

# MINIMAL REPRESENTATIVES OF ENDOFUNCTIONS

SUJOY MUKHERJEE AND PETR VOJTĚCHOVSKÝ

ABSTRACT. We present a simple and efficient algorithm for finding a 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  $t : X \rightarrow X$ . 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  $t : X \rightarrow X$ , we obtain a functional digraph  $G(t) = (X, E)$  by setting  $E = \{(x, f(x)) : x \in X\}$ . 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  $G \rightarrow H$  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  $x = z_0, z_1, \dots, z_m = y$  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  $C(U)$  and its length by  $c(U)$ . Upon removing the edges of  $C(U)$ ,  $U$  becomes a union of disjoint directed trees rooted in (or, better, with the sink located in)  $C(U)$ . 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

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

be the set of all in-neighbors of  $x$ . For a functional digraph  $G(t)$ , we of course have  $N(x) = t^{-1}(x)$ , 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  $A$ . A *minimal representative* of  $a \in A$  is an element  $m_a \in a^H$  such that  $m_a \leq b$  for all  $b \in a^H$ . Note that a minimal representative is a canonical representative. Generally speaking, finding minimal representatives appears to be more difficult than finding canonical representatives.

Linton [11] introduced a general algorithm for finding a minimal representative of the action of a permutation group  $H$  on  $k$ -element subsets of a set  $A$ ; the algorithm has been implemented in [15]. Jefferson et al. [8] presented several algorithms for finding a canonical 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  $|A| = n^n$ .

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 a 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 [10, 12, 13] but none of these tools calculates minimal representatives. It is proved in [5] that the problem of finding a 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 a 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 a minimal representative in the isomorphism class of  $(X, \cdot)$ .

The diagonals of groupoids were exploited by Ježek [9] 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 [7, p. 18], where  $a_n$  has been explicitly calculated for  $n \leq 15$ , and the OEIS sequence [14, A001372] for  $a_n$  with  $n \leq 1000$ .

## 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 can be found in §2.5.

Let  $t \in T_n$  be a minimal representative (of its  $S_n$ -conjugacy class). 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  $C(U)$  is the cycle  $(1, 2, \dots, c(U))$ , possibly with  $c(U) = 1$ . We do not yet determine which vertex of  $C(U)$  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.

Suppose that  $G(t)$  is connected. Upon removing the edges of the unique cycle  $C(X)$  of  $G(t)$ , we obtain disjoint directed trees rooted in  $C(X)$ . We can decompose  $X$  as a disjoint union  $L_0 \cup \dots \cup L_k$ , where  $L_i$  consists of the vertices of  $x \in X$  at distance  $i$  from  $C(X)$ . Let  $\ell_i = |L_i|$ . We already know that  $L_0 = C(X) = \{1, \dots, \ell_0\}$  and it easily follows that  $L_{i+1} = \{\sum_{0 \leq j < i} \ell_j + 1, \dots, \sum_{0 \leq j < i} \ell_j + |L_i|\}$ , that is, the level sets  $L_i$  consist of consecutively labeled vertices. 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 structural labels will also allow us to determine the location of 1 in  $C(X)$ . This is done in §2.3.

### 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  $1 \in C(U)$  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  $0 \leq j < k$  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.

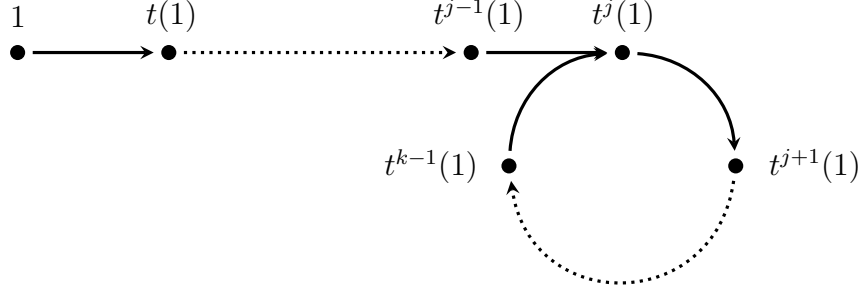


FIGURE 1. The path starting at 1.

