FUNDAMENTA
MATHEMATICAE
235 (2016)

Algorithms for Nielsen type periodic numbers of maps with remnant on surfaces with boundary and on bouquets of circles II

by

Evelyn L. Hart (Hamilton, NY), **Philip R. Heath** (St. John's, NL), and **Edward C. Keppelmann** (Reno, NV)

Abstract. In this second and final paper of the series, we sketch an algorithm for the computation of the Nielsen type number $N\Phi_n(f)$ for periodic points of maps f of remnant 2 on surfaces with boundary, and on bouquets of circles. The number $N\Phi_n(f)$ is the second, and more complicated, of two Nielsen type periodic point numbers. In the first paper we exhibited an algorithm for the computation of $NP_n(f)$, the first of these numbers.

The class of spaces and maps under consideration in this paper do not satisfy the, by now familiar, conditions that give rise to computational theorems for $N\Phi_n(f)$. Because of this we are thrown back on the definition which requires that we find the minimum height among all sets of n-representatives for f. Our technique, which is presented with a view to clarity rather than for efficiency of computation, is to sketch an algorithm for the construction of a finite weighted graph $(\mathcal{U}(h,n),\mathcal{D})$ whose nodes are orbits, and whose edges are the "boosts" between individual orbits. The graph $\mathcal{U}(h,n)$ is universal in the sense that the set of nodes contains all minimum sets of n-representatives. The graph is weighted by means of data labels $\mathcal{D}$, which we use to determine which subset of the nodes of $\mathcal{U}(h,n)$ are sets of n-representatives and to compute their heights.

Two factors complicate the algorithmic computation of $(\mathcal{U}(h,n),\mathcal{D})$. The first is the need to include certain nodes in $\mathcal{U}(h,n)$ that represent empty orbits, which of course are not detectable by the Reidemeister trace. The second is the need to attach data $\mathcal{D}(P^k)$ to the nodes P^k of $\mathcal{U}(h,n)$.

Our method requires that we modify and extend word length arguments from the first paper. In the process we solve the twisted conjugacy problem for homomorphisms with limited cancellation among the generators. This potentially important result is surprisingly simple to prove.

2010 *Mathematics Subject Classification*: Primary 55M20.
Key words and phrases: Nielsen number, Nielsen periodic number, fixed point, periodic point, Reidemeister number, surface, punctured disc, surface with boundary, bouquet of circles, wedge of circles, figure eight, remnant.
Received 15 October 2014; revised 3 January 2016.
Published online 18 July 2016.

DOI: 10.4064/fm45-1-2016

1. Introduction. In this paper, which continues [2], we sketch an algorithm for the computation of the Nielsen periodic point number $N\Phi_n(f)$ for maps f that have limited cancellation between the images of generators in the fundamental group (maps with remnant at least 2, Definition 2.5). The number $N\Phi_n(f)$ is a homotopy invariant of f and a lower bound for the number of periodic points of all periods dividing n.

The numbers $NP_n(f)$ and $N\Phi_n(f)$ are complicated (a) by the fact that they are homotopy invariants of f rather than of the iterate f^n, (b) because we need to deal with orbits rather than classes (see [6] or [5] for an explanation), and (c) because both the Nielsen and Reidemeister classes and orbits at various levels interact. In particular a Nielsen or Reidemeister orbit can "contain" Nielsen respectively Reidemeister orbits at lower levels. These containments are registered algebraically by means of what we call boosting functions ($\iota_{k,m}$ for $k \mid m \mid n$, Section 2.1). In general the boosting functions are not injective. If $\iota_{k,m}(P^k) = O^m$ for Reidemeister orbits P^k and O^m at levels k and m respectively, we say that P^k boosts to O^m, and that O^m reduces to P^k. If an orbit reduces only to itself ($\iota_{k,k}$ is the identity) then that orbit is said to be irreducible.

The number $NP_n(f)$ is an f homotopy invariant lower bound for the number of periodic points of period exactly n. It is computed by taking n times the number of essential (Section 2.1 again) irreducible Reidemeister orbits. Thus our task in computing $NP_n(f)$ in [2] was "simply" to determine this latter number.

The definition of the Nielsen type number $N\Phi_n(f)$ (Definition 2.3) is much trickier than that of $NP_n(f)$. It is easy to misunderstand, and even more so to underestimate what is involved in giving an algorithm to compute it (see Remark 6.6(ii)). The main complicating factor, when looking at $NP_n(f)$ rather than $N\Phi_n(f)$, is that when two or more essential orbits reduce to a single orbit P^k at a lower level, the various essential orbits are only detecting a single periodic point orbit. What complicates it even more is that such a P^k can be inessential, even empty [1]. In this regard, it seems prudent to remind the reader that a procedure outlined in [6] is wrong because it fails to take this important point into account [2] (see Remark 2.4).

In a large class of spaces and maps, the scenario just described cannot occur. In particular this is the case if the space (all maps) or an individual

[1] A Reidemeister class is said to be *empty* if it is the image, under the usual inclusion, of an empty Nielsen class. An orbit is empty if it is the union of empty classes.

[2] An example illustrating this is given in [5, Example 3.1]. It is accounted for geometrically by the possibility that two or more periodic point orbits at higher iterates can, through a homotopy of f, be coalesced to an inessential (or empty) orbit at the lower level.

map is essentially reducible [3]. Under this and other conditions, algorithmic procedures, and even formulas, are more easily procured (see [5, 3]). All maps on nil or solvmanifolds are essentially reducible [3, Corollary 4.5]. However there are many (perhaps even most) maps on the spaces under consideration in this paper, that are not [4]. What all this means in practice is that when computing $N\Phi_n(f)$, we need to take into account the possibility of a (potentially large) number of possibly empty orbits to which the various essential orbits reduce. Of course, such orbits are not visible in the Reidemeister trace.

Thus as far as this paper goes, we are thrown back on the definition which requires that we compute the minimum height among all possible sets of n-representatives (Definition 2.3) [5]. Our strategy is to use the techniques of [2] together with our modified word length arguments to algorithmically construct a weighted graph $(\mathcal{U}(h,n),\mathcal{D})$. Here $\mathcal{U}(h,n)$ is the graph (Definition 3.3), h is the homomorphism induced by f, and $\mathcal{D}$ the weighting of $\mathcal{U}(h,n)$ by data labels (Definition 4.4). The nodes of $\mathcal{U}(h,n)$ are orbits, and there is a directed edge between two orbits precisely when one boosts to the other (Definition 3.3). The graph is universal in the sense that its nodes contain all minimum sets of n-representatives (Theorem 4.8). In particular it must contain all necessary empty orbits. We refer to the weighting of each node of $\mathcal{U}(h,n)$ as data labels ($\mathcal{D}$). More precisely, for each node P^k of of $\mathcal{U}(h,n)$, we attach certain data $\mathcal{D}(P^k)$. This is needed in order to algorithmically determine which subsets of the nodes of $\mathcal{U}(h,n)$ are sets of n-representatives, and to determine their heights. After wrestling with the problem for some time, we felt that the simplest way to proceed with our algorithm is to construct $\mathcal{U}(h,n)$ and $\mathcal{D}$ simultaneously and progressively, as the information becomes available.

Intuitively an abstract theoretical proof that $(\mathcal{U}(h,n),\mathcal{D})$ exists, is finite and satisfies the required properties, follows relatively easily from the finite-to-one nature of reductions guaranteed by the modified word length arguments we give here. On the other hand, when you start to look at the details of how to construct both $\mathcal{U}(h,n)$ and $\mathcal{D}$ systematically, complicat-

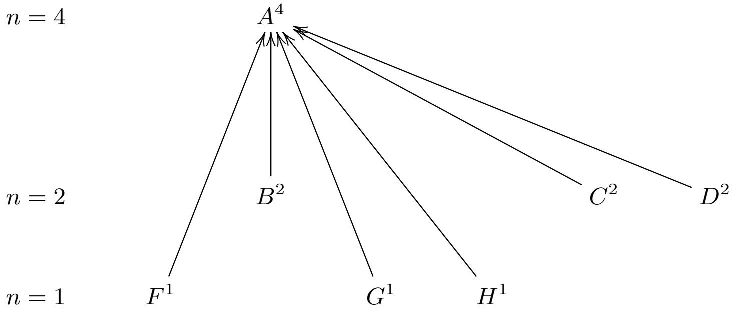
[3] A map is *essentially reducible* if essential orbits reduce only to essential orbits [3, Definition 4.1].

[4] In [2, Example 2.9 (part three)] the orbit $\langle[1]^1\rangle$ of the wedge point is inessential at level 1, but $\iota_{1,2}(\langle[1]^1\rangle)=\langle[1]^2\rangle$ is essential at level 2. This phenomenon is not uncommon. Though in principle it is not difficult to construct meaningful examples, they do tend to be long and involved.

[5] The reader unfamiliar with the definitions might find it helpful to study the fictitious illustration (6.5) to see how a fixed set of geometric orbits (without specifying which are essential) can exhibit a wide variety of possible values for $N\Phi_n(f)$ (see also [5, Section 3]).

ing factors start to reveal themselves. We need to know where to start (for example, do we start with orbits at the bottom or the top, or with essential orbits?). Having chosen where to start, we need to know how to choose representatives of the chosen orbits. Depending on the choices, these orbits may or may not be visible in the reduced Reidemeister trace. This is where the work of [2] comes in. As we shall see with our choice of $\mathcal{U}(h,n)$ (see Remark 3.5), the graph has the structure of the disjoint union of components indexed by the non-empty orbits (denoted O^n) at the top (level n), and this is where we start. These orbits need not be essential, and when they are not, they are not of course detected by the reduced Reidemeister trace. They are however detected by the geometry of the special representative of the homotopy class of f (see [2] for details). This choice of starting point also furnishes us with a natural way to choose representatives. In particular the extended Wagner algorithm given in [2] allows us to choose Wagner tails, u say, of shortest length as representatives of the O^n (Subsection 2.2). Using these u, we use our modified word length arguments (Lemma 4.1) to find all group element reductions of u, at each level dividing n. Word length arguments again can then be used, at each level, to determine from the group element reductions first the Reidemeister classes, and then orbits. We can also easily add the edges from our reduced set of orbits to O^n at this point, and deduce that what we have constructed thus far is finite.

