1}hg \) for all \( h \in G \).

**Definition 5.1.** The *action subgroup* of \( G \), denoted \( A_G \), is the set \( \{ \phi_g \mid g \in G \} \).

Note that \( A_G = \text{Inn}(G) \). Also, if \( G \) is an extension of \( H \), then \( H \leq G \) and so we have that \( \text{Inn}(G) \leq A_G \leq \text{Aut}(H) \).

**Definition 5.2.** Let \( G \) be a finitely presented group. Then \( A \leq \text{Aut}(G) \) is orbit decidable if, given any \( a, b \in G \), there is an algorithm that decides (positively or negatively) whether there is a \( \phi \in A \) and \( x \in G \) such that \( a\phi \sim_x b \).

If \( \text{Inn}(G) \leq A \), then orbit decidability asks, given any \( a, b \in G \), to search for a \( \phi \in A \) such that \( a\phi = b \).

The algorithmic condition in the following theorem means that certain computations for \( D, E \), and \( F \) are possible. In our case, this will be satisfied since these groups are recursively presented and the maps between them are defined by the images of the generators.

**Theorem 5.3** (Bogopolski, Martino, Ventura [BMV10]). Let

\[
1 \rightarrow D \rightarrow E \rightarrow F \rightarrow 1
\]

be an algorithmic short exact sequence of groups such that

(i) \( D \) has solvable twisted conjugacy problem,

(ii) \( F \) has solvable conjugacy problem, and
(iii) for every \( 1 \neq f \in F \), the subgroup \( \langle f \rangle \) has finite index in its centralizer \( C_F(f) \), and there is an algorithm which computes a finite set of coset representatives, \( z_{f,1}, \ldots, z_{f,t_f} \in F \),  
\[ C_F(f) = \langle f \rangle z_{f,1} \sqcup \cdots \sqcup \langle f \rangle z_{f,t_f}. \]

Then, the conjugacy problem for \( E \) is solvable if and only if the action subgroup \( A_E = \{ \phi_g \mid g \in E \} \leq \text{Aut}(D) \) is orbit decidable.

\textbf{Remark 5.4.} For all that follows, the action subgroup \( A_E \) is provided as a recursive presentation where the generators are words from \( \text{Aut}(D) \).

\textbf{Definition 5.5.} Let \( G \) and \( H \) be recursively presented groups such that \( H \leq G \). The membership problem for \( G \) and \( H \) asks if there exists an algorithm which, given a word \( g \in G \), decides (positively or negatively) whether \( g \in H \).

\textbf{Lemma 5.6.} Let \( A \) be recursively presented and \( \text{Inn}(H_n) \leq A \leq \text{Aut}(H_n) \) (for some \( n \geq 2 \)). Then the membership problem for \( A \) and \( \text{Aut}(H_n) \) is solvable.

\textbf{Proof.} For each generator \( \phi_c \in A \), compute \( \tau_c \): this can be done by Lemma 5.2. Let \( S := \{ \tau \mid \tau c \) is a generator of \( A_E \} \) and compute \( \langle S \rangle \) which may be considered as a subgroup of \( S_n \) by considering the support of each \( \tau \) to be a subset of the rays \( \{1, \ldots, n\} \). Now, given any \( \tau'c \in \text{Aut}(H_n) \), we have \( \tau'c \in A \) if and only if \( \tau' \in \langle S \rangle \). \( \square \)

\textbf{Proposition 5.7.} Let \( D, E, \) and \( F \) form an algorithmic short exact sequence as in Theorem 5.6. Also let \( D \) have solvable conjugacy problem and \( F \) satisfy conditions (ii) and (iii) of the theorem. If all three of the conditions of (18) are satisfied, then the action subgroup \( A_E \) is orbit decidable.

\textbf{Proof.} First we show that \( \text{Aut}(D) \) is orbit decidable. We wish to find an automorphism \( \hat{\phi}_c \in \text{Aut}(D) \) (where \( \hat{\phi} \in \text{Out}(D) \) and \( \phi_c \in \text{Inn}(D) \)) such that
\[ a\hat{\phi}_c = b \]
\[ \Rightarrow c^{-1} a\hat{\phi} c = b. \]

Enumerating all possible \( \hat{\phi} \) (possible since \( \text{Out}(D) \) is finite) allows us to run the algorithm for conjugacy in \( D \) for all pairs \( (a, b) \).