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  $1 \in C(U)$  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.  $\square$

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  $C(U)$  should be used.

Let us introduce notation for some subsets of a connected component  $U$  of a functional digraph. For  $k \geq 0$ , 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

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

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)  $U = \{1, \dots, |U|\}$ .

*Proof.* Let  $c = c(U)$  and  $d_x = |M(x)|$ . By Lemma 2.1, we have  $L_0 = C(U) = \{1, \dots, c\}$ . 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  $M(i) = \{c+1, \dots, c+d_i\}$ . Next let  $i < j \in L_0$  be least such that  $M(j) \neq \emptyset$ . If such  $j$  exists, we must have  $M(j) = \{c+d_i+1, \dots, c+d_i+d_j\}$ , 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.  $\square$

For any two subsets  $U = \{u_1, \dots, u_k\}$ ,  $V = \{v_1, \dots, v_k\}$  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  $U_1, \dots, U_m$  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  $[t(1), \dots, t(n)]$ , is the concatenation of the tuples  $t_{U_{\pi(1)}}, t_{U_{\pi(2)}}, \dots, t_{U_{\pi(m)}}$ , 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,  $U_1 = \{1, \dots, |U_1|\}$  by Lemma 2.2 and hence  $X \setminus U_1 = \{|U_1|+1, \dots, |X|\}$ . Modulo a formal shift of vertex labels, the induction assumption implies that  $t_{X \setminus U_1}$  is a concatenation of  $t_{U_{\pi(2)}}, \dots, t_{U_{\pi(m)}}$  for some  $\pi \in S_{\{2, \dots, m\}}$ , as well as the rest of the claim.  $\square$

**2.2. Ordering the connected components.** Let  $t \in T_n$  be a minimal representative and let  $U_1, \dots, U_m$  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  $t_{U_{\pi(1)}}, t_{U_{\pi(2)}}, \dots, t_{U_{\pi(m)}}$ . Moreover,  $U_{\pi(1)}, \dots, U_{\pi(m)}$  are consecutive intervals. We will now determine  $\pi$ .

For each  $i$ ,  $t_{U_i \rightsquigarrow \{1, \dots, |U_i|\}}$  is a minimal representative on  $\{1, \dots, |U_i|\}$ . 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),  $s_V(|U|+1) < |U|+1$ .  $\square$

**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  $1 \leq i \leq m$ .*

*Proof.* Suppose that  $s_i \prec s_1$  for some  $i > 1$ . Let  $k = \min\{|U_1|, |U_i|\}$ . If there is  $j \leq k$  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  $j \leq k$ ,  $s_1$  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  $t_{U_2}(k+1) \geq k+1$ . Therefore, if we use  $U_i$  first and relabel suitably, we obtain a smaller representative than  $t$ , a contradiction.  $\square$

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*  $\sigma(v) = [\sigma(v)_j : j \geq 0]$  of its vertices and  $\sigma(e)$  of its edges recursively as follows. If  $N(v) = \emptyset$  (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  $N(v) = \{v_1, \dots, v_d\} \neq \emptyset$ , suppose that  $v_1, \dots, v_d$  are ordered lexicographically according to their structural labels so that  $\sigma(v_1) \geq \sigma(v_2) \geq \dots \geq \sigma(v_d)$ . Then  $\sigma(v) = [d, \sigma_0, \sigma_1, \dots]$ , where  $\sigma_j$  is the concatenation of  $\sigma(v_1)_j, \dots, \sigma(v_d)_j$ . In addition, label the edges  $(v_i, v)$  by  $\sigma(v_i, v) = i$ . 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  $N(v) = \{v_1, \dots, v_d\}$ , ordered first according to *their* indegrees, and then recursively. The edge labels  $\sigma(v_i, v)$  make it clear how the vertices of  $N(v)$  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  $M(x)$  are ordered according to the structural edge labels on the edges  $(y, x)$ ,  $y \in M(x)$ .*