At this stage we have constructed what we call the skeleton $\mathrm{Sk}(\mathcal{S}(O^n))$ of the support $\mathcal{S}(O^n)$ of O^n (Definitions 5.3 and 3.1). The graph $\mathrm{Sk}(\mathcal{S}(O^n))$ is a subgraph of the component of $\mathcal{U}(h,n)$ determined by O^n. It does not yet contain all the edges, and we need to find a way to systematically add them. By way of illustration, the skeleton $Sk(\mathcal{S}(A^4))$ of the support of a single orbit A^4 at level 4 could be represented by the diagram



where B^2, C^2 and D^2 represent all reductions of A^4 to level 2, and F^1, G^1 and H^1 represent all orbit reductions to level 1. We need to know how the last three orbits F^1, G^1 and H^1 boost to A^4, that is, how they factor through B^2, C^2 and D^2. Of course the skeletons can be much more complex than in this illustration. The skeleton of the support of a single orbit at level 12, for example, will be missing edges from level 1 to levels 2, 3, 4 and 6, from level 2 to levels 4 and 6, and from level 3 to level 6.

If all that was needed was to add these edges, we could do this in the illustration for example by computing each of $\iota_{1,2}(F^1)$, $\iota_{1,2}(G^1)$ and $\iota_{1,2}(H^1)$ and then, using word length arguments, to identify the boosted orbits among B^2, C^2 and D^2. In fact if we were to construct $\mathcal{U}(h,n)$ in this way, it would contain, abstractly at least, most ([6]) of the information we need in order to determine which subsets of the nodes are sets of n representatives, and to compute their heights. In particular the crucial information about the depth ([7]) of each orbit would already be encoded in $\mathcal{U}(h,n)$. However, accessing this information in an algorithmic way is a different question. On the other hand, with the construction of the skeleton of the support of each orbit, its depth comes for free. For example in the illustration the depth of A^4 is 1. For this reason, the way we construct the component $\mathcal{S}(O^n)$ of $\mathcal{U}(h,n)$ is to construct the skeleta of each of the nodes of $\mathrm{Sk}(\mathcal{S}(O^n))$, and then merge (Lemma 5.5) the newly constructed skeletons into $\mathrm{Sk}(\mathcal{S}(O^n))$. Since we know the depth of each skeleton we are merging, we can encode this information into the data labels at this time. In this way we are able to construct both $\mathcal{U}(h,n)$ and $\mathcal{D}$ simultaneously and progressively as indicated earlier.

We illustrate the merging of the skeleton $\mathrm{Sk}(\mathcal{S}(B^2))$ of the support $\mathcal{S}(B^2)$ of B^2 into $\mathrm{Sk}(\mathcal{S}(A^4))$ in the illustration above. Suppose that $\mathrm{Sk}(\mathcal{S}(B^2))$ is represented by the diagram

$$\begin{tikzcd}
n=2 & & B^2 & \\
n=1 & J^1 \arrow[ur] & & K^1. \arrow[ul]
\end{tikzcd}$$

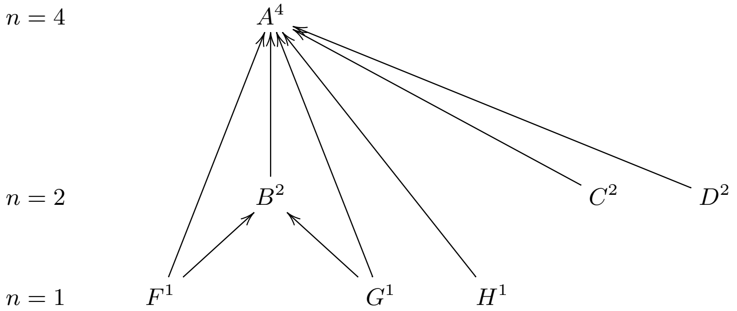
The depth of B^2 is of course 1, and we record this as one component ($\mathcal{D}_d(B^2)$, d for depth) of the data label $\mathcal{D}(B^2)$ for B^2. We have labeled the reductions J^1 and K^1, because our procedure will not immediately identify these orbits among F^1, G^1 and H^1. And of course, the next step is to do exactly that, using word length arguments again ([8]). Suppose that $J^1 = G^1$ and $K^1 = F^1$.

[6] We would also need essentiality, which is easily obtained by identifying essential orbits using the techniques of [2], and then using word length arguments to match them in the constructed graph.

[7] The depth of an orbit is the lowest level to which the orbit reduces. The height of a set of n-representatives is the sum of the depths of its members (Section 2.1).

[8] Alternatively, after $\mathrm{Sk}(\mathcal{S}(O^n))$ has been constructed, we can use information thus obtained to construct and merge the other $\mathrm{Sk}(\mathcal{S}(P^m))$. It is, however, not as easily described (see Section 6.2).

Then the merged graph of the latter skeleton into the former skeleton will look like



In the illustration the depth of B^2 could have been easily determined without the process of merging. However, the more factors a divisor m of n has, the more complex an *algorithmic* procedure would be for determining the depth of an orbit at that level from $\mathcal{U}(h,n)$ alone. As we have also said, our primary goal is to show the process can be made algorithmic. Though perhaps the process of merging is not the most efficient way to show this, having wrestled with it for some time we think it is probably the simplest way to understand the dual tasks of algorithmically adding the edges, and determining the depth of each orbit [9]. In the end, if we combine it with the ideas outlined in the efficiency section, merging may also turn out to be the most efficient way to construct $(\mathcal{U}(h,n),\mathcal{D})$.

As remarked in the Abstract, the word length argument here is a modification of one we used in [2]. It required that we solve the twisted conjugacy problem for homomorphisms with remnant 2 (Theorem 4.2). Since it partially solves the general (difficult) twisted conjugacy problem in algebra, it is our belief that its easy proof belies its importance.

The paper is organized as follows: After this introduction, in Section 2, we outline the definitions and notation we need to make the paper somewhat (but not completely) self-contained. We assume some familiarity with [2] (where our algorithm for $NP_n(f)$ was given). In Section 3, we give the definition and structure of our choice of the universal graph $\mathcal{U}(h,n)$. In Section 4, we give our word length arguments, solving the Twisted Conjugacy Theorem for maps of remnant 2. We define the data labels and show that $\mathcal{U}(h,n)$ is universal in the sense indicated. The algorithm itself comes next in Section 5. Finally in Section 6 we make the comments that relate to simplifying and making the computation of $N\Phi_n(f)$ more tractable. We use a fictitious example that illustrates how it might be possible to shortcut, in various ways, full details in the construction of $\mathcal{U}(h,n)$.

[9] The reader is invited to attempt an algorithmic way of computing $N\Phi_n(f)$ using the method suggested in the earlier paragraph that starts with “If all that was needed.”

2. Preliminaries. Though the definitions of $NP_n(f)$ and $N\Phi_n(f)$ can be made on more general spaces, we work in this paper (as in [2]) with spaces X which have the homotopy type of a bouquet (wedge) of circles with r oriented loops. Since X is a $K(\pi, 1)$, we can, without loss, regard X as any compact surface with boundary of the same homotopy type as f. Also each homotopy class of self-maps is uniquely determined by the induced homomorphism $f_* : \pi_1(X) \to \pi_1(X)$. Therefore we may concentrate largely on the homomorphism f_* which we denote by h. The fundamental group of X is a free group $G := \pi_1(X) = \langle a_1, \ldots, a_r\rangle$ of rank r with generators $\mathcal{G} = \{a_1, \ldots, a_r\}$. By abuse of notation, having chosen a fixed orientation for each loop of X, we label the loops by the same a_i for $i = 1, \ldots, r$.

2.1. $NP_n(f)$ **and** $N\Phi_n(f)$ **and the basic periodic point definitions.** In this subsection we recall the basic definitions. Let $w_1, w_2 \in G$ and $m, n \in \mathbb{N}$ be such that $m \mid n$ and n is fixed throughout the paper. We say that w_1 and w_2 are *Reidemeister equivalent* at level m (or that w_1 and w_2 are Reidemeister equivalent for h^m) if there is a $z \in G$ satisfying the equation $zw_1h^m(z^{-1}) = w_2$. We write $w_1 \sim_m w_2$, and use the symbol $\mathcal{R}(h^m)$ to denote the set of Reidemeister classes at level m. The Reidemeister class of α for h^m is denoted $[\alpha]^m$. We use $\mathcal{RO}(h^m)$ to denote the set of Reidemeister orbits at level m. The *Reidemeister orbit* containing $[\alpha]^m$ is the set

$$\langle[\alpha]^m\rangle := \{[\alpha]^m, [h(\alpha)]^m, [h^2(\alpha)]^m, \ldots, [h^{\ell-1}(\alpha)]^m\}.$$

Here ℓ (called the *length* of the orbit) is the number of classes in the orbit. That is, ℓ is the smallest positive integer for which $[h^\ell(\alpha)]^m = [\alpha]^m$. Note that $\ell \mid m$ and can be strictly smaller than m (i.e. the length of a representative group element, see [4]). We write $\langle[\alpha]^m\rangle \in \mathcal{RO}(h^m)$, but we will also use symbols such as O^m, P^m and Q^m etc. to denote orbits at level m.

Let $k, m \in \mathbb{N}$ be such that $k \mid m \mid n$. We define $\iota_{k,m} : G \to G$ on $\alpha \in G$ by

$$\iota_{k,m}(\alpha) = \alpha h^k(\alpha)h^{2k}(\alpha)\cdots h^{m-k}(\alpha).$$

Note firstly that $\iota_{k,m}$ is not a homomorphism on these spaces, but that it is well-defined on Reidemeister classes, and also on Reidemeister orbits (see [4]). By abuse of notation we use $\iota_{k,m}$ to denote all three functions: $\iota_{k,m} : G \to G$, $\iota_{k,m} : \mathcal{R}(h^k) \to \mathcal{R}(h^m)$, and $\iota_{k,m} : \mathcal{RO}(h^k) \to \mathcal{RO}(h^m)$. We call the $\iota_{k,m}$ *boosting functions* for all three contexts (group elements, Reidemeister classes, orbits). If $\iota_{k,m}(y) = z$, we say that y *boosts* to z and that z *reduces* to y, or that y is a *reduction* of z, and for any divisor r of k, the set $\iota_{r,k}^{-1}(\{z\})$ is called the set of *reductions* of y from level k to level r. The *depth* d of y at level k is the minimum d for which the set (set of classes, set of orbits) $\iota_{d,k}^{-1}(\{y\})$ is non-empty. Depth is the same on group elements, Reidemeister classes and orbits (see Lemma 2.1 below). Finally, the element y at level k is said to be *irreducible* if its depth is k. Otherwise, it is *reducible*.

