MINIMAL REPRESENTATIVES OF ENDOFUNCTIONS

JAMES D. MITCHELL, SUJOY MUKHERJEE, AND PETR VOJTĚCHOVSKÝ

Abstract. We present a simple and efficient algorithm for finding the lexicographically minimal representative in a given conjugacy class of the action of the symmetric group on endofunctions.

1. Introduction

An *endofunction* on X is a function $t : X \to X$. All endofunctions on X form the full transformation monoid T_X under composition. The symmetric group S_X acts on T_X by conjugation $t \mapsto sts^{-1}$. The orbits of this action will be called the S_X*-conjugacy classes* of T_X, or just *conjugacy classes of* T_X.

Endofunctions on X can be identified with functional digraphs on the vertex set X. A digraph $G = (X, E)$ is a *functional digraph* if for every $x \in X$ there is a unique $y \in X$ such that $(x, y) \in E$. Given an endofunction $t : X \to X$, we obtain a functional digraph $G(t) = (X, E)$ by setting $E = \{(x, t(x)) : x \in X\}$. Conversely, given a functional digraph $G = (X, E)$, we obtain an endofunction $t_G \in T_X$ by setting $t_G(x) = y$ if and only if $(x, y) \in E$. The two constructions are inverse to each other.

Furthermore, if $t \in T_X$, $s \in S_X$ and $t(x) = y$, then $sts^{-1}(s(x)) = st(x) = s(y)$. Hence the functional digraph $G(sts^{-1})$ is isomorphic to $G(t)$ and it is obtained from $G(t)$ by renaming every vertex x to $s(x)$. Conversely, if $G = (X, E)$ and $H = (X, F)$ are isomorphic functional digraphs on X and if $s \in S_X$ is an isomorphism $G \to H$ then $(s(x), s(y)) \in F$ if and only if $(x, y) \in E$, so $t_H(s(x)) = s(y)$ if and only if $t_G(x) = y$, that is, $t_H = st_Gs^{-1}$. Classifying endofunctions up to conjugacy is therefore equivalent to classifying (vertex-labeled) functional digraphs up to isomorphism.

From now on let $X = X_n = \{1, \dots, n\}$ be a fixed finite set, $T_X = T_n$ and $S_X = S_n$. We will identify $t \in T_n$ with the tuple $[t(1), t(2), \dots, t(n)]$ and order T_n lexicographically. If C is an S_n-conjugacy class of T_n then $t \in C$ is the *minimal representative of* C if $t \leq s$ for every $s \in C$.

In this note we present a simple and efficient algorithm for finding the minimal representative of the S_n-conjugacy class of T_n containing a given endofunction t. The complexity of the algorithm is $O(n^2)$.

2020 *Mathematics Subject Classification.* Primary: 05E18. Secondary: 05C20, 20M20.

Key words and phrases. Endofunction, mapping type, full transformation monoid, minimal representative, conjugation, lexicographic order.

S. Mukherjee supported by the American Mathematical Society and the Simons Foundation through the AMS–Simons Travel Grant. P. Vojtěchovský supported by the Simons Foundation Mathematics and Physical Sciences Collaboration Grant for Mathematicians no. 855097 and by the Enhanced Sabbatical Fund of the University of Denver.

We can transfer the linear order on endofunctions to digraphs by letting $G \leq H$ if and only if $t_G \leq t_H$. When $s \in T_n$ is given, we will often search for the minimal representative t in the conjugacy class of s by suitably relabeling the vertices of the functional digraph $G(s)$.

A subset $U \subseteq X$ of a digraph $G = (X, E)$ is (*weakly*) *connected* if for every $x, y \in U$ there is a sequence $x = z_0, z_1, \ldots, z_m = y$ of vertices of U such that for every $0 \leq i < m$ we have either $(z_i, z_{i+1}) \in E$ or $(z_{i+1}, z_i) \in E$. A *connected component* of $G = (X, E)$ is a maximal connected subset of X.

It is easy to see that every connected component U of a functional digraph G contains a unique (directed) cycle, possibly degenerated into a loop. We will denote the unique cycle of U by $\mathrm{C}(U)$ and its length by $c(U)$. Upon removing the edges of $\mathrm{C}(U)$, U becomes a union of disjoint directed trees rooted in (or, more precisely, with the sink located in) $\mathrm{C}(U)$. See the examples in Subsection 2.4 for a typical connected component of a functional digraph.

Finally, if $G = (X, E)$ is a digraph and $x \in X$, we let

$$\mathrm{N}(x) = \{y \in X : (y, x) \in E\}$$

be the set of all in-neighbors of x. For a functional digraph $G(t)$, we of course have $\mathrm{N}(x) = t^{-1}(x)$, the preimage of x under t.

1.1. **Related work.** Suppose that a group H acts on a set A and denote by a^H the orbit of $a \in A$ under the action of H. A *canonical representative* of $a \in A$ is an element $c_a \in a^H$ such that for every $a, b \in A$ we have $c_a = c_b$ if and only if $a^H = b^H$. Suppose further that $\leq$ is a total order on A. The *minimal representative* of $a \in A$ is the element $m_a \in a^H$ such that $m_a \leq b$ for all $b \in a^H$. Note that the minimal representative is a canonical representative. Generally speaking, finding minimal representatives appears to be more difficult than finding canonical representatives.