Restricting ourselves to a \( \hat{\phi}_c \in A_E \) we must again satisfy equation (19) since \( \text{Inn}(D) \leq A_E \). We begin, as above, by checking if there exists a \( c \in D \) for any of the pairs \( (a, b) \). If no such \( c \) exists for any \( \hat{\phi} \in \text{Out}(D) \), then neither will there be in \( A_E \leq \text{Aut}(D) \). For all of the pairs where such a \( c \) exists, we wish to check if \( \hat{\phi} \in A_E \). This is the membership problem for \( A_E \) and \( \text{Aut}(D) \) which we assumed to be solvable. \( \square \)

\textbf{Proof of Theorem 5.8.} In the terminology of Theorem 5.6, set \( F \) to be a finite group and \( D := H_n \) (where \( n \geq 2 \)). These satisfy conditions (i), (ii), and (iii) of this theorem, with no extra conditions needed for \( F \). Thus \( F \) may realise any finite extension of \( H_n \) (for \( n \geq 2 \)). Since \( H_n \) satisfies condition (18) of Proposition 5.7, \( A_E \) is orbit decidable. Thus \( E \) has solvable conjugacy problem as required. \( \square \)
Note that, since $S_n \ltimes H_n \cong \text{Aut}(H_n)$ is a finite extension of $H_n$ for all $n \geq 2$, Theorem 2 implies that it has solvable conjugacy problem. In fact, our algorithm for twisted conjugacy can be adapted to solve conjugacy for any given $\sigma a, \sigma'b \in S_n \ltimes H_n$ by some $\tau x$. First, a necessary condition for $\sigma a \sim_{\tau x} \sigma'b$ is that $\sigma \sim_{\tau} \sigma'$. Thus we may enumerate all $\sigma'$ which conjugate $\sigma$ and $\sigma'$. Then, through a simple computation, the conjugacy problem for $S_n \ltimes H_n$ can be solved by running the algorithm we have constructed in Section 4 for the finite collection of pairs $(\sigma'a', \sigma'b)$.

5.2. All finite index subgroups of $H_n$ have solvable conjugacy problem.

In this section we solve the conjugacy problem for all finite index subgroups of $H_n$ (where $n \geq 2$). First, from [BCMR14], we have the subgroup $U_p$ and Lemma 5.8 (Note that their paper defines $g_i$ to be a different generator of $H_n$, but this does not affect the definition of $U_p$.)

\[
\text{For } p \geq 1, \ U_p := \langle \text{FAlt}, g_i^p \mid i \in \{1, \ldots, n\} \rangle.
\]

**Notation.** Let $A \subset B$ denote that $A$ has finite index in $B$. Also, $\text{FAlt}(X)$ denotes the subgroup of $\text{FSym}(X)$ consisting only of even permutations on $X$.

**Lemma 5.8** (Burillo, Cleary, Martino, Röver [BCMR14]). Let $n \geq 2$. For every finite index subgroup $U$ of $H_n$, there exists an $p \geq 2$ with $\text{FAlt} = U'_p \subset U_p \subset_f U \subset_f H_n$.

This $p$ is chosen to be the smallest even integer such that $(p\mathbb{Z})^{n-1}$ is contained in the image of $U$ under $\pi$ where $\pi : H_n \to \mathbb{Z}^{n-1}$, $(g)\pi \mapsto (t_1(g), \ldots, t_{n-1}(g))$. Since $U_p$ and $U$ are finite index in $H_n$, they are recursively presented.

**Remark 5.9.** Note that, within $U_p$, permutations in $\text{FAlt}$ can be generated by permutations in $\text{FAlt}$ with support contained within $\{(1, m) \mid m \leq p\}$ together with $\{g_i^p \mid i \in \{1, \ldots, n\}\}$. We will use this generating set for all $p \in 2\mathbb{N}$.