The following lemma from [1] allows us to refer to reducibility, irreducibility and depth simultaneously on elements, classes and orbits without ambiguity. We refer to it as "on the nose" boosting at the group level.

LEMMA 2.1 (On the nose boosting, Hart–Keppelmann [1]). *Let $\langle[\alpha]^m\rangle \in RO(h^m)$ and $\langle[\beta]^k\rangle \in RO(h^k)$ be such that $\iota_{k,m}(\langle[\beta]^k\rangle) = \langle[\alpha]^m\rangle$. Furthermore let $\gamma \in G$ be such that $[\gamma]^m \in \langle[\alpha]^m\rangle$. Then there exists a $\nu \in G$ for which $[\nu]^k \in \langle[\beta]^k\rangle$ and $\iota_{k,m}(\nu) = \gamma$.* ■

We assume that the reader is familiar with the concept of essential and inessential Reidemeister classes and orbits, as found in the modified fundamental group approach (i.e. [4, 3, 2]). In particular we assign essentiality (or not) to each Reidemeister class or orbit. So $[v]^k$ is *essential* if it is in the image, under the usual inclusion, of an essential Nielsen class into the corresponding Reidemeister class. Otherwise it is *inessential.* Clearly empty orbits are inessential (see footnote 1). Essentiality is a property of orbits [4, Corollary 1]. We use the symbol $\mathcal{EO}(h^m) \subseteq \mathcal{RO}(h^m)$ to denote the set of essential orbits at level m, while $\mathcal{IEO}(h^m)$ denotes the set of *irreducible* essential orbits. The *depth* of an orbit is the lowest level to which the orbit reduces. The *height* of a set of n-representatives is the sum of the depths of its members.

DEFINITION 2.2 (Set of n-representatives for h). A subset S of $\bigcup_{m|n} \mathcal{RO}(h^m)$ is a *set of n-representatives for h* if each orbit in $\bigcup_{m|n} \mathcal{EO}(h^m)$ reduces [10] to some orbit in S. A *minimal set of n-representatives for h* is a set of n-representatives with minimal height.

The reader will recall that $NP_m(f) := m \cdot \#(\mathcal{IEO}(h^m))$, and that the set $\bigcup_{m|n} \mathcal{IEO}(h^m)$ is a subset of any set of n-representatives for h [6, 5].

DEFINITION 2.3 (Jiang [6]). The *Nielsen type periodic point number $N\Phi_n(f)$* for f is the height of a minimal set of n-representatives for h.

REMARK 2.4 (Misunderstanding $N\Phi_n(f)$). The definition of $N\Phi_m(f)$ is due to Boju Jiang [6], and is quite brilliant. It is however very easy to misunderstand, and there are many pitfalls (see [5] for a discussion of this). It seems prudent therefore to remind the reader that a procedure suggested for its computation in [6] does not work. The suggestion, if correct, would greatly simplify things and in fact make redundant many of the things we do here. The suggestion in question was to take the height of the set S of all essential orbits of any period $m \mid n$, which do not reduce to any essential orbit of lower period. However, two or more such orbits may reduce to a

[10] We include identity boosts, in other words we regard an element as reducing to itself.

single inessential orbit at a lower level, and hence be detecting only one periodic point orbit. An example of this is given in [5, Example 3.1], where for $n = 6$ on a simply connected space, the single orbits at each of the levels 6, 3 and 2 are all essential, but reduce to the single orbit at level 1. This orbit is inessential. In fact the given map can be homotoped to one with a single periodic point which occurs at level 1. In particular $N\Phi_6(f) = 1$, while the false procedure suggested above would compute it as 3 (see also [5, comments on p. 226]).

2.2. Wagner tails, the sets $\mathcal{WO}(h^m)$, remnant. In this subsection we briefly recall the definitions and concepts from [2] that we need in the first step of the construction of $\mathcal{U}(h, n)$. We refer the reader there for details.

Let $\mathcal{G}^{\pm} = \{a_1, \ldots, a_r, A_1, \ldots, A_r\}$, where A_i denotes the inverse of a_i. Then any element $w \in G$ ($= \pi_1(X)$, recall) is a finite word in the alphabet $\mathcal{G}^{\pm}$. We will denote the length of a reduced word $w \in G$ by $|w|$, with the usual understanding that the identity element has length 0. Recall from [2, Section 2.2, p. 108] that using only h, we construct a very precise piecewise linear representative f of the single homotopy class of maps that induces h. We call f the *special representative* of its homotopy class. If we write $h^m(a_i) = c_1 \cdots c_s$ as the *unreduced* image of a_i where each c_j is in $\mathcal{G}^{\pm}$, then there is a one-to-one correspondence between fixed points of f^m and the occurrences of a_i or A_i in the unreduced word $h^m(a_i)$. We emphasize that this has to be unreduced, since this one-to-one correspondence fails if the words are reduced. Moreover (as discussed in [2, p. 110]), certain subwords of the $h^m(a_i)$ determine the Reidemeister classes of f^m at level m. We call the set of all such subwords *Wagner tails.* The Reidemeister classes determined by the Wagner tails represent all non-empty Nielsen classes of our special representative f. Such classes may however become empty for maps homotopic to f.

Let

$$\mathcal{WO}(h^m) = \{P^m \in \mathcal{RO}(h^m) : P^m \text{ is represented by a Wagner tail}\}.$$

We call such P^m *Wagner orbits.* Since essential classes are non-empty under deformation, they can also be represented by one or more of the Wagner tails. Similarly for orbits. Thus we have the following inclusions:

$$\mathcal{IEO}(h^m) \subseteq \mathcal{EO}(h^m) \subseteq \mathcal{WO}(h^m) \subseteq \mathcal{RO}(h^m). \tag{2.1}$$

Definition 2.5 (Remnant). Let $h : G \to G$ be a homomorphism. Given a generator $a \in \mathcal{G}$, $h(a)$ is said to *have remnant* if there is a subword w of $h(a)$ such that for every $b \in \mathcal{G}^{\pm} - \{A\}$ the subword w does not cancel from $h(a)$ in the products $h(a)h(b)$ and $h(b)h(a)$. The longest such subword, R_a, is defined to be the *remnant* of $h(a)$. The *homomorphism h has remnant* if

$h(a)$ has remnant for each generator $a \in \mathcal{G}$. Let $s := \min\{|R_a| : a \in \mathcal{G}\}$. If $s \geq 1$ we say that h *has remnant* s.

We remind the reader that for maps with remnant, the Wagner tails completely determine the $N(f^m)$ (see [10]). However as seen in [2] this is not enough to determine either $NP_n(f)$ or $N\Phi_n(f)$. The tools given in [2] allow us to determine if a Wagner orbit is essential or not. If P^m is not a Wagner orbit, it is automatically inessential.

The next lemma is a summary of results from [2]. Its proof needed the extension of Wagner's algorithm [10], proved in [2].

Lemma 2.6 ([2]). *If h has remnant greater than or equal to 2, then for each $m \mid n$ the determination of each of the sets $\mathcal{IEO}(h^m)$, $\mathcal{EO}(h^m)$ and $\mathcal{WO}(h^m)$ is algorithmic.* ■

3. The definition and structure of $\mathcal{U}(h,n)$. In this section we give the definition of the universal graph $\mathcal{U}(h,n)$, and show that it has the structure of a disjoint union of components indexed by $\mathcal{WO}(h^n)$. This fact will allow us to construct $\mathcal{U}(h,n)$ component by component starting at the top. We will assume throughout that we have fixed G, h and n, and that h has remnant at least 2. We start with the definition of the support of a collection of orbits.

Definition 3.1 (Support). Let A be a set of Reidemeister orbits at various levels, that is, let $A \subseteq \bigcup_{m|n} \mathcal{RO}(h^m)$. Let $\mathcal{B}(A)$ be the union of A together with the set of all orbits that boost to an orbit in A. The *support* $\mathcal{S}(A)$ of A is the transitive directed graph that has $\mathcal{B}(A)$ as the set of nodes, and which contains an edge from an orbit P^k at level k to an orbit O^m at level m if and only if $\iota_{k,m}(P^k) = O^m$.

Since it is convenient to regard an orbit as reducing to itself, we automatically include the identity edges in $\mathcal{S}(A)$. In the illustration in the introduction, the support $\mathcal{S}(A^4)$ of A^4 would be complete after the skeletons $\mathrm{Sk}(\mathcal{S}(B^2))$, $\mathrm{Sk}(\mathcal{S}(C^2))$ and $\mathrm{Sk}(\mathcal{S}(D^2))$ are merged into $\mathrm{Sk}(\mathcal{S}(A^4))$. We did not, however, show the identity edges there. With some extra details, our algorithm essentially generalizes that illustration.

The easy proof of the following lemma is left to the reader.

Lemma 3.2. *Let $B \subseteq A \subseteq \bigcup_{m|n} \mathcal{RO}(h^m)$. Then $\mathcal{S}(B)$ is a subgraph of $\mathcal{S}(A)$.* ■

Definition 3.3 (The universal graph). For fixed n, the *universal graph* $\mathcal{U}(h,n)$ is defined to be the support $\mathcal{S}(\mathcal{WO}(h^n))$ of $\mathcal{WO}(h^n)$.

We cannot take $\mathcal{U}(h,n)$ to be the support of all orbits at level n, since this set is infinite. It is enough to consider only Wagner orbits at this level.

We will, however, need to consider all reductions of Wagner orbits at each of the lower levels. This will include many non-Wagner, and in particular empty, orbits.

Proposition 3.4 (Components of $\mathcal{U}(h,n)$). *The universal graph $\mathcal{U}(h,n)$ is the disjoint union of connected components indexed by $\mathcal{WO}(h^n)$. In particular, two nodes O_1^q and O_2^r are in the same component of $\mathcal{U}(h,n)$ if and only if $\iota_{q,n}(O_1^q) = \iota_{r,n}(O_2^r)$ ($\in \mathcal{WO}(h^n)$). Thus*

$$\mathcal{U}(h,n) = \bigsqcup_{O^n \in \mathcal{WO}(h^n)} \mathcal{S}(O^n)$$

is a disjoint union. Furthermore $\mathcal{S}(\bigcup_{m|n} \mathcal{EO}(h^m)) \subseteq \mathcal{U}(h,n)$.