Linton [14] introduced a general algorithm for finding the minimal representative of the action of a permutation group H on k-element subsets of a set A; the algorithm has been implemented in [18]. Jefferson et al. [10, 11] presented several algorithms for finding the minimal representative in the same setting, and demonstrated on several examples that the running time of their algorithm is better than the running time of Linton's algorithm, sometimes by an order of magnitude. Linton's algorithm runs in time that is at least linear in $|A|$, and it is therefore not practical for the problem considered in this paper because we have $A = T_n$ and hence $|A| = n^n$. Nevertheless we will use the state-of-the-art implementation in [11] for a running time comparison with our algorithm and to check our algorithm for correctness.

Since endofunctions are in one-to-one correspondence with functional digraphs, our problem can be solved by graph algorithms. It is known that canonical representatives (or, better, canonical labels) of trees and planar graphs can be calculated in linear time [4]. For trees, the folklore algorithm is often attributed to Edmonds (see [3] and [1, Example 3.2]). Our tree labeling algorithm of §2.3 is similar to Edmonds's algorithm in that it labels rooted trees based on degrees of vertices, but it follows a different labeling procedure since our goal is to find the minimal representative, not just a canonical representative suitable for solving the isomorphism problem. State-of-the-art canonical labeling algorithms for graphs can be found in [13, 15, 16] but none of these tools calculates minimal representatives. It is proved in [5] that the problem of finding the minimal incidence matrix of a graph (under the action of permuting rows and columns of the incidence matrix) is NP-complete. Quoting from [5]: "We trust that the above demonstration will discourage finding lex-leading-incidence-matrices as an approach to finding canonical forms for graphs and, thereby, to graph isomorphism."

Our motivation for this work is the problem of finding the minimal representative of a groupoid in its isomorphism class. A total ordering on groupoids defined on a totally ordered set X is usually obtained by concatenating the rows of the multiplication table and ordering the resulting vectors lexicographically. But we can consider a different ordering of cells in the multiplication tables, say with the diagonal cells being considered first. The diagonal of a groupoid $(X,\cdot)$ can be seen as an endofunction on X. By finding the minimal representative of the corresponding endofunction, together with a permutation that certifies how the minimal representative has been obtained, one can quickly find an isomorphic copy of $(X,\cdot)$ with the smallest possible diagonal, an important step toward finding the minimal representative in the isomorphism class of $(X,\cdot)$.

The diagonals of groupoids were exploited by Ježek [12] while enumerating small left distributive groupoids up to isomorphism. In effect, Ježek performed the enumeration one conjugacy class representative of an endofunction at a time. There does not seem to be an efficient algorithm for generating representatives of all S_n-conjugacy classes of endofunctions on $X_n = \{1,\dots,n\}$. Such representatives are also known as *mapping types* or *mapping patterns*. Let a_n be the number of mapping types on X_n, the first ten values being $a_1 = 1$, $a_2 = 3$, $a_3 = 7$, $a_4 = 19$, $a_5 = 47$, $a_6 = 130$, $a_7 = 343$, $a_8 = 951$, $a_9 = 2615$ and $a_{10} = 7318$. A formula for a_n was first derived by Davis [6, Theorem 6]. See also [8, p. 18], where a_n has been explicitly calculated for $n \le 15$, and the OEIS sequence [17, A001372] for a_n with $n \le 1000$.

2. The algorithm

The algorithm is developed in §2.1–§2.3 together with a proof of correctness. Examples are presented in §2.4. A summary of the algorithm and its running time can be found in §2.5.

Let $t \in T_n$ be a minimal representative (that is, the minimal representative of some S_n-conjugacy class of T_n). Let $U_1,\dots,U_m$ be the connected components of the associated digraph $G(t)$.

In §2.1 we show that if $U_1 = U$ contains 1, then $\mathrm{C}(U)$ is the cycle $(1,2,\dots,c(U))$. We do not yet determine which vertex of $\mathrm{C}(U)$ is 1, but we show that $U = \{1,\dots,|U|\}$. It follows that the tuple $t = [t(1),\dots,t(n)]$ is a concatenation of the tuples $t(U_1), t(U_2),\dots,t(U_m)$ for *some* ordering of the connected components.

In §2.2 we show how the components of $G(t)$ must be ordered and hence reduce the problem to the connected case, which we handle in §2.3.

Suppose that $G(t)$ is connected. Upon removing the edges of the unique cycle $\mathrm{C}(X)$ of $G(t)$, we obtain disjoint directed trees rooted in $\mathrm{C}(X)$. We can decompose X as a disjoint union $L_0 \cup \cdots \cup L_k$, where L_i consists of the vertices of $x \in X$ at distance i from $\mathrm{C}(X)$. Let $\ell_i = |L_i|$. We already know that $L_0 = \mathrm{C}(X) = \{1,\dots,\ell_0\}$ and it easily follows that

$$L_i = \{\sum_{0\le j<i} \ell_j + 1, \dots, \sum_{0\le j\le i} \ell_j\},$$

that is, the level sets L_i consist of consecutively labeled vertices and the labels increase with the distance from the cycle. To determine the fine structure of the level sets L_i, we enhance the trees with certain structural labels, essentially keeping track of the ordered sequence of indegrees in a recursive fashion. The minimal representative is then determined by the structural labels and by the position of 1 on the cycle. To locate 1 on the cycle, we can try all possibilities or we can once again take advantage of the structural labels.