The rest of this section will be devoted to solving the twisted conjugacy problem for $U_p \leq H_n$ for any $p \in 2\mathbb{N}$. Using Theorem 5.3 of Section 5.1 we may then conclude that all finite extensions of $U_p$ have solvable conjugacy problem. By Lemma 5.8 this will imply that all finite index subgroups of $H_n$ have solvable conjugacy problem.

Let $U_p \leq_f H_n$ for a fixed $n \geq 2$ and $p \in 2\mathbb{N}$. We wish to know the structure of $\text{Aut}(U_p)$. From arguments for $\text{Aut}(H_n)$ in [BCMR14], we have that every automorphism is conjugation by an element of $\text{Sym}(X_n)$ i.e. $\text{Aut}(U_p) \cong N_{\text{Sym}}(U_p)$, that $N_{\text{Sym}}(U_p) \supseteq S_n \ltimes H_n$, and that almost all of each set $\{(i, qp + d) \mid q \in \mathbb{N}\}$ (where $1 \leq d < q$) is sent to the same branch. We may then deduce that if almost all of one orbit of a branch $j$ remains on $j$, then almost all of the points of branch $j$ remain on $j$. Thus, if $\rho$ is in $N_{\text{Sym}}(U_p)$ and $\sigma_\rho$ is the permutation of the rays by $\rho$, then $\check{\rho} := \rho(\sigma_\rho)^{-1}$ sends almost all of every branch to itself and so we may assume that $\rho$ sends almost all of each branch to itself. We may then use the proof of [BCMR14, Thm 3.2] so that for any such $\rho$ (which sends almost all of each branch to itself) and $g := g_i^p g_j^p$ (where $g_i, g_j$ are standard generators of $H_n$ and $i, j \in \{2, \ldots, n\}$) we have that

\[
t_k(g) = t_k(\rho^{-1} g \rho) \text{ for all } k \in \{1, \ldots, n\}.
\]
Continuing with our assumption on \( \rho \) (that it sends almost all of each branch to itself) we have that where it sends a point ‘sufficiently far out’ defines its action on that orbit on that branch. In the \( H_n \) case this meant that the orbit of a standard generator could be translated along the branch. For \( U_p \), the translation length for every branch is a multiple of \( p \), and so additional automorphisms are defined by where they ‘eventually’ send each of the \( p \) infinite orbits of elements. We will show that in fact any such permutation of the orbits may be achieved by a \( \rho \in N_{\text{Sym}}(U_p) \).

As we will use again the sets used to describe the infinite orbits of \( S_n \ltimes H_n \), recall that for any given \( \tau g \in S_n \ltimes H_n \) we have

\[
X_{i,a}(g) := \{(i_1, m) : m \equiv d \mod |t_{i_1}(\tau g)|\}.
\]

Consider the permutation in \( \text{FSym} \) which permutes \( (1, qp + 1) \) and \( (1, qp + 2) \) where \( q \) is a natural number. By combining all such permutations, i.e

\[
(20) \quad \prod_{q \in \mathbb{N}_0} ((1, qp + 1) (1, qp + 2))
\]

we produce an element with infinite support. Notice that such an element is in \( N_{\text{Sym}}(U_p) \) as it permutes the orbits of one ray (since the translation lengths of \( U_p \) are all a multiple of \( p \)). Conjugation by any standard generator of \( H_n \) means that any finite permutation on the points \( \{(1, qp + d) \mid 1 \leq d < p\} \) may be achieved for all \( q \in \mathbb{N} \). Let \( h_{d_1,d_2} \) denote the element which transposes \( (1, qp + d_1) \) and \( (1, qp + d_2) \) for all \( q \in \mathbb{N} \). With this notation, the element \( (20) \) is denoted be \( h_{1,2} \).

By multiplying \( h_{1,p} \) and \( h_{p,p+1} \), we obtain an infinite cycle. This sends \( (1, (q+1)p) \) to \( (1, qp) \), \( (1, p) \) to \( (1, 1) \) and \( (1, qp + 1) \) to \( (1, (q+1)p + 1) \) for every \( q \in \mathbb{N} \). By conjugating this element by some \( h_{d,d'} \), we may then produce an infinite cycle with support equal to \( X_{i,d_1}(g_i^p) \cup X_{1,d_2}(g_i^p) \) where \( 1 \leq d_1, d_2 \leq p \). Also, conjugation by a permutation of the rays will give us such infinite cycles on any branch.