Remarks 3.5. (i) Another option for our universal graph would be to take $\mathcal{S}(\bigcup_{m|n} \mathcal{EO}(h^m))$ rather than $\mathcal{S}(\mathcal{WO}(h^n))$. Both of these graphs satisfy the fundamental properties of Theorem 4.8 in the next section. However, because not every orbit in $\mathcal{WO}(h^n)$ need be essential, the graph $\mathcal{S}(\bigcup_{m|n} \mathcal{EO}(h^m))$ may not contain all of $\mathcal{WO}(h^n)$. When this happens, the components of $\mathcal{S}(\bigcup_{m|n} \mathcal{EO}(h^m))$ would be more complex than those of $\mathcal{U}(h,n)$, and we would not be able to use Proposition 3.4 to construct our universal graph componentwise from the top down. Though likely more efficient, it would make the algorithm more complex to describe. See also the introductory remarks to Section 6.3.

(ii) Part of what we did in [2] was essentially to create a subgraph of $\mathcal{U}(h,n)$ from the geometry of our special representative. In general this will be a proper subgraph of $\mathcal{U}(h,n)$ (unless, for example, f happens to be essentially reducible [3, Definition 4.1]). Though it does in fact contain all the essential irreducible orbits, it is not big enough even for the computation of $NP_n(f)$. This is because it fails to include reductions to empty orbits, which of course are not detectable in the Reidemeister trace. Neither is the geometric graph big enough for the computation of $N\Phi_n(f)$. To attempt to use it for its computation is to make a similar mistake to the one mentioned in Remark 2.4.

Proof of Proposition 3.4. Clearly for each $O^n \in \mathcal{WO}(h^n)$ we have $\{O^n\} \subseteq \mathcal{WO}(h^n)$, so then $\mathcal{S}(O^n) := \mathcal{S}(\{O^n\}) \subseteq \mathcal{S}(\mathcal{WO}(h^n))$ for all $O^n \in \mathcal{WO}(h^n)$ by Lemma 3.2. So the (ordinary) union of the $\mathcal{S}(O^n)$ is contained in $\mathcal{U}(h,n)$.

Now let Q^k and P^m be nodes in $\mathcal{U}(h,n)$. Clearly if $\iota_{k,n}(Q^k) = \iota_{m,n}(P^m)$ ($= O^n$ say), then there is an edge from Q^k to O^n and an edge from P^m to O^n, so Q^k and P^m are in the same component of $\mathcal{U}(h,n)$ by definition. Conversely, suppose Q^k and P^m are in the same component of $\mathcal{U}(h,n)$. Since $\mathcal{U}(h,n)$ is a directed graph, and all edges go from lower levels to higher levels, then either both Q^k and P^m boost to a third orbit, T^s say, or one of them boosts to the other. In any case (considering identity boosts) we have $\iota_{k,n}(Q^k) = \iota_{s,n}\iota_{k,s}(Q^k) = \iota_{s,n}(T^s) = \iota_{s,n}\iota_{m,s}(P^m) = \iota_{m,n}(P^m)$ and so Q^k and P^m are in the component determined by $\iota_{s,n}(T^s)$. ■

4. Word length arguments, the Twisted Conjugacy Theorem, data labels, and more properties of $\mathcal{U}(h,n)$.

In this section we make the modifications to the word length arguments mentioned in the introduction. In the process, we solve the twisted conjugacy problem for the class of maps of remnant at least 2, and exhibit more properties of $\mathcal{U}(h,n)$.

For the computation of $NP_n(f)$ in [2], we used an extended form of Wagner's algorithm (on the unreduced form of iterates) to find Reidemeister equivalences between Wagner elements of h, and to determine the essentiality or not of these classes. Other techniques were used to find orbits. In order to complete the computation of $NP_n(f)$ we then needed only to determine, for each essential orbit $O^m \in \mathcal{EO}(h^m)$, if there was a reduction of O^m or not. In order to do this we used word length arguments on the readily discernible shortest Wagner tail that represented the given orbit.

In the proof that the computation of $N\Phi_n(f)$ is algorithmic we need to do much more than this. In fact, we need to compute all possible reductions of a multitude of orbits, including many not represented by Wagner tails (i.e. not visible in the Reidemeister trace). We start by defining a procedure which determines all reductions of a given word at the group level. We do this by extending the result of [2, Theorem 5.4].

Lemma 4.1 (Finite Reduction Lemma—Words). *Let h have remnant at least 2, $k \mid m$ and u and v be words such that $\iota_{k,m}(u) = v$. Then $|u| \le |v|$, and the set $\mathrm{RED}(v,m,k) = \{u : \iota_{k,m}(u) = v\}$ of all such words is finite. The procedure which tries all words u with $|u| \le |v|$ in the equation $\iota_{k,m}(u) = v$ is algorithmic.*

Proof. For any u' with $\iota_{k,m}(u') = v$ we know from [2, Theorem 5.4] that $|u'| \le |v|$. But $|v|$ and G are finite, so in any case there are only a finite number of words $u' \in G$ with $|u'| \le |v|$. Thus we search among all words u with $|u| \le |v|$ for those u which satisfy $\iota_{k,m}(u) = v$. This will determine $\mathrm{RED}(v,m,k)$. ■

Let $g = h^m$. In the definition below, which we recall from [2], we assume that all words $g(a_i)$ are written in *reduced* form. So

$$\mathrm{MR}(m) := \min\{|R_i| : R_i \text{ is the remnant of } g(a_i)\}$$

is the minimum length of the remnants R_i of $g(a_i)$. It seems worth remarking that for most homomorphisms the minimum remnant of the mth iterate increases with m.

Theorem 4.2 (Twisted Conjugacy Theorem [11]). *For homomorphisms h with remnant at least 2, the twisted conjugacy problem for all iterates of h is reducible to a finite search.*

More precisely, if G, h are as in this paper, and if $\mathrm{MR}(m) = q \geq 2$, then for $u, v \in G$, $[u]^m = [v]^m$ if and only if there is a word $w \in G$ for which

$$|w| \leq \frac{|u| + |v|}{q-1} \quad \textit{and} \quad wuh^m(w^{-1}) = v.$$

In particular, if $u \neq v$ and $1 > \frac{|u|+|v|}{q-1}$, then $[u]^m \neq [v]^m$.

Proof. Let $\mathrm{MR}(m) = q \geq 2$. By the definition of remnant, for any $w \in G$ we have $|h^m(w)| \geq q \cdot |w|$. Let $u, v \in G$ be such that $u \neq v$. Suppose that $w \in G$ is such that $wvh^m(w^{-1}) = u$. Then $|u| = |wvh^m(w^{-1})| \geq (q-1)|w| - |v|$. Thus $|w| \leq \frac{|u|+|v|}{q-1}$. ■

Theorem 4.2 allows us to see (sometimes trivially) that a number of computations and procedures needed in the computation of $N\Phi_n(f)$ are algorithmic. We start with

Corollary 4.3 (Equiv). *Let $w_1, w_2 \in G$, $m \,|\, n$ and* $\mathrm{MR}(m) \geq 2$. *The procedure* $\mathrm{Equiv}(w_1, w_2, m)$, *which returns "yes" if there is a word $z \in G$ with $w_1 = zw_2h^m(z^{-1})$ and "no" otherwise, is algorithmic.* ■

It should be clear that we need to include the m in the procedure $\mathrm{Equiv}(w_1, w_2, m)$, since w_1 and w_2 may well be equivalent at one level but not at another.

We introduce the concept of "data labels" which we attach to each orbit to give a "weight" to $\mathcal{U}(h, n)$. These labels are extremely useful for our algorithm.

Definition 4.4 (Data labels). A *data label* for $P^m \in \mathcal{RO}(h^m)$ is a five-tuple $\mathcal{D}(P^m) = (\mathcal{D}_p(P^m), \mathcal{D}_r(P^m), \mathcal{D}_\ell(P^m), \mathcal{D}_\mathcal{E}(P^m), \mathcal{D}_d(P^m))$, where $\mathcal{D}_p(P^m) = m$ is the **p**eriod (level) of P^m (i.e. $P^m \in \mathcal{RO}(h^m)$), $\mathcal{D}_r(P^m)$ is a word that **r**epresents P^m (i.e. if $\mathcal{D}_r(P^m) = w$, then $P^m = \langle [w]^m \rangle$), $\mathcal{D}_\ell(P^m)$ is the algebraic **l**ength of P^m, $\mathcal{D}_\mathcal{E}(P^m) \in \{0, 1\}$ denotes that P^m is **e**ssential if $\mathcal{D}_\mathcal{E}(P^m) = 1$, or is 0 otherwise, and finally $\mathcal{D}_d(P^m)$ is the **d**epth of P^m.

Since if $P^m \notin \mathcal{WO}(h^n)$ then P^m represents an empty orbit (see [2, p. 104]), for such a P^m we must have $\mathcal{D}_\mathcal{E}(P^m) = 0$.

Corollary 4.5 ($\mathrm{GOrb}(z, m)$ and orbit length $\mathcal{D}_\ell(\langle [z]^m \rangle)$). *Let $z \in G$. There are algorithmic procedures, described below, for determining the length*

[11] Chris Staeker has some analogous results for coincidences, which were discovered independently. He apparently improved his result after seeing our methods. He explains all this in [9].

$\ell = \mathcal{D}_\ell(\langle [z]^m \rangle)$ of $\langle [z]^m \rangle$ and for $\mathrm{GOrb}(z, m) := \{z, f(z), \dots, f^{\ell-1}(z)\} \subseteq G$ *at level m. The level of course is indicated by $\mathcal{D}_p(P^m)$.*

Note that we are not claiming that $f^\ell(z) = z$, only that $f^\ell(z)$ is Nielsen equivalent to z at level m. Of course $\ell \,|\, m$ [4, Lemma 1.13].

Proof. We compute ℓ as the smallest positive integer for which the procedure $\mathrm{Equiv}(z, f^\ell(z), m)$ returns yes (Corollary 4.3). The algorithmic determination of $\mathrm{GOrb}(z, m)$ follows trivially. ■

Corollary 4.5 and the definitions easily show that the determination of when two orbits are equal is also algorithmic. It is convenient to name this as a procedure.

Corollary 4.6 (Orbit identification: OrbIdn). *Let $P_1, P_2 \in \mathcal{RO}(h^m)$ with $w_1 = \mathcal{D}_r(P_1)$ and $w_2 = \mathcal{D}_r(P_2)$ chosen. Then $P_1 = P_2$ if and only if there is a $u \in$* $\mathrm{GOrb}(w_1, k)$ *with $w_2 \sim_k u$, and this can be determined algorithmically using the procedure above.* ■

We can now give an orbit version of Lemma 4.1. For each $O^m \in \mathcal{RO}(h^m)$, define $\mathrm{OrbRedn}(O^m, k, m) := \{P^k \in \mathcal{RO}(h^k) : \iota_{k,m}(P^k) = O^m\}$. So the set $\mathrm{OrbRedn}(O^m, k, m)$ is the set of all orbits at level k that boost to O^m.