2.1. **The cycle of a connected component.**

Lemma 2.1. *Let $t \in T_n$ be a minimal representative and let U be the connected component of $G(t)$ containing 1. Then $1 \in \mathrm{C}(U)$ and the directed path from 1 is the cycle $(1, 2, \dots, c(U))$.*

Proof. Consider the vertex 1. The path p consisting of 1, $t(1)$, $t^2(1)$, etc, eventually cycles back to itself. Let $0 \le j < k$ be the smallest values such that $t^j(1) = t^k(1)$, so that the cycle of p has length $k - j$; see Figure 1.

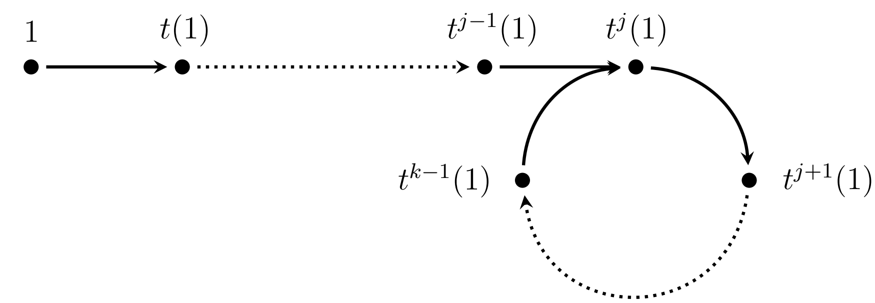

FIGURE 1. The path starting at 1.

Consider the edge $(1, t(1))$, the unique edge of $G(t)$ with source 1. If $t(1) = 1$, we are done, so suppose that $t(1) > 1$. If $t(1) > 2$ then conjugating t by the transposition $(t(1), 2)$ yields a smaller representative. Hence $t(1) = 2$. Similarly, $t^i(1) = i + 1$ for every $i < k$. Hence $t = [t(1), t(2), \dots, t(n)] = [2, 3, \dots, k, t(k), \dots, t(n)]$.

If $t(k) = 1$ then $1 \in \mathrm{C}(U)$ and we are done. We can therefore assume that $t(k) = t^k(1) = t^j(1) = j + 1 > 1$. Conjugating by any permutation s such that $s(j + 1) = 1$, $s(j + 2) = 2$, $\dots$, $s(k) = s(j + (k - j)) = k - j$ then yields a smaller representative $[2, 3, \dots, k - j, 1, \dots]$, a contradiction. □

Hence every minimal representative $t = [t(1), t(2), \dots]$ must start with $[2, 3, \dots, c(U), 1]$ for some connected component U of $G(t)$, where we understand $[2, 3, \dots, c(U)]$ to be empty if $c(U) = 1$. In terms of abstract functional digraphs, we have so far labeled the cycle of some connected component U, but it is not clear yet which connected component should be chosen and which of the $c(U)$ possible cyclic labelings of $\mathrm{C}(U)$ should be used.

Let us introduce notation for some subsets of a connected component U of a functional digraph. For $k \ge 0$, let L_k consist of all $x \in U$ at distance k from $C(U)$. Note that $L_0 = C(U)$ and $U = \bigcup_{k<\infty} L_k$. For $x \in U$, let

$$M(x) = N(x) \setminus C(U)$$

be the set of all in-neighbors of x not on the cycle. Note that $L_{k+1} = \bigcup_{x \in L_k} M(x)$, and $M(x) = N(x)$ whenever $x \notin C(U)$. (To illustrate these sets, in Figure 3, we have $L_0 = \{1, 2, 3\}$, $L_1 = \{4, 5, 6, 7\}$, $M(1) = \{4, 5\}$, and so on.)

Lemma 2.2. *Let $t \in T_n$ be a minimal representative and let U be the connected component of $G(t)$ containing 1. Then:*

(i) *$M(x)$ is an interval for every $x \in U$,*
(ii) *L_k is an interval for every k,*
(iii) *$t(i) < i$ for every $i \in U \setminus C(U)$,*
(iv) *$U = \{1, \ldots, |U|\}$.*

Proof. Let $c = c(U)$ and $d_x = |M(x)|$. By Lemma 2.1, we have $L_0 = \mathrm{C}(U) = \{1, \ldots, c\}$. If $U = L_0$, we are done, so suppose that $U \neq L_0$. Then there is a least $i \in L_0$ such that $M(i) \neq \emptyset$. Since t is minimal, we must have $M(i) = \{c+1, \ldots, c+d_i\}$. Next let $i < j \in L_0$ be least such that $M(j) \neq \emptyset$. If such j exists, we must have $M(j) = \{c+d_i+1, \ldots, c+d_i+d_j\}$, and so on. This shows that $M(x)$ is an interval for every $x \in L_0$ and that $L_1 = \bigcup_{x \in L_0} M(x)$ is an interval. If $U = L_0 \cup L_1$, we are done, otherwise we proceed to the least $i \in L_1$ such that $M(i) \neq \emptyset$, and so on. □

For any two subsets $U = \{u_1, \ldots, u_k\}$, $V = \{v_1, \ldots, v_k\}$ of X, if t_U is an endofunction on U, let $t_{U \rightsquigarrow V}$ be the endofunction on V defined by $t_{U \rightsquigarrow V}(v_i) = v_j$ if and only if $t_U(u_i) = u_j$.

Note that if t_U is a minimal representative on U (under the action of S_U) then $t_{U \rightsquigarrow V}$ is a minimal representative on V (under the action of S_V), and vice versa. We will use this formal shift in the proof of the following result and in §2.2.