Sending \( (1, (q+1)p) \) to \( (1, qp) \), \( (1, p) \) to \( (2, p) \), and \( (2, qp) \) to \( (2, (q+1)p) \) for every \( q \in \mathbb{N} \) is also an element in \( N_{\text{Sym}}(U_p) \). The support of this element is the set \( X_{1,q} \cup X_{2,q} \). Again, conjugation by the appropriate \( h_{d,d'} \) and conjugation by a permutation of the branches means we may change the support to be any infinite orbit equal to \( X_{i,d_1}(g_i^p) \cup X_{j,d_2}(g_j^p) \) where \( 1 \leq d_1, d_2 \leq p \) and \( i,j \) are distinct elements of \( \{1, \ldots, n\} \). Thus, for any \( i \in \{2, \ldots, n\} \), we have that \( g_i^p \) may be sent to any element of \( U_p \) which has exactly \( p \) orbits, all of which are infinite. This means that we have found all possible automorphisms of \( U_p \).

Next, we show that if \( U_p \leq H_n \), then \( N_{\text{Sym}}(U_p) \) embeds into \( S_{np} \ltimes H_{np} \). We do this by considering an appropriate generating set of \( N_{\text{Sym}}(U_p) \).

Let \( g_{i,d} \) denote the element with support equal to \( X_{1,1}(g_i^p) \cup X_{i,d}(g_i^p) \) and which acts as a translation by \( p \) on all points in \( X_{1,1}(g_i^p) \), sends \( i, d \) to \( (1, 1) \) and act as a translation by \( -p \) on all other points in \( X_{i,d}(g_i^p) \). With the restriction that \( i \in \{1, \ldots, n\} \) and \( 1 \leq d \leq p \) (and excluding the possibility of \( i = d = 1 \)) we have \( np - 1 \) generators. The key observation is that these generators correspond to the standard generating set of \( H_{np} \) (defined in Section 2.4). One way to see this is to divide each branch \( i \) of \( X_n \) (on which \( N_{\text{Sym}}(U_p) \) acts) into \( p \) branches using the partition

\[
(21) \quad X_{i,1}(g_i^p) \sqcup X_{i,2}(g_i^p) \sqcup \ldots \sqcup X_{i,p}(g_i^p).
\]

Such a bijection between \( X_n \) and \( X_{np} \) can be seen explicitly by bijectively sending the ordered points of \( X_{i,d}(g_i^p) \) to the \( p(i-1) + d^{th} \) branch using the normal ordering
of \(\mathbb{N}\). We can now move on to the other elements of \(N_{\text{Sym}}(U_p)\) and see how they act on \(X_{np}\) (constructed as in (21)). First, the permutation of the branches of \(X_n\) corresponds to a permutation of the branches of \(X_{np}\) (acting on \(p\) times as many branches). Next, the permutation seen in (20) corresponds to a permutation of two branches of \(H_{np}\). This means that we can include all \(p!\) such permutations on the first branch \(X_n\) as generators for \(N_{\text{Sym}}(U_p)\). Finally, \(g_i \in N_{\text{Sym}}(U_p)\), the standard generator of \(H_{np}\), corresponds to an element of \(S_{np} \rtimes H_{np}\). This bijection therefore defines a monomorphism from \(N_{\text{Sym}}(U_p)\) to \(S_{np} \rtimes H_{np}\) where the image is a finite extension of \(H_{np}\), explicitly \((S_p \wr S_n) \rtimes H_{np}\) where \(\wr\) refers to the permutational wreath product. Figure 3 gives the idea of this map for \(n = p = 3\).

\[\begin{align*}
\text{(A) The set } X_n & \quad \text{(B) The set } X_{np}
\end{align*}\]

**Figure 3.** Our monomorphism which sends \(np - 1\) generators of \(N_{\text{Sym}}(U_p)\) to the standard generators of \(H_{np}\), visualised as rotating the coloured rectangles by 90 degrees clockwise.