Corollary 4.7 (Orbit reduction). *Let* $\mathrm{MR}(1) \geq 2$, *$O^m \in \mathcal{RO}(h^m)$ and $u = \mathcal{D}_r(O^m)$ be given or chosen. Then* $\mathrm{OrbRedn}(O^m, k, m)$ *is finite and there are algorithmic procedures for determining it (provided below), and $\mathcal{D}_r(P^k)$ and $\mathcal{D}_\ell(P^k)$ for each $P^k \in$* $\mathrm{OrbRedn}(O^m, k, m)$ *can be determined algorithmically using* $\mathrm{RED}(u, m, k)$ (*Lemma* 4.1).

Proof. Use the algorithm from Lemma 4.1 to determine the set $\mathrm{RED}(u, m, k) = \{v \in G : \iota_{k,m}(v) = u\}$. We show first that any $P^k \in \mathrm{OrbRedn}(O^m, k, m)$ is represented by some element of $\mathrm{RED}(u, m, k)$. Thus $\#(\mathrm{OrbRedn}(O^m, k, m)) \leq \#(\mathrm{RED}(u, m, k))$ and so $\mathrm{OrbRedn}(O^m, k, m)$ is finite. So suppose $P^k = \langle [z]^k \rangle$ for some z (not necessarily in $\mathrm{RED}(u, m, k)$). By Lemma 2.1 there is a word v with $v \sim_k z$ and $\iota_{k,m}(v) = u$. Clearly $v \in \mathrm{RED}(u, k, m)$, and $P^k = \langle [z]^k \rangle = \langle [v]^k \rangle$.

Next if $k = m$ then $\mathrm{OrbRedn}(O^m, k, m) := \{O^m\}$, so we may assume $k < m$ without loss. We give the following pseudo code to determine $\mathrm{OrbRedn}(O^m, k, m)$. The above shows that this process determines the whole of $\mathrm{OrbRedn}(O^m, k, m)$.

```
Step 1. Set $\mathrm{OrbRedn}(O^m, k, m) := \emptyset$.
Step 2. If $\mathrm{RED}(u, m, k) = \emptyset$, print $\mathrm{OrbRedn}(O^m, k, m)$, else choose $v \in \mathrm{RED}(u, m, k)$. Reset $\mathrm{OrbRedn}(O^m, k, m) := \mathrm{OrbRedn}(O^m, k, m) \cup \{\langle [v]^k \rangle\}$.
Step 3. Set $\mathcal{D}_r(\langle [v]^k \rangle) = v$.
Step 4. Compute $\mathcal{D}_\ell := \ell(\langle [v]^k \rangle)$, and $\mathrm{GOrb}(v, k)$ (Corollary 4.5).
```

Step 5. Determine $\mathcal{O}(v) := \{v' \in \mathrm{RED}(u,m,k) : \exists z \in \mathrm{GOrb}(v,k)$ with $v' \sim_k z\}$. Reset $\mathrm{RED}(u,m,k) := \mathrm{RED}(u,m,k) - \mathcal{O}(v)$.

Step 6. Return to step 2. ∎

Theorem 4.8 (More properties of $\mathcal{U}(h,n)$). *Let G, f and n be as given earlier, and* $\mathrm{MR}(1) \geq 2$. *Then the directed graph $\mathcal{U}(h,n)$ is finite, and each minimal set of n-representatives for f is a subset of $\mathcal{N}(\mathcal{U}(h,n))$, the set of nodes of $\mathcal{U}(h,n)$.*

Proof. For each $O^n \in \mathcal{WO}(h^n)$ we have

$$\mathcal{N}(\mathcal{S}(O^n)) = \bigcup_{m|n} \mathrm{OrbRedn}(O^n, m, n),$$

by definition. By Corollary 4.7 each $\mathrm{OrbRedn}(O^n, m, n)$ is finite, and from [2] we know that $\mathcal{WO}(h^n)$ is finite. The finiteness of $\mathcal{U}(h,n)$ follows.

For the second part, we prove that if B is a set of n-representatives, then it contains a (not necessarily proper) subset $C \subset \mathcal{N}(\mathcal{U}(h,n))$ that is also a set of n-representatives. Note first that for any essential O^m at level m we have $O^m \in \mathcal{WO}(h^m)$, and from [2, Lemma 3.5] we have $\iota_{m,n}(O^m) \in \mathcal{WO}(h^n)$. So let B be a set of n-representatives for h. Then for each essential orbit O^m there is an orbit $P^q \in B$ that boosts to O^m. Since O^m boosts to some orbit (O^n say) in $\mathcal{WO}(h^n)$, then P^q boosts to O^n too. Since $\mathcal{U}(h,n)$ contains all reductions of $\mathcal{WO}(h^n)$, by Definition 3.1 for such a P^q we must have $P^q \in \mathcal{N}(\mathcal{U}(h,n))$. This means that $C = B \cap \mathcal{N}(\mathcal{U}(h,n))$ will also be a set of n-representatives for h. Clearly for any such C we have $\mathrm{height}(C) \leq \mathrm{height}(B)$. That is, any set B of n representatives has a subset C of n representatives of height at least as small as that of B and that is contained in $\mathcal{N}(\mathcal{U}(h,n))$. In particular, any minimal set of n-representatives of h is a subset of $\mathcal{N}(\mathcal{U}(h,n))$. ∎

It seems worth remarking that the point of the second part of the proof of Theorem 4.8 is that a set of n-representatives can contain orbits which are not nodes of $\mathcal{U}(h,n)$. The proof shows such orbits are completely redundant. Since they are allowed by the definition, it means that the collection of all sets of n-representatives on these spaces is infinite!

5. The algorithm for $N\Phi_n(f)$. We are now ready to give our algorithm for $N\Phi_n(h)$. As will be obvious from the proof of Corollary 5.2, the main step (and the main work of this section) is in the following theorem.

Theorem 5.1 (The algorithm for $(\mathcal{U}(h,n), \mathcal{D})$). *Suppose that* $\mathrm{MR}(1) \geq 2$. *Then there are algorithms, which we provide, for* (a) *the construction of $\mathcal{U}(h,n)$, and* (b) *the assignment of a data label $\mathcal{D}$ for each $P^m \in \mathcal{N}(\mathcal{U}(h,n))$.*

Corollary 5.2 (The algorithm for $N\Phi_n(h)$). *Let G, h and n be given as in this paper, with* $\mathrm{MR}(1) \geq 2$. *Then there is an algorithm, provided below, for the determination of $N\Phi_n(h)$.*

Proof of Corollary 5.2. The following steps give the result:

Step 1. Determine $\mathcal{U}(h,n)$ and $\mathcal{D}$ as in Theorem 5.1.
Step 2. Enumerate all subsets of $\mathcal{N}(\mathcal{U}(h,n))$. Using the data labels, discard those subsets that are not sets of n-representatives.
Step 3. Using the data labels, compute the height of each set of n-representatives determined in step 2 and determine the minimum height.
Step 4. Return the minimum height determined in step 3. This is $N\Phi_n(h)$. ∎

The rest of the section is devoted to the proof of Theorem 5.1. Our plan is to construct the skeleton (Definition 5.3) of each orbit O^n at level n, and then inductively construct the support of O^n from the bottom up by merging the skeletons of each of the orbits at each level. While we are doing this, we simultaneously add the depth component of the data label of each orbit. The other components of the data labels are added at appropriate times.

Definition 5.3 (Skeleton). Let $O^m \in \mathcal{RO}(h^m)$. *The skeleton* $\mathrm{Sk}(\mathcal{S}(O^m))$ of $\mathcal{S}(O^m)$ is the subgraph of $\mathcal{S}(O^m)$ which consists of all the nodes P^k of $\mathcal{S}(O^m)$ (for all $k \mid m$) together with exactly one edge from P^k to O^m.

As illustrated in the introduction, $\mathrm{Sk}(\mathcal{S}(A^4))$ contains all of the arrows from nodes at levels 1 and 2 to A^4, but no arrows from nodes at levels 1 to 2. These were filled in later as we merged the skeletons of the nodes of $\mathrm{Sk}(\mathcal{S}(A^4))$ at level 2.

Lemma 5.4. *For $O^m \in \mathcal{RO}(h^m)$, there are algorithms, given below, for the determination of* (a) $\mathrm{Sk}(\mathcal{S}(O^m))$ *from a given or chosen $u := \mathcal{D}_r(O^m)$,* (b) *a data label $\mathcal{D}(O^m)$ of O^m, and* (c) *a canonical choice of $\mathcal{D}_r(P^k)$ for every $P^k \in \mathcal{N}(\mathcal{S}(O^m))$ $(\subseteq \mathcal{N}(\mathrm{Sk}(\mathcal{S}(O^m))))$.*

In practice the "given or chosen u" will either be a Wagner tail of smallest length representing O^m or, if not available, a reduction of such a Wagner tail.

Proof. Set $\mathcal{D}_p(O^m) = m$ and $\mathcal{D}_r(O^m) = u$. We define $\mathrm{Sk}(\mathcal{S}(O^m))$ inductively starting with $\mathrm{Sk}(\mathcal{S}(O^m)) = \{O^m\}$. Let $1 = d_1 < \cdots < d_{\nu(m)} = m$ be the ordered list of positive divisors of m, where $\nu(m)$ is the number of such divisors.

For each $j = 1, \ldots, \nu(m) - 1$ determine $\mathrm{OrbRedn}(O^m, d_j, m)$ by Corollary 4.7, using $u = \mathcal{D}_r(O^m)$. If $\mathcal{D}_d(O^m)$ is determined, proceed to the next step (below). Else if $\mathrm{OrbRedn}(O^m, d_j, m) \neq \emptyset$, set $\mathcal{D}_d(O^m) = d_j$, and then proceed to the next step.

Next, for each orbit $P^{d_j} \in \mathrm{OrbRedn}(O^m, d_j, m)$ add the node P^{d_j} and an arrow from P^{d_j} to O^m to the inductively constructed $\mathrm{Sk}(\mathcal{S}(O^m))$. If $j = \nu(m) - 1$, record $\mathrm{Sk}(\mathcal{S}(O^m))$ and proceed to the next step, else proceed to the next j.