Corollary 2.3. *Let $t \in T_n$ be a minimal representative and let $U_1, \ldots, U_m$ be the connected components of $G(t)$. For $1 \leq i \leq m$, let t_{U_i} be the restriction of t to U_i. Then t, seen as the tuple $[t(1), \ldots, t(n)]$, is the concatenation of the tuples $t_{U_{\pi(1)}}$, $t_{U_{\pi(2)}}$, …, $t_{U_{\pi(m)}}$, for some $\pi \in S_m$. Moreover, every U_i is an interval and $\max U_{\pi(i)} + 1 = \min U_{\pi(i+1)}$.*

Proof. We proceed by induction on the number m of connected components of X. Suppose that $1 \in U_1$, without loss of generality. If $m = 1$, we are done. Suppose that $m > 1$. Certainly, t restricts to an endofunction on $X \setminus U_1$ which has fewer connected components. Moreover, $U_1 = \{1, \ldots, |U_1|\}$ by Lemma 2.2 and hence $X \setminus U_1 = \{|U_1|+1, \ldots, |X|\}$. Modulo a formal shift of vertex labels, the induction assumption implies that $t_{X \setminus U_1}$ is a concatenation of $t_{U_{\pi(2)}}$, …, $t_{U_{\pi(m)}}$ for some $\pi \in S_{\{2,\ldots,m\}}$, as well as the rest of the claim. □

2.2. **Ordering the connected components.** Let $t \in T_n$ be a minimal representative and let $U_1, \ldots, U_m$ be the connected components of $G(t)$. By Corollary 2.3, there is $\pi \in S_m$ such that t is the concatenation of the restrictions $t_{U_{\pi(1)}}$, $t_{U_{\pi(2)}}$, …, $t_{U_{\pi(m)}}$. Moreover, $U_{\pi(1)}$, …, $U_{\pi(m)}$ are consecutive intervals. We will now determine π.

For each i, $t_{U_i \rightsquigarrow \{1,\ldots,|U_i|\}}$ is a minimal representative on $\{1, \ldots, |U_i|\}$. We can therefore transfer the endofunctions on the connected components U_i to the disjoint union

$$T_{\leq n} = \bigcup_{1 \leq k \leq n} T_k$$

and compare them there. We will need a suitable order on $T_{\leq n}$.

Consider a modification $\preceq$ of the lexicographic order in which words are ordered as usual except that when a word is a prefix of another word then the *longer* word is listed first. For instance, "antelope" comes before "ant". For obvious reasons, we will call $\preceq$ the *spelling bee dictionary order*.

Let us equip $T_{\leq n}$ with the spelling bee dictionary order $\preceq$. Note that $\preceq$ restricted to T_k gives the usual lexicographic order on T_k since all "words" in T_k have the same length.

Lemma 2.4. *Let $t \in T_n$ be a minimal representative. Let U, V be connected components of $G(t)$ and let t_U, t_V be the restrictions of t to U and V, respectively. Let $s_U = t_{U \rightsquigarrow \{1,\dots,|U|\}}$ and $s_V = t_{V \rightsquigarrow \{1,\dots,|V|\}}$. If s_U is a prefix of s_V and $|V| > |U|$ then $s_V(|U|+1) < |U|+1$.*

Proof. Suppose that s_U is a prefix of s_V and $|V| > |U|$. Then the restriction of s_V to $\{1,\dots,|U|\}$ coincides with s_U and therefore the cycles of s_U and s_V also coincide. By Lemma 2.2(iii), $s_V(|U|+1) < |U|+1$. □

Lemma 2.5. *Let $t \in T_n$ be a minimal representative and let $U_1, \dots, U_m$ be the connected components of $G(t)$ ordered so that t is the concatenation of $t_{U_1}, \dots, t_{U_m}$. Let $s_i = t_{U_i \rightsquigarrow \{1,\dots,|U_i|\}}$. Then $s_1 \preceq s_i$ for every $1 \le i \le m$.*

Proof. There is nothing to prove when $m = 1$, so we can assume that $m > 1$. Suppose that $s_i \prec s_1$ for some $i > 1$. Let $k = \min\{|U_1|, |U_i|\}$. If there is $j \le k$ such that s_i and s_1 agree on $\{1,\dots,j-1\}$ and $s_i(j) < s_1(j)$, then by renaming the elements of U_i to $\{1,\dots,|U_i|\}$ and listing U_i first, we find a smaller representative than t, a contradiction. We can thus assume that $s_i(j) = s_1(j)$ for all $j \le k$, s_1 is a proper prefix of s_i and $k = |U_1| < |U_i|$. By Lemma 2.4, $s_i(k+1) < k+1$. Now, t_{U_2} is defined on $U_2 = \{k+1,\dots,k+|U_2|\}$ and hence $t_{U_2}(k+1) \ge k+1$. Therefore, if we use U_i first and relabel suitably, we obtain a smaller representative than t, a contradiction. □

Combining Corollary 2.3 and Lemma 2.5, we have:

Corollary 2.6. *Let $t \in T_n$ be a minimal representative and let $U_1, \dots, U_m$ be the connected components of $G(t)$ ordered so that t is the concatenation of $t_{U_1}, \dots, t_{U_m}$. Let $s_i = t_{U_i \rightsquigarrow \{1,\dots,|U_i|\}}$. Then $s_1 \preceq s_2 \preceq \dots \preceq s_m$.*

2.3. **The connected case.** Suppose that t is a minimal representative and $G(t)$ is connected. Lemma 2.2 gives some restrictions on the vertices of $G(t)$. We will now establish additional restrictions.