From now on let this isomorphism between \(N_{\text{Sym}}(U_p)\) and \((S_p \wr S_n) \rtimes H_{np}\) be denoted \(\hat{\phi}\). Note that to solve twisted conjugacy for \(U_p\) we must decide, given an \(a, b \in U_p\) and \(\phi \in \text{Aut}(U_p)\) (which may be realised as conjugation by some \(\rho \in \text{Sym}(X_n)\)) if there exists an \(x \in U_p\) such that

\[(x^{-1})\phi ax = b \Rightarrow x^{-1}\rho ax = \rho b.\]

We will solve this in the image of \(\hat{\phi}\), i.e. we will aim to find an \(x' \in (U_p)\hat{\phi}\) such that

\[(x')^{-1}(pa)\hat{\phi}x' = (\rho b)\hat{\phi}.\]

From the arguments above we have that \((pa)\hat{\phi}\) and \((\rho b)\hat{\phi}\) are elements of \(S_{np} \rtimes H_{np}\). We introduce the following partition to easily define how \(x'\) acts on \(X_{np}\). For \(k \in \mathbb{N}_0\) let \(P_k := \{pk + 1, pk + 2, \ldots, p(k + 1)\}\) so that

\[
\{1, 2, \ldots, np\} = \bigsqcup_{0 \leq k < n} P_k.
\]

We then have that \(x' \in H_{np}\) and if \(i\) and \(j\) lie in the same set \(P_k\), then \(t_i(x') = t_j(x')\). The algorithm of Section 4 therefore provides a large step towards obtaining our algorithm for twisted conjugacy in \(U_p \subset H_n\).

**Lemma 5.10.** Fix an \(n \geq 2\), \(p \in \mathbb{N}\), and let \(U_p \subseteq H_n\). Given any \(a, b \in N_{\text{Sym}}(U_p)\), there exists an algorithm which decides, positively or negatively, whether there exists a conjugator \(x \in U_p\) for \(a\) and \(b\) where \(x\) is a word made from the generators

\[
\{g_i.a \mid 2 \leq i \leq n, 1 \leq d \leq p\} \cup \{g_i.a \mid 2 \leq d \leq p\}.
\]
Proof. Notice that the generating set for $x$ corresponds to the generating set for $H_{np}$ under the map $\hat{\phi}$. This is then the problem solved in Section 4.

If the algorithm of Section 4 produces an $x$ which happens to be in $(U_p)\hat{\phi}$, then we have produced the necessary conjugator. It is possible to decide when we have the necessary conjugator by the following.

**Lemma 5.11.** Let $n \geq 2$. Then the membership problem for $(U_p)\hat{\phi}$ and $S_{np} \ltimes H_{np}$ is solvable.

**Proof.** For any word $\tau g \in S_{np} \ltimes H_{np}$ we may compute the action of $\tau$ and the vector $t(g)$. We may then decide whether $\tau$ is trivial and if $t_i(g) = t_j(g)$ whenever $i$ and $j$ lie in the same set $P_k$. □

We will now extend the algorithm of Section 4 to decide if a conjugator exists in $U_p$. This algorithm provides an $x \in H_{np}$. We must therefore decide, positively or negatively, whether there exists another conjugator $x' \in (U_p)\hat{\phi} \leq H_{np}$. This means that we are only interested in possible translation lengths for elements of $C_{H_n}(\sigma a)$, as once we know $t(x')$ for the conjugator $x' \in (U_p)\hat{\phi}$, we may run our algorithm of Section 4 with these particular translation lengths. From Section 3.4 the translation lengths for elements of $C_{H_n}(\sigma a)$ can be generated by

(i) the computable elements $c_i$ where $i \in I$
(ii) generators which act as a generator of $\mathbb{Z}^{s-1}$ on branches $j \in I^c$ such that there exists a $k \in I^c \setminus \{j\}$ with $||k|| = ||j||$

and that no generators are required for those branches $j \in I^c$ where for all $k \in I^c \setminus \{j\}$ we have $||j|| \neq ||k||$. Recall that the number $s$ shown in (ii) is exactly the number of distinct classes in $I^c$ with size $r_j$.