Determine the orbit length $\mathcal{D}_\ell(O^m)$ from $\mathcal{D}_r(O^m)$ using Corollary 4.5.

To determine $\mathcal{D}_{\mathcal{E}}(O^m)$ (essentiality), note first that the determination of $\mathcal{EO}(h^m)$ is algorithmic by Lemma 2.6. We then simply use OrbIdn (Corollary 4.6) to determine if O^m is in this set. If it is, set $\mathcal{D}_{\mathcal{E}}(O^m) = 1$, otherwise it is 0. ■

We can of course apply Lemma 5.4 to each $O^n \in \mathcal{WO}(h^n)$ to obtain $\mathrm{Sk}(\mathcal{S}(O^n))$, and this already has all the nodes of $\mathcal{S}(O^n)$. As was illustrated in the introduction, what we do not have is all the arrows. We fill in the missing arrows by using the following procedure inductively. We remark that the $\mathcal{D}_d(P^m)$ can be determined efficiently here.

LEMMA 5.5 (Merge). *Let $O^n \in \mathcal{WO}(h^n)$, and suppose that* $\mathrm{Sk}(\mathcal{S}(O^n))$ *and $\mathcal{D}(O^n)$ have been constructed and recorded together with $\mathcal{D}_r(P^k)$ for each $P^k \in \mathcal{N}(\mathrm{Sk}(O^n))$ computed from $w = \mathcal{D}_r(O^n)$ all as in Lemma* 5.4. *Let $P^m \in \mathcal{N}(\mathrm{Sk}(\mathcal{S}(O^n)))$ and $\mathcal{A}$ be a graph with* $\mathrm{Sk}(\mathcal{S}(O^n)) \subseteq \mathcal{A} \subseteq \mathcal{S}(O^n)$. *The following steps constitute an algorithm for the determination of $\mathcal{D}_d(P^m)$ and* $\mathrm{Merge}(\mathcal{A}, \mathrm{Sk}(\mathcal{S}(P^m)))$.

Step 1. Set $\mathrm{Merge}(\mathcal{A}, \mathrm{Sk}(\mathcal{S}(P^m))) = \mathcal{A}$.

Step 2. For each $k \mid m$ use Lemma 4.1 to construct $\mathrm{RED}(u, k, m)$, where $u = \mathcal{D}_r(P^m)$, and determine $\mathcal{D}_d(P^m)$ as the smallest k for which $\mathrm{RED}(u, k, m) \neq \emptyset$.

Step 3. Choose a $k \mid m$. If $RED(u, k, m) = \emptyset$, go to step 7, else go to step 4.

Step 4. Choose a $v \in \mathrm{RED}(u, k, m)$, and identify $\langle [v]^k \rangle$ with the corresponding $Q^k \in \mathcal{N}(\mathrm{Sk}(\mathcal{S}(O^n)))$ (Corollary 4.6), and reset $\mathrm{Merge}(\mathcal{A}, \mathrm{Sk}(\mathcal{S}(P^m)))$ by adding an arrow from Q^k to P^m.

Step 5. Determine $\mathcal{O}(v)$ defined in step 5 of Corollary 4.7. Reset $\mathrm{RED}(u, m, k) := \mathrm{RED}(u, m, k) - \mathcal{O}(v)$.

Step 6. If $\mathrm{RED}(u, k, m) = \emptyset$, go to step 7, else return to step 4.

Step 7. If all k are exhausted, return $\mathrm{Merge}(\mathcal{A}, \mathrm{Sk}(\mathcal{S}(P^m)))$, else choose a new k in step 3.

By way of illustration, note that there are no arrows from level 1 to 2, and none from level 1 to 3 in $\mathrm{Merge}(\mathrm{Sk}(\mathcal{S}(O^{12})), \mathrm{Sk}(\mathcal{S}(P^6)))$. In fact to compute $\mathcal{S}(O^{12})$, we need merge the $\mathrm{Sk}(\mathcal{S}(Q^k))$ for all $\mathcal{Q}^k \in \mathcal{N}(\mathrm{Sk}(O^{12}))$. As the proof of Theorem 5.1 (next) shows, we do this divisor by divisor starting with the smallest and moving to the largest.

Proof of Theorem 5.1 (The algorithm for $\mathcal{U}(h,n)$). Using Lemma 2.6 and Corollary 4.5 determine the set $\mathcal{WO}(h^n)$, together with data label components $\mathcal{D}_p(O^n) = n$, $\mathcal{D}_r(O^n)$, $\mathcal{D}_\ell(O^n)$ and $\mathcal{D}_\mathcal{E}(O^n)$ for each $O^n \in \mathcal{WO}(h^n)$.

Let $1 = d_1 < \cdots < d_{\nu(n)} = n$ be the ordered list of positive divisors of n. For each $O^n \in \mathcal{WO}(h^n)$ determine $\mathcal{S}(O^n)$ and the data labels for each $P^k \in \mathcal{N}(\mathcal{S}(O^n))$ according to the following steps which are algorithmic as indicated.

```
Step 1. Compute $\mathrm{Sk}(\mathcal{S}(O^n))$ and $\mathcal{D}_d(O^n)$, together with the data label
        components $\mathcal{D}_p(P^k)$ and $\mathcal{D}_r(P^k)$ for each $P^k \in \mathcal{N}(\mathrm{Sk}(\mathcal{S}(O^n)))$
        $(= \mathcal{N}(\mathcal{S}(O^n)))$ as in Lemma 5.4.
Step 2. For each $P^k \in \mathcal{N}(\mathrm{Sk}(\mathcal{S}(O^n)))$ compute $\mathcal{D}_\ell(P^k)$ and $\mathcal{D}_\mathcal{E}(P^k)$
        using $\mathcal{D}_r(P^k)$, Corollaries 4.5 and 4.6, and Lemma 2.6.
Step 3. Set $\mathcal{S}(O^n) := \mathrm{Sk}(\mathcal{S}(O^n))$.
Step 4. Assign $\mathcal{D}_d(Q^1)$ to be 1 for any $Q^1 \in \mathcal{N}(\mathcal{S}(O^n))$ at level 1.
Step 5. For $j = 2, \ldots, \nu(n) - 1$, and for each $P^{d_j} \in \mathcal{N}(\mathcal{S}(O^n))$ at
        level $d_j$, compute $\mathrm{Sk}(\mathcal{S}(P^{d_j}))$ and $\mathcal{D}_d(P^{d_j})$ using $\mathcal{D}_r(P^{d_j})$ (Lemma 5.4).
Step 6. Compute $\mathrm{Merge}(\mathcal{S}(O^n), \mathrm{Sk}(\mathcal{S}(P^{d_j})))$ (Lemma 5.5).
Step 7. Reset $\mathcal{S}(O^n) := \mathrm{Merge}(\mathcal{S}(O^n), \mathrm{Sk}(\mathcal{S}(P^{d_j})))$.
Step 8. Continue to the next orbit in step 6 at level $d_j$. If the $P^{d_j}$ are
        exhausted at level $d_j$, proceed to step 9.
Step 9. Proceed to the next $j$ in step 5, unless $j = \nu(n)$, in which case
        go to step 10.
Step 10. Go to the next $O^n \in \mathcal{WO}(h^n)$ and return to step 1, unless the
         $O^n$ are exhausted, in which case go to step 11.
Step 11. Collect the components of each of the data labels, and set
         $\mathcal{U}(h,n)$ to be $\bigcup_{O^n \in \mathcal{WO}(h^n)} \mathcal{S}(O^n)$. Record $(\mathcal{U}(h,n), \mathcal{D})$. ∎
```

After Wagner produced her algorithm for the ordinary Nielsen number on the same spaces under consideration in this paper, various attempts have been made to produce an algorithm that does not require remnant or the like [8, 7]. This leads us to the following open (and difficult) question.

Open Question 5.6. *To what extent can the condition of remnant greater than or equal to two be improved, or eliminated, in the case of the Nielsen periodic point numbers $NP_n(f)$ and $N\Phi_n(f)$?*

6. Remarks on practicality and efficiency. In the preceding sections we have focused on simplicity of description when sketching pseudo code for our algorithm. In particular we have done this without regard to efficiency. With the growth of n, of the number of generators and of the length of the images of generators, the inefficiencies in the described pseudo code will at some stage outstrip the ability of even the largest computer to complete the necessary searches. For this reason we have sought, in this section, to outline a number of ways to make our algorithm more efficient. In particular we give some hints as to how the algorithm can perhaps be made a little more practical. We do not claim that our list is exhaustive.

We have deliberately put word length arguments first, since they can often be incorporated into the efficiencies described later.

6.1. Efficiencies in word length arguments. There are a number of (potential) improvements in terms of the efficiency of our word length arguments. Theorem 4.2 already points out that the remnant of h^m tends to increase as m increases. Thus we obviously use $\mathrm{MR}(m)$ in $\mathrm{Equiv}(w_1, w_2, m)$ in this procedure outlined in 4.3. But we can do more.

Enhanced remnant. This idea works at all levels, but we illustrate it at level 1. Consider the example where $\mathcal{G} = \{a, b\}$, so we are working on the figure eight. Suppose that $h(a) = BABa$ and $h(b) = ABABAbbaBBAb$. Then h has remnant 2, so if $|z| = t$, then of course $h(z) \geq 2t$. But note that $h(b)$ has remnant 10, and if z in reduced form contains r b's or B's, then $|h(z)| \geq 10r + 2s$, where $r + s = |z|$. And this can be used to reduce the number of words z that need to be tried in the fundamental equations $zw_1h(z^{-1}) = w_2$. In particular, in the example, if r and s are as above, then no z with $10r + 2s - |w_1| - (r + s) > |w_2|$ can have a solution. Clearly such z can be eliminated without a computer search. This idea of course can be extended to more than two generators.

"Remnant" of boosting functions. Lemma 4.1 which states that if $\iota_{k,m}(u) = v$, then $|u| \leq |v|$, is ridiculously (if not criminally) inefficient. We illustrate by boosting from level 1 to 3. Suppose that h has remnant q. Then we can define a remnant for $\iota_{1,3}$. In particular if $a \in \mathcal{G}$, then $|h(a)| \geq q$ and $|h^2(a)| = |h(h(a))| \geq q^2$ so $|h(a)h^2(a)| \geq q + q^2$. So allowing for the possibility that the whole of a cancels, we have $|\iota_{1,3}(a)| = |ah(a)h^2(a)| \geq q^2$ and we can think of $\iota_{1,3}(a)$ as having remnant q^2. In this way we can define remnant for each $\iota_{k,s}$, and use it to greatly improve the bound $|u| \leq |v|$ in Lemma 4.1, and of course places where it is used.