For any directed tree in which the sink is the root, define the *structural labels* $\sigma(v) = [\sigma(v)_j : j \ge 0]$ of its vertices and the *structural labels* $\sigma(e)$ of its edges recursively as follows. If $\mathrm{N}(v) = \emptyset$ (so v is a leaf), let $\sigma(v) = [\sigma(v)_0] = [0]$, with the convention that $\sigma(v)_j$ is empty for $j > 0$. If $\mathrm{N}(v) = \{v_1,\dots,v_d\} \ne \emptyset$, suppose that $v_1,\dots,v_d$ are ordered lexicographically according to their structural labels so that $\sigma(v_1) \ge \sigma(v_2) \ge \dots \ge \sigma(v_d)$. Then $\sigma(v) = [d, \sigma_0, \sigma_1, \dots]$, where σ_j is the concatenation of $\sigma(v_1)_j, \dots, \sigma(v_d)_j$. Furthermore, label the edge (v_i, v) by $\sigma(v_i, v) = j$ if $\sigma(v_i)$ is the jth largest entry in $\{\sigma(v_1),\dots,\sigma(v_d)\}$. Note that a *tie* $\sigma(v_i, v) = \sigma(v_k, v)$ occurs whenever $\sigma(v_i) = \sigma(v_k)$. See Example 2.8, where the structural labels have been calculated for all trees rooted in the cycle.

The meaning of the structural labels is as follows. The first entry $\sigma(v)_0$ of $\sigma(v)$ is the indegree of v. The second entry $\sigma(v)_1$ is the tuple of the indegrees of the vertices in $\mathrm{N}(v) = \{v_1,\dots,v_d\}$, ordered first according to *their* indegrees, and then recursively. Note that if $\sigma(v) = \sigma(w)$ then the subtrees rooted at v and w, respectively, are isomorphic. The edge labels $\sigma(v_i, v)$ make it clear how the vertices of $\mathrm{N}(v)$ were ordered at v, which is not necessarily apparent from the vertex label at v alone. For instance, the structural vertex label $[2, 31, 0001, 0]$ in Example 2.8 shows that the vertex has indegree 2, its predecessors have indegrees 3 and 1, and so on, while the structural edge labels at the vertex specify that the predecessor with indegree 3 should be considered first.

Lemma 2.7. *Let $t \in T_n$ be a minimal representative with $G(t)$ connected. Then for every $x \in X$, the vertices of the interval $M(x)$ are ordered in an increasing order according to the structural edge labels on the edges (y, x), $y \in M(x)$, modulo the ties.*

Proof. Let $d = |M_x|$ and let $M(x) = \{a, \dots, a + d - 1\}$, cf. Lemma 2.2. Since t is minimal, a must have maximal indegree among the vertices of $M(x)$. In case of a tie, the ordered sequence of indegrees of the vertices in $M(a)$ must be maximal, and so on. This is precisely what the structural labels keep track off. □

Note that the minimal representative t is uniquely determined by the structural labels in $G(t)$ and by the position of 1 in the cycle of $G(t)$. Indeed, once 1 is placed, the labels on the cycle are determined by Lemma 2.1. The remaining vertices are then labeled consecutively one set $M(x)$ at a time, always using the smallest labeled vertex x for which $M(x)$ is nonempty and not yet labeled. The vertices of $M(x)$ are labeled as in Lemma 2.7. The ties among structural edge labels can be resolved in any way. The resulting vertex labelings of the graph $G(t)$ will be distinct, but all such labeled graphs are isomorphic and the corresponding endofunctions are the same. For instance, in Figure 3, we can interchange the labels of vertices 10 and 11, or we can interchange the labels of the two subtrees rooted at 6.

We can therefore determine t by considering all possible placements of 1 in the cycle and by keeping the minimal resulting endofunction. Alternatively, the position of 1 can be determined from the structural labels as follows. Let $(v_1, \dots, v_c)$ be the cycle and let $\sigma(v_i) = [\sigma(v_i)_0, \sigma(v_i)_1, \dots]$ be the corresponding structural vertex labels. For every starting position $1 \le i \le c$ let $s(v_i)$ be the concatenation of the sequences $s(v_i)_0$, $s(v_i)_1$, …, where $s(v_i)_j = [\sigma(v_i)_j, \sigma(v_{i+1})_j, \dots, \sigma(v_c)_j, \sigma(v_1)_j, \dots, \sigma(v_{i-1})_j]$. (Thus $s(v_i)$ is obtained by starting at v_i and going around the cycle repeatedly, collecting one entry of the structural vertex labels at a time.) Then 1 must be located at a vertex v_i with the lexicographically largest sequence $s(v_i)$.

2.4. **Examples.** In all figures below, we have removed brackets around structural vertex labels to improve legibility and we added asterisks on structural edge labels to indicate a tie.