From these translation lengths we are asking, given $t(x) \in \mathbb{Z}^{np}$, whether there exists a $c \in t(C_{H_n}(\sigma a))$ such that $t(xc) \in A$ where $A$ consists of those $a \in \mathbb{Z}^{np}$ such that for any $k$, if $i, j \in P_k$, then $a_i = a_j$. Explicitly

$$
(22) \quad A = \{(a_1, \ldots, a_1, a_2, \ldots, a_2, \ldots, a_n, \ldots, a_n) \in \mathbb{Z}^{np} \mid \sum_{i=1}^{np} a_i = 0\}
$$

and so may be considered as being $\mathbb{Z}^{n-1}$.

We now show that an algorithm to decide if such a $c \in t(C_{H_n}(\sigma a))$ exists, computing it if it does. First, we have that a generating set for $t(C_{H_n}(\sigma a))$ is computable. It also has several important properties. First, the generators have support almost equal to entire branches in either $I$ or in $I^c$, and those acting on almost all of a branch in $I$ act disjointly as a partition of these branches (i.e. for every branch in $I$ there is one, and only one, generator which acts on almost all of that branch). Those acting on $I^c$ provide a different picture, as either no generator acts on almost all of a branch or a generator acts on all branches in $I^c$ with the same class size.

It will be useful to describe the translation lengths of elements of the centraliser with a vector $q \in \mathbb{Z}^{np}$ which we define as follows.

**Definition 5.12.** Given the description of the generators of the centraliser we may input them into a vector $q \in \mathbb{Z}^{np}$, using our descriptions (i) and (ii) of the centraliser above. Let the $k^{th}$ entry of $q$ relate to the translation length of the centraliser on the $k^{th}$ branch. For those branches $i \in I$, let the entries of $q$ correspond to the
is computable, and from our definition of $q$, we also have three necessary and sufficient conditions for the \{\alpha_j \mid j = 1, \ldots, np\}:

- if $j, k \in [i]$, then $\alpha_j = \alpha_k$
- if $i \in I$ and $[i] \sim \alpha [j]$, then $\alpha_i = \alpha_j$
- if $j \in \eta(r)$, then $\sum_{k \in \eta(r)} \alpha_k = 0$

where $\eta(r) = \{j' \in \mathcal{I}^c : |[j']| = r\}$. This gives us a system of equations which have a solution if and only if a suitable element of the centraliser exists. We note that it is decidable whether or not these linear Diophantine equations have a solution (since general solutions can be described with additional linear expressions).

Our aim is now to decide whether or not there exists a conjugator whose finite permutations lie in FAlt. One can decide whether the conjugator $x \in H_{np}$ (and so $x_\hat{\phi}$) satisfies this condition, for example by using that the word problem to tell us that all such finite permutations lie in a computable set (since $U_p \leq H_n$). Again, should these finite permutations of $x$ not lie in FAlt we may decide whether the required conjugator lies in $(U_p)\hat{\phi}$ by deciding if there is an odd finite permutation in the centraliser (because of Lemma 3.15).

First we consider, for our given element $\sigma a \in S_{np} \rtimes H_{np}$, whether there are any branches $j \in \mathcal{I}^c$. If so, either $|[j]|$ is odd or even. If it is even, then we have that the first $|[j]|$-cycle on this branch is an odd, finite order element of the centraliser. Alternatively if $|[j]|$ is odd, then the element that permutes only the first two $|[j]|$ cycles of the branches $[j]$ provides an odd, finite order element of the centraliser.

If there are no such branches (i.e. if $I(\sigma a) = \{1, 2, \ldots, np\}$), then all of the finite permutations of $\sigma a$ lie inside a finite, computable subset of $X_{np}$ (by the solution to word problem for $H_n$ seen in [ABM13]). Therefore we may search for an odd permutation within this set to decide positively or negatively whether there exists a conjugator in $(U_p)\hat{\phi}$.

We are now free to apply Theorem 5.3 to obtain that if $E$ is a finite extension of $U_p$, then $\text{CP}(E)$ is solvable if and only if $A_E \leq \text{Aut}(U_p)$ is orbit decidable.