Enhanced remnant of boosting functions. In the same way that the remnant of individual generators can enhance the ordinary remnant, the individual remnants of the boosting function remnant can also be enhanced. Details are left to the reader.

6.2. Efficiencies using the data labels. In the process of identifying orbits in Merge, or determining depth etc., we can use the data labels.

Lemma 6.1. *Two orbits P^k and Q^m are equal if and only if all of the following hold true: $\mathcal{D}_p(P^k) = \mathcal{D}_p(Q^m)$, $\mathcal{D}_\ell(P^k) = \mathcal{D}_\ell(Q^m)$, $\mathcal{D}_{\mathcal{E}}(P^k) = \mathcal{D}_{\mathcal{E}}(Q^m)$, $\mathcal{D}_d(P^k) = \mathcal{D}_d(Q^m)$ and* OrbIdn *using $\mathcal{D}_r(P^k)$ and $\mathcal{D}_r(Q^m)$ produces a match.* ■

One way to use Lemma 6.1 would be in identifying orbits in step 4 of Merge (Lemma 5.5): sort out those orbits with the same length and depth. Only orbits with the same length and depth and level can be equal. Notice also that since length divides depth [4], if for an orbit P^m we have $\mathcal{D}_p(P^m) = \mathcal{D}_\ell(P^m)$, then P^m is irreducible.

Matching the $\mathcal{D}_r$ in the construction of the $\mathrm{Sk}(\mathcal{S}(P^m))$**,** $m \neq n$**.** If we follow the pattern for the construction of the $\mathrm{Sk}(\mathcal{S}(P^m))$ outlined in Lemma 5.4, we will need for each $k \mid m$ firstly that we determine the set $\mathrm{RED}(u, m, k)$ from some given u, secondly that we determine the set $\mathrm{OrbRedn}(O^m, k, m)$ from $\mathrm{RED}(u, m, k)$, and finally that we add the edges using Lemma 5.4. However, after we have constructed $\mathrm{Sk}(\mathcal{S}(O^n))$ we can, without word length arguments, create and merge the $\mathrm{Sk}(\mathcal{S}(P^m))$ (for $m < n$) using information present in the construction of $\mathrm{Sk}(\mathcal{S}(O^n))$.

To explain this, let $k \mid m \mid n$, and suppose that we have already created $\mathrm{Sk}(\mathcal{S}(O^n))$ as just outlined, together with the $\mathcal{D}_r$ component of the labels for each of its nodes. Then clearly $\mathrm{RED}(\mathcal{D}_r(P^m), k, m) \subset \mathrm{RED}(\mathcal{D}_r(O^n), k, n)$. From the proof of Corollary 4.7 we observe that $\mathrm{RED}(\mathcal{D}_r(O^n), k, n)$ is the disjoint union of subsets of the form $\mathcal{O}(\mathcal{D}_r(Q^k))$. It follows that the set $\mathrm{RED}(\mathcal{D}_r(P^m), k, m)$ is the disjoint union of subsets of the form $\mathcal{O}(\mathcal{D}_r(Q^k)) \cap \mathrm{RED}(\mathcal{D}_r(P^m), k, m)$. Now some of these may be empty (representing orbits that do not boost to P^m), but the non-empty ones determine a unique orbit (node) at this level k, and this unique orbit lies in both $\mathrm{Sk}(\mathcal{S}(O^n))$ and $\mathrm{Sk}(\mathcal{S}(P^m))$. Now the $\mathcal{D}_r$ component of the label has already been assigned in $\mathrm{Sk}(\mathcal{S}(O^n))$, and we simply assign this same $\mathcal{D}_r$ component of the label to the node considered as sitting in $\mathrm{Sk}(\mathcal{S}(P^m))$. The point is that the orbit reductions of P^m can be both constructed and labeled in this way, without going through the whole process described in Lemma 5.4. What is more, when we come to merge $\mathrm{Sk}(\mathcal{S}(P^m))$, we now have the obvious strengthened form of Lemma 6.1 to assist us in the process.

6.3. Writing $N\Phi_n(f)$ as $\sum_{m|n} NP_m(f) + \min(\mathrm{height}(\mathcal{T}))$. In this subsection we give an alternative (potentially simpler) more tractable algorithmic procedure for finding $N\Phi_n(f)$. In Remark 3.5 we suggested that a possibly more efficient option for our universal graph would be to take $\mathcal{S}(\bigcup_{m|n} \mathcal{E}\mathcal{O}(h^m))$. However, the suggestion of this subsection is likely to be more tractable.

When a map satisfies the condition of essential reducibility (footnotes 3 and 4), we can deduce (see [5]) that

$$N\Phi_n(f) = \sum_{m|n} NP_m(f).$$

When this is the case, we can simply use the algorithms of [2] to compute $N\Phi_n(f)$. Actually, the above equation holds true under weaker hypothesis than that of essential reducibility (see Proposition 6.3). However, since $\bigcup_{m|n} \mathcal{IEO}(h^m)$ is always a subset of any set of n representatives, it is always the case that $N\Phi_n(f) \geq \sum_{m|n} NP_m(f)$, so it will always be true that

$$N\Phi_n(f) = \sum_{m|n} NP_m(f) + (\text{something} \geq 0),$$

and the "something" is potentially easier to compute.

We need some preliminaries. We define the *fruit* of $\bigcup_{m|n} \mathcal{IEO}(h^m)$ to be the set

$$\mathcal{F}\Big(\bigcup_{m|n} \mathcal{IEO}(h^m)\Big) := \Big\{P^m \in \bigcup_{m|n} \mathcal{EO}(h^m) : \exists Q^k \in \bigcup_{m|n} \mathcal{IEO}(h^m) \text{ with } \iota_{k,m}(Q^k) = P^m\Big\}.$$

The set $\mathcal{F}(\bigcup_{m|n} \mathcal{IEO}(h^m))$ includes $\bigcup_{m|n} \mathcal{IEO}(h^m)$ together with all essential classes that reduce to some element of $\bigcup_{m|n} \mathcal{IEO}(h^m)$. The symbol $\mathcal{C}(\mathcal{F}(\bigcup_{m|n} \mathcal{IEO}(h^m)))$ denotes the complement of $\mathcal{F}(\bigcup_{m|n} \mathcal{IEO}(h^m))$ in $\bigcup_{m|n} \mathcal{EO}(h^m)$.

A subset $\mathcal{T}$ of $\bigcup_{m|n} \mathcal{RO}(h^m)$ is said to be a *set of complement n-representatives* if each orbit in $\mathcal{C}(\mathcal{F}(\bigcup_{m|n} \mathcal{IEO}(h^m)))$ reduces to some orbit in $\mathcal{T}$. The *height* of $\mathcal{T}$ is the sum of the depths of the orbits in $\mathcal{T}$.

Definition 6.2 (The universal complement graph). The *universal complement graph* $\mathcal{UC}(h,n)$ is the support $\mathcal{S}(\mathcal{C}(\mathcal{F}(\bigcup_{m|n} \mathcal{IEO}(h^m))))$ of the graph $\mathcal{C}(\mathcal{F}(\bigcup_{m|n} \mathcal{IEO}(h^m)))$.

The easy proof of the following proposition is left to the reader.

Proposition 6.3. *The number $N\Phi_n(f)$ is equal to $\sum_{m|n} NP_m(f) +$ height$(\mathcal{T})$, where $\mathcal{T}$ is a set of complement n-representatives of minimal height. In particular*

$$N\Phi_n(f) = \sum_{m|n} NP_m(f) \quad \textit{if and only if} \quad \mathcal{C}\Big(\mathcal{F}\Big(\bigcup_{m|n} \mathcal{IEO}(h^m)\Big)\Big) = \emptyset.$$

We are now ready for our alternative algorithm.

Alternative Algorithm 6.4. Let G, h and n be given as in this paper, with $\mathrm{MR}(1) \geq 2$. The determination of $N\Phi_n(h)$ by the following steps is algorithmic.

Step 1. Determine $\bigcup_{m|n} \mathcal{EO}(h^m)$ and $\bigcup_{m|n} \mathcal{IEO}(h^m)$ using the algorithms of [2].
Step 2. Determine $\mathcal{F}(\bigcup_{m|n} \mathcal{IEO}(h^m))$ using the methods of [2].
Step 3. Determine $\mathcal{C}(\mathcal{F}(\bigcup_{m|n} \mathcal{IEO}(h^m)))$ using steps 1 and 2.
Step 4. Use the techniques of this paper, construct $(\mathcal{UC}(h,n),\mathcal{D})$ by merging the support $\mathcal{S}(P^k)$ of each P^k in $\mathcal{C}(\mathcal{F}(\bigcup_{m|n} \mathcal{IEO}(h^m)))$, while simultaneously constructing the complement data labels $\mathcal{D}$.
Step 5. As in the proof of Corollary 5.2, determine all sets $\mathcal{T}$ of complement n-representatives.
Step 6. Compute height$(\mathcal{T})$ of each set of complement n-representatives using the complement data labels. Record the minimum height min(height$(\mathcal{T})$).
Step 7. For each $m|n$ compute $NP_m(f)$ using the techniques of [2].
Step 8. Compute $N\Phi_n(f)$ as $\sum_{m|n} NP_m(f) + \min(\mathrm{height}(\mathcal{T}))$. ■

6.4. Ad hoc methods, an illustration. Even from the illustration in the introduction it was clear that our algorithm was not efficient, and that ad hoc methods could likely short circuit the lengthy inefficient procedures outlined here. In particular, the ad hoc methods outlined in [2] can also be used to identify $\mathcal{IEO}(h^m)$, $\mathcal{EO}(h^m)$ and $\mathcal{WO}(h^m)$ in Lemma 2.6, or in the alternative Algorithm 6.4. We briefly illustrate some of these things below.

Illustration 6.5. For the purpose of illustration we look at one component of a fictitious example. In this illustration (see Figure 1), we specify the Wagner orbits (the non-empty orbits to use the language of [2]), but we are not specifying essentiality (or not). We are looking at $n = 6$, and a component topped by the Wagner orbit A, which we specify has algebraic length 1. We further assume that the single orbit A contains two geometric orbits, one of length 3 and the other of length 2. These of course give rise to Wagner orbits which we label B and C at heights 3 and 2 respectively (see [2] for explanations). Since A has algebraic length 1, and reduces to both B and C, they also have algebraic length 1 at their respective levels. By specifying only this much, we leave open a wide range of possibilities for $N\Phi_n(f)$. We tabulate these possibilities below. For now, we just want to point out a number of things, to show why in the illustration we do not need to compute the whole of either $\mathcal{U}(h,n)$ or $\mathcal{CU}(h,n)$.