Example 2.8. Figures 2 and 3 show how the structural labels of a connected component can be used to find the minimal representative of an endofunction. The missing structural vertex labels are $a = [2, 22, 3100, 0001, 0]$ and $b = [2, 22, 2200, 0000]$. Starting at a and going around the cycle once results in the sequence $[2, 2, 0]$ of indegrees (not counting edges in the cycle), while starting at b (resp. at the third vertex of the cycle) produces the sequence $[2, 0, 2]$ (resp. $[0, 2, 2]$). It is therefore clear in this case after just one round that 1 must be positioned at a. One of the minimal graph representatives of the component (giving rise to the unique minimal representative of the corresponding endofunction) is depicted in Figure 3. It is obtained by first labeling the cycle $(1, 2, 3)$, then the set $M(1) = \{4, 5\}$ according to the structural edge labels, then the set $M(2)$, and so on.

Example 2.9. Consider the example in Figure 4. The minimal representative $[2, 3, 1, 1, 2, 5]$ is obtained by placing 1 at $[1, 0]$. (Placing 1 at $[1, 1, 0]$ results in the larger endofunction $[2, 3, 1, 1, 3, 4]$, and placing 1 at $[0]$ yields the even larger $[2, 3, 1, 2, 3, 5]$.) Hence 1 certainly does not have to occur at the cycle vertex with the largest structural label. Even if we consider the sequences $[1, 0, 1, 1, 0, 0]$, $[1, 1, 0, 0, 1, 0]$ and $[0, 1, 0, 1, 1, 0]$ obtained by going around the cycle from different starting points and simply concatenating the structural vertex labels,

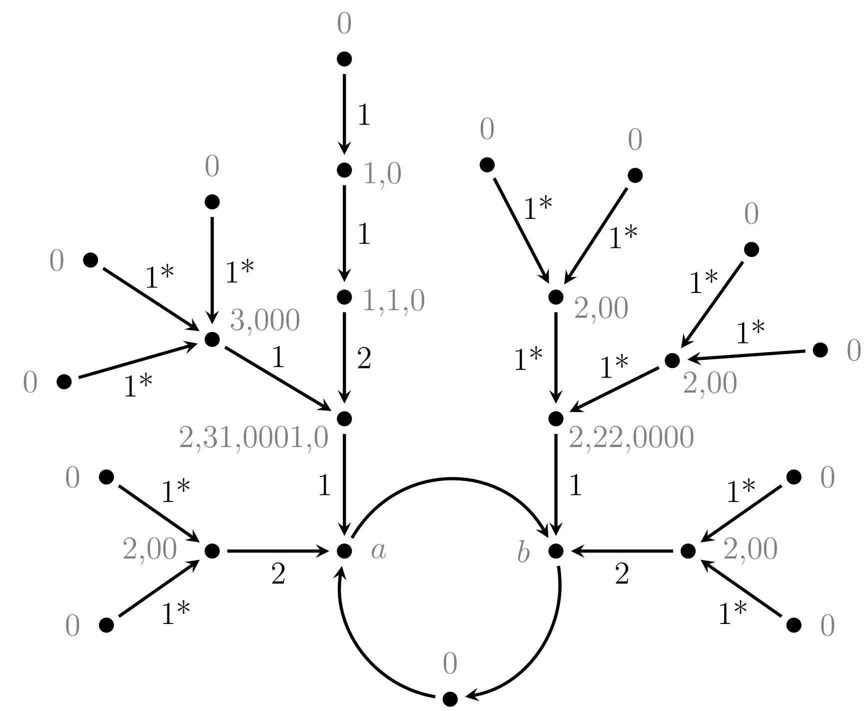

FIGURE 2. Structural labels in a connected component.

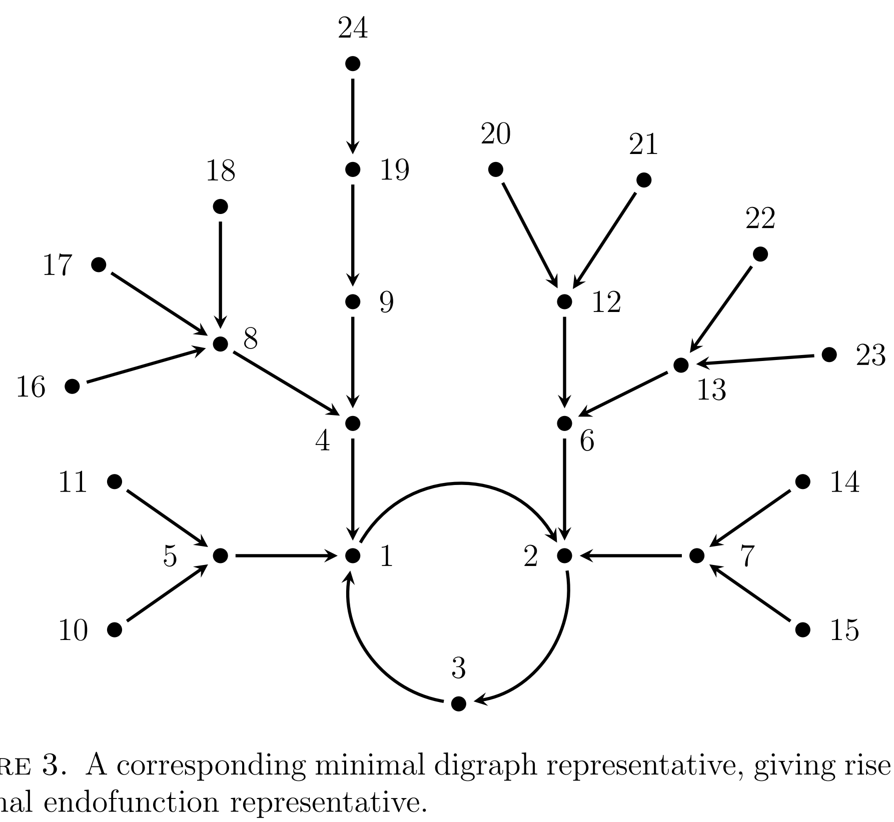

FIGURE 3. A corresponding minimal digraph representative, giving rise to the minimal endofunction representative.