**Proposition 5.13.** Let $n \geq 2$, $p \in 2\mathbb{N}$, and $U_p \subseteq H_n$. If $E$ is a finite extension of $U_p$, then $A_E$ is orbit decidable.

**Proof.** Recall that for $A_E$ to be orbit decidable, there must exist an algorithm which decides, given any $a, b \in U_p$, whether there exists a $\psi \in A_E$ such that

$$(a)\psi \sim b$$

and since $\text{Inn}(U_p) \leq A_E$, this is equivalent to finding a $\phi \in A_E$ such that

$$(a)\phi = b$$

which implies that $\rho^{-1}a\rho = b$ for some $\rho \in N_{\text{Sym}}(U_p)$. Using the monomorphism $\hat{\phi}$ rephrases this problem with elements in $S_{np} \rtimes H_{np}$, and so it can be solved in the same way as the proof of Theorem 2 in the previous section. \qed
References

[ABM13] Y. Antolín, J. Burillo, and A. Martino, *Conjugacy in Houghton’s groups*, arXiv:1305.2044 (2013).

[BCMR14] J. Burillo, S. Cleary, A. Martino, and C. E. Röver, *Commensurations and Metric Properties of Houghton’s Groups*, arXiv:1403.0026 (2014).

[BMV10] O. Bogopolski, A. Martino, and E. Ventura, *Orbit decidability and the conjugacy problem for some extensions of groups*, Trans. Amer. Math. Soc. 362 (2010), no. 4, 2003–2036. MR 2574885 (2011e:20045)

[Boo59] W.W. Boone, *The word problem*, Annals of mathematics 70(2) (1959), 207–265.

[Bro87] Kenneth S. Brown, *Finiteness properties of groups*, Proceedings of the Northwestern conference on cohomology of groups (Evanston, Ill., 1985), vol. 44, 1987, pp. 45–75. MR 885095 (88m:20110)

[Bro94] Kenneth S. Brown, *Cohomology of groups*, Graduate Texts in Mathematics, vol. 87, Springer-Verlag, New York, 1994, Corrected reprint of the 1982 original.

[CM77] Donald J. Collins and Charles F. Miller, III, *The conjugacy problem and subgroups of finite index*, Proc. London Math. Soc. (3) 34 (1977), no. 3, 535–556. MR 0435227 (55 #8187)

[DM96] John D. Dixon and Brian Mortimer, *Permutation groups*, Graduate Texts in Mathematics, vol. 163, Springer-Verlag, New York, 1996. MR 1409812 (98m:20003)

[GK75] A. V. Gorjaga and A. S. Kirkinski˘ı, *The decidability of the conjugacy problem cannot be transferred to finite extensions of groups*, Algebra i Logika 14 (1975), no. 4, 393–406. MR 0414718 (54 #2813)

[Hou78] C. H. Houghton, *The first cohomology of a group with permutation module coefficients*, Archiv der Mathematik 31 (1978), 254–258.

[Joh99] D. L. Johnson, *Embedding some recursively presented groups*, Groups St. Andrews 1997 in Bath, II, London Math. Soc. Lecture Note Ser., vol. 261, Cambridge Univ. Press, Cambridge, 1999, pp. 410–416. MR 1676637 (2000h:20057)

[Lee12] S. R. Lee, *Geometry of Houghton’s Groups*, arXiv:1212.0257v1 (2012).

[Mil71] C.F. Miller III, *On group-theoretic decision problems and their classification*, Annals of Math. Studies 68, (1971).

[Nov58] P.S. Novikov *On the algorithmic unsolvability of the word problem in group theory*, Trudy Mat. Inst. Steklov. 44 (1955), 143 pages. Translation in Amer. Math. Soc. Transl. 9(2) (1958), 1–122.

[Sco87] W. R. Scott, *Group theory*, second ed., Dover Publications, Inc., New York, 1987. MR 896269 (88a:20001)

[SJG12] S. St. John-Green, *Centralisers in Houghton’s groups*, to appear Proc. Edin. Math. Soc.