This initial data is then represented by the stars and the solid lines between them in our picture. A star may or may not be essential. The circles

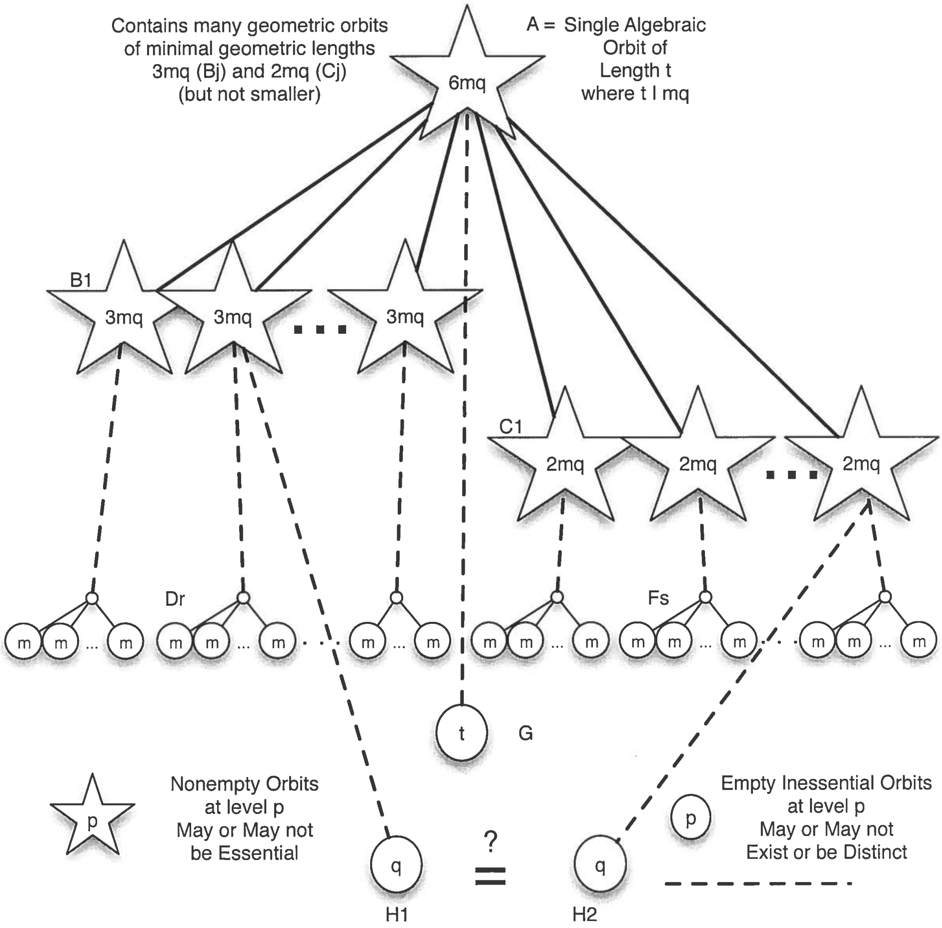


Fig. 1. Illustration 6.5

in the picture indicate inessential and possibly empty reductions. The dotted lines are edges in $\mathcal{U}(f, 6)$ which will exist when the circle orbits exist. Note that there can be multiple D's and multiple F's (and actually multiple G's too, not shown). The question mark at the bottom is to indicate the possibility that some of the D's and F's are the same. This of course would affect the computation of $N\Phi_6(f)$. The G is there because, independently of the existence of D's and F's, there could be orbits G which, while they must necessarily boost to levels 2 and 3, need not boost to any of the B's or C's.

As an indication of just how inefficient our algorithm is, we note that even allowing for multiple G's, the diagram for $\mathcal{S}(A)$ could be far larger than indicated. In particular there could be many empty orbits (not shown) at levels 2 and 3 that boost to A. We have not shown these possibilities because (with the exception that they may reduce to a G) their existence does not affect the computation of $N\Phi_n(f)$. In fact the only time we will need to know about the existence of G is when the B's and C's are irreducible. In this case we need to determine the depth of A, and finding if such G exist is the most efficient way to do this. We note, in particular, that the only time G appears in a set of complement 6-representatives is in case 11 in Figure 2.

We list the *spectrum* of possibilities for $N\Phi_6(f)$ in Figure 2. To explain our convention, we use "Es" for Essential, and "In" for Inessential. The symbols $\exists$ and $\nexists$ have the usual meaning of "exist" or "do not exist". Finally the blank spaces represent questions we need not ask, because the various options have no effect on the outcome.

Case	**A**	**B**	**C**	**D**	**F**	**G**	D, F	$\bigcup_{m\mid n} \mathcal{IEO}(f,m)$	$\min\mathcal{T}$	$N\Phi_6(f)$
1		Es	Es	$\nexists$	$\nexists$			$\{B,C\}$	$\emptyset$	5
2		Es	Es	$\exists$	$\nexists$			$\{C\}$	$\{D_i\}$	3
3		Es	Es	$\nexists$	$\exists$			$\{B\}$	$\{F_j\}$	4
4		Es	Es	$\exists$	$\exists$		$\exists D_i = F_i$	$\emptyset$	$\{D_i\}$	1
5		Es	Es	$\exists$	$\exists$		$\nexists D_i = F_i$	$\emptyset$	$\{D_i, F_j\}$	2
6		Es	In	$\nexists$				$\{B\}$	$\emptyset$	3
7		Es	In	$\exists$				$\{\emptyset\}$	$\{D_i\}$	1
8		In	Es		$\nexists$			$\{C\}$	$\emptyset$	2
9		In	Es		$\exists$			$\emptyset$	$\{F\}$	1
10	In	In	In					$\emptyset$	$\emptyset$	0
11	Es	In	In			$\exists$		$\emptyset$	$\{G\}$	1
12	Es	In	In		$\exists$			$\emptyset$	$\{F_j\}$	1
13	Es	In	In	$\exists$				$\emptyset$	$\{D_i\}$	1
14	Es	In	In	$\nexists$	$\nexists$	$\nexists$		$\emptyset$	$\{C\}$	2

Fig. 2. Illustration 6.5

We make a few observations. Note firstly that reductions of A (or not) play no role if either B or C are essential (first nine cases). Secondly, we never need to know all of the G's, and we only need to know existence when A is essential and both B and C are inessential, and not always even then (see cases 12 and 13). In particular, for this scenario, we never need to compute the whole of $\mathcal{CU}(h,6)$ before we know $N\Phi_n(f)$. In the case where A is essential and B and C are inessential, we need only know the depth of A as cases 11 to 14 show.

Finally, it seems worth pointing out that the table of values for $N\Phi_6(f)$ covers all possible options for 0 to 5. There is no need to stop there. For example, still with just one single orbit A at level 6, we could have $N\Phi_6(f) = 6$ either for a single essential irreducible A, or two essential irreducible B's. The number $N\Phi_6(f)$ can take the value 7, for example, as a single essential irreducible B together with two essential irreducible C's etc.

Remarks 6.6. (i) It should be obvious, even from our simple illustration 6.5, that there are many cases where the determination of appropriate $\mathcal{T}$'s involves a lot less work than the determination of $\mathcal{U}(h,n)$. Case 1 of that illustration and the combined example 4.1 and 4.1 (part 2) of [2] give cases where Proposition 6.3 is useful. Illustration 6.5 also shows that many times we do not need to complete either $\mathcal{U}(h,n)$ or $\mathcal{CU}(h,n)$ before we know $N\Phi_n(f)$.

(ii) It would be interesting to have a computer program that incorporated some of the efficiencies of this section and that could give reasonable length examples illustrating this paper and its predecessor.

Acknowledgements. We would like to thank the referee for extremely useful comments that led us to completely rewrite this paper in what we believe is a much more readable and comprehensible way.

References

[1] E. Hart and E. Keppelmann, *Explorations in Nielsen periodic point theory for the double torus*, Topology Appl. 95 (1999), 1–30.

[2] E. Hart, P. R. Heath and E. Keppelmann, *Algorithms for Nielsen type periodic numbers of maps with remnant on surfaces with boundary and on bouquets of circles I*, Fund. Math. 200 (2008), 101–132.

[3] P. R. Heath and E. Keppelmann, *Fibre techniques in Nielsen periodic point theory on nil and solvmanifolds. I*, Topology Appl. 76 (1997), 217–247.

[4] P. R. Heath, R. Piccinini and C. Y. You, *Nielsen-type numbers for periodic points. I*, in: Topological Fixed Point Theory and Applications (Tianjin, 1988), Lecture Notes in Math. 1411, Springer, Berlin, 1989, 88–106.

[5] P. R. Heath and C. Y. You, *Nielsen-type numbers for periodic points. II*, Topology Appl. 43 (1992), 219–236.

[6] B. Jiang, *Lectures on Nielsen Fixed Point Theory*, Contemp. Math. 14, Amer. Math Soc., Providence, RI, 1983.

[7] S. Kim, *The WYK algorithm for maps of aspherical figure-eight type finite polyhedra*, J. Pure Appl. Algebra 216 (2012), 1652–1666.

[8] S. Kim and P. Yi, *Nielsen numbers of maps of aspherical figure-eight type polyhedra*, Forum Math. 27 (2015), 1277–1307.

[9] C. Staecker, *Remnant inequalities and doubly-twisted conjugacy in free groups*, J. Pure Appl. Algebra 215 (2011), 1702–1710.

[10] J. Wagner, *An algorithm for calculating the Nielsen number on surfaces with boundary*, Trans. Amer. Math. Soc. 351 (1999), 41–62.

Evelyn L. Hart
Department of Mathematics
Colgate University
13 Oak Drive
Hamilton, NY 13346-1398, U.S.A.
E-mail: ehart@colgate.edu

Philip R. Heath
Department of Mathematics and Statistics
Memorial University of Newfoundland
St. John's, NL, Canada, A1C 5S7
E-mail: prheath@mun.ca

Edward C. Keppelmann
Department of Mathematics
University of Nevada – Reno
1664 N. Virginia Street
Reno, NV 89557-0084, U.S.A.
E-mail: keppelma@unr.edu