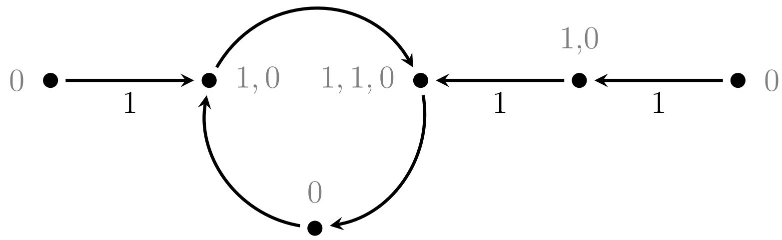

FIGURE 4. Placing vertex 1.

we observe that 1 does not have to occur at the cycle vertex with the largest sequence. But constructing the sequences by going around the cycle several times and collecting only one entry of each structural vertex label at a time (as in §2.3) yields $[1, 1, 0, 0, 1, 0]$, $[1, 0, 1, 1, 0, 0]$ and $[0, 1, 1, 0, 1, 0]$, respectively, correctly determining the placement of 1.

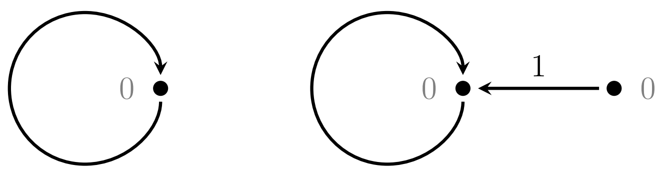

FIGURE 5. Ordering connected components.

Example 2.10. Figure 5 shows why the spelling bee dictionary ordering arises while comparing connected components. The left connected component results in the endofunction $[1]$, while the right connected component results in the endofunction $[1, 1]$. When the left component is listed first, we obtain the endofunction $[1, 2, 2]$, while if the right component is listed first, we obtain the smaller endofunction $[1, 1, 3]$.

2.5. **The algorithm and its running time.** Here is a summary of the algorithm for obtaining the smallest representative of the S_n-conjugacy class of a given endofunction $t \in T_n$.

Step 1: Construct the digraph $G(t)$, identify its connected components and its cycles.

Step 2: For each connected component, calculate structural vertex and edge labels for every tree rooted in the cycle, see §2.3.

Step 3: For each connected component, determine the location of 1 on the cycle, see §2.1 and §2.3.

Step 4: Order the connected components according to the spelling bee dictionary order, see §2.2, and concatenate the minimal representatives of connected components, shifting the entries of each component suitably.

As for the complexity of the algorithm, we will use the following result:

Lemma 2.11. *Let a_1, …, a_m be sequences over an r-letter alphabet and of total combined length n. Then a_1, …, a_m can be lexicographically sorted in time $O(n)$ using m bins of total capacity n, assuming that each of the r symbols can be processed in unit time and stored in unit space.*

Proof. Scan the first entry of every sequence and place the sequences in one of the r bins accordingly. Ignoring empty bins, we need at most m bins since there are only m sequences. Since n is the total length of the sequences, the total bin capacity of n suffices. In any given bin we can now ignore the first entry of sequences therein and sort the sequences recursively by focusing on the second entry, etc. Since every entry of every sequence is scanned at most once, we finish in time $O(n)$. □

Step 1 of the algorithm can be done in time $O(n)$.

In Step 2, at each of the $\leq n$ vertices of the trees, we need to sort at most n structural labels of total combined length at most n. By Lemma 2.11, this can be done in $nO(n) = O(n^2)$.

In Step 3, consider a connected component U_i of size u_i with a cycle of size c_i. To place 1 on the cycle, we consider all c_i possible locations, construct the corresponding endofunctions, and keep the smallest one. Each of the c_i endofunctions can be constructed in $O(u_i)$, so all can be constructed in $c_iO(u_i) \leq u_iO(u_i) = O(u_i^2)$. The smallest endofunction can then be found in $c_iO(u_i) \leq O(u_i^2)$. Since $\sum u_i = n$, we have $\sum u_i^2 \leq n^2$, and the entire step can be completed in $O(n^2)$.

In Step 4, we need to sort sequences of total length n in the spelling bee dictionary order (which for the purposes of computational complexity is the same as the lexicographic order), which can be done in $O(n)$ by Lemma 2.11.

Overall, the algorithm runs in $O(n^2)$.

The algorithm has been implemented in `GAP` [9] using package `Digraphs` [7]. It can be downloaded from the webpage of the third author `http://www.math.du.edu/~petr`.

We have tested the algorithm against the general state-of-the-art algorithm for minimal images under actions of permutation groups, implemented in the `GAP` package `images` [11]. Given a transformation f on $\{1, \dots, n\}$, the minimal representative of the S_n-orbit of f can be calculated with `images` by the command `MinimalImage( SymmetricGroup( n ), f )`, where the default action `OnPoints` is the conjugation of transformations by permutations.

Table 1. A comparison of running times (in milliseconds) of our algorithm and the algorithm of `images`.

n	10	20	30	40	50	100	200	500	1000	10000
ours	0.2	0.3	0.4	0.6	0.9	2.2	6.0	30.3	89.3	8471.6
`images`	1.5	8.8	33.3	95.9	284.9	8112.8				

Table 1 summarizes the average running time of the two algorithms on a random transformation of $\{1, \dots, n\}$. We have used one hundred transformations for every $n \leq 50$ and ten transformations for every $n > 50$. For a given n, the same list of random transformations has been used by both algorithms. The algorithm of `images` did not finish in ten minutes on a transformation of length $n \geq 200$ and we therefore do not report the running times in those cases. Whenever both algorithms finished, we compared the results—they always agreed.

3. Concluding remarks

A special case of the algorithm occurs when $t \in T_n$ is a permutation. The standard way of finding the minimal representative of a conjugacy class in S_n containing a permutation t is to calculate the cycles of t, order the cycles by length in increasing order, and label elements in cycles consecutively. Note that this can be performed in $O(n)$ by a bin sort, where all k-cycles are stored in the bin labeled k.

The algorithm presented here can also be used to calculate the automorphism group of a functional digraph G on n vertices. Let us label the vertices of G arbitrarily by $\{1, \dots, n\}$. We can now calculate the minimal representative of the corresponding endofunction. In the process we obtain structural labels, we locate 1 in every cycle, and we order the connected components. Anytime a tie occurs among the structural edge labels leading to a vertex v (as indicated by asterisks in Example 2.8), the corresponding isomorphic subtrees can be permuted accordingly, giving rise to an automorphism of G. Rotational symmetries of a connected component can be found upon detecting the same concatenation of structural vertex labels while going around the cycle from two different starting points. Finally, while ordering the connected components, every tie gives rise to yet another automorphism of G that flips the connected components. The automorphisms so obtained generate the automorphism group of G.

References

[1] A.V. Aho, J.E. Hopcroft and J.D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, MA, 1974.

[2] L. Babai, *Graph isomorphism in quasipolynomial time [extended abstract]*, STOC'16–Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, 684–697, ACM, New York, 2016.

[3] G. Busacker and T.L. Saaty, Finite Graphs and Networks, McGraw Hill, New York, 1965.

[4] C.J. Colbourn and K.S. Booth, *Linear time automorphism algorithms for trees, interval graphs, and planar graphs*, SIAM J. Comput. **10** (1981), no. **1**, 203–225.

[5] J. Crawford, M. Ginsberg, E. Luks and A. Roy, *Symmetry-breaking predicates for search problems*, Proceedings 5th International Conference on Principles of Knowledge Representation and Reasoning, 1996, 148–159.

[6] R.L. Davis, *The number of structures of finite relations*, Proc. Amer. Math. Soc. **4** (1953), 486–495.

[7] J. De Beule, J. Jonušas, J.D. Mitchell, M. Torpe, M. Tsalakou and W.A. Wilson, Digraphs - `GAP` package, Version 1.7.1, Feb 2024, `https://digraphs.github.io/Digraphs`

[8] N.G. de Bruijn, *Enumeration of mapping patterns*, J. Combinatorial Theory Ser. A **12** (1972), 14–20.

[9] The GAP Group, *GAP – Groups, Algorithms, and Programming, Version 4.13.1*; 2024, `https://www.gap-system.org`.

[10] C. Jefferson, E. Jonauskyte, M. Pfeiffer and R. Waldecker, *Minimal and canonical images*, J. Algebra **521** (2019), 481–506.

[11] C. Jefferson, M. Pfeiffer, R. Waldecker and E. Jonauskyte, `images`, Minimal and Canonical Images, Version 1.3.2 (2024), `GAP` package, `https://gap-packages.github.io/images/`

[12] J. Ježek, *Enumerating left distributive groupoids*, Czechoslovak Math. J. **47**(**122**) (1997), no. **4**, 717–727.

[13] T. Junttila and P. Kaski, *Conflict propagation and component recursion for canonical labeling*, in TAPAS 2011, 151–162.

[14] S. Linton, *Finding the smallest image of a set*, in ISSAC '04: Proceedings of the 2004 international symposium on symbolic and algebraic computation, July 2004, 229–234.

[15] B.D. McKay and A. Piperno, *Practical graph isomorphism, II*, J. Symbolic Comput. **60** (2014), 94–112.

[16] B.D. McKay and A. Piperno, `nautyTraces`, software distribution webpage `http://cs.anu.edu.au/~bdm/nauty` and `http://pallini.di.uniroma1.it`.

[17] N.J.A. Sloane and The OEIS Foundation Inc., *The on-line encyclopedia of integer sequences*, `http://oeis.org`.

[18] L.H. Soicher, GRAPE, graph algorithms using permutation groups, version 4.9.0, `https://gap-packages.github.io/grape`, December 2022, refereed GAP package.

(Mitchell) School of Mathematics and Statistics, University of St Andrews, St Andrews, Scotland
Email address, Mitchell: `jdm3@st-andrews.ac.uk`

(Mukherjee,Vojtěchovský) Department of Mathematics, University of Denver, 2390 S. York St., Denver, CO 80208, USA
Email address, Mukherjee: `sujoymukherjee.math@gmail.com, sujoy.mukherjee@du.edu`

Email address, Vojtěchovský: `petr@math.du.edu`