*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 a 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.  $\square$

Finally, we can determine the location of 1 in  $C(X)$  by considering all  $c(X)$  options. For every tentative placement of 1 on the cycle, we can produce a sequence by concatenating

the structural vertex labels while going around the cycle, starting at 1. We then select a lexicographically largest sequence among the  $c(X)$  sequences. Note that in many situations we can quickly limit the possible locations of 1. For instance, 1 must certainly have the largest indegree of any vertex on the cycle, by Lemma 2.7.

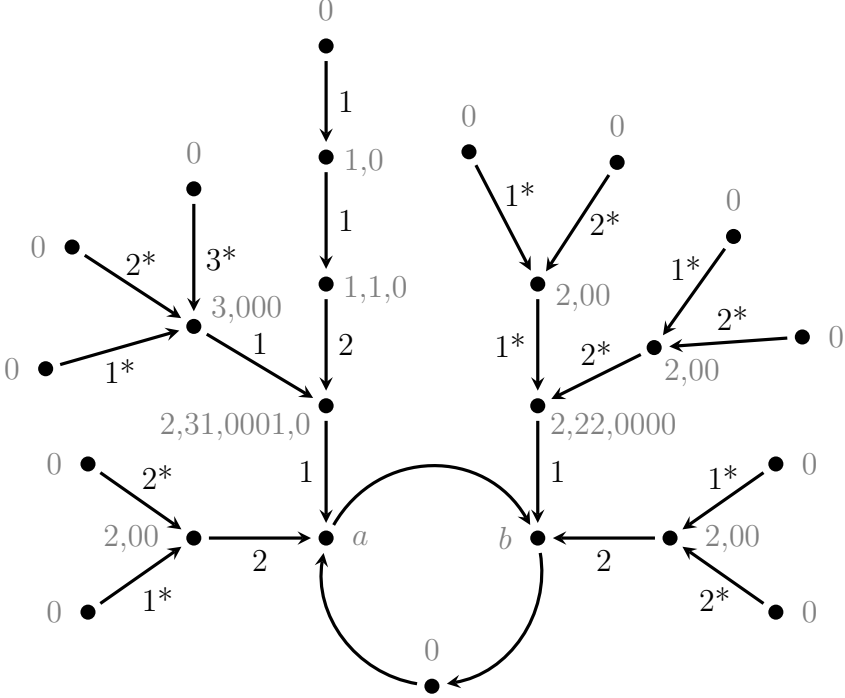


FIGURE 2. Structural labels in a connected component.

2.4. Examples.

**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. In Figure 2 we have removed brackets around structural vertex labels to improve legibility, and added asterisks on structural edge labels to indicate a tie. The missing structural vertex labels are  $a = [2, 22, 3100, 0001, 0]$  and  $b = [2, 22, 2200, 0000]$ . Since  $a$  is the unique largest structural label on the cycle, 1 must be positioned at  $a$ . One of the minimal representatives of the component (giving rise to a unique minimal representative of the corresponding endofunction) is therefore depicted in Figure 3.

**Example 2.9.** Figure 4 illustrates that the cycle labels of a connected component need not be uniquely determined on account of a rotational symmetry.

If we position 1 at the top (or the bottom) vertex of the cycle, we obtain the sequence 010010 going around the cycle, while if we position 1 at the right (or the left) vertex of the cycle, we obtain the larger sequence 100100. Hence 1 must be placed at the right or the left vertex of the cycle to obtain a minimal representative.

**Example 2.10.** Finally, Figure 5 shows why the spelling bee dictionary ordering arises while comparing connected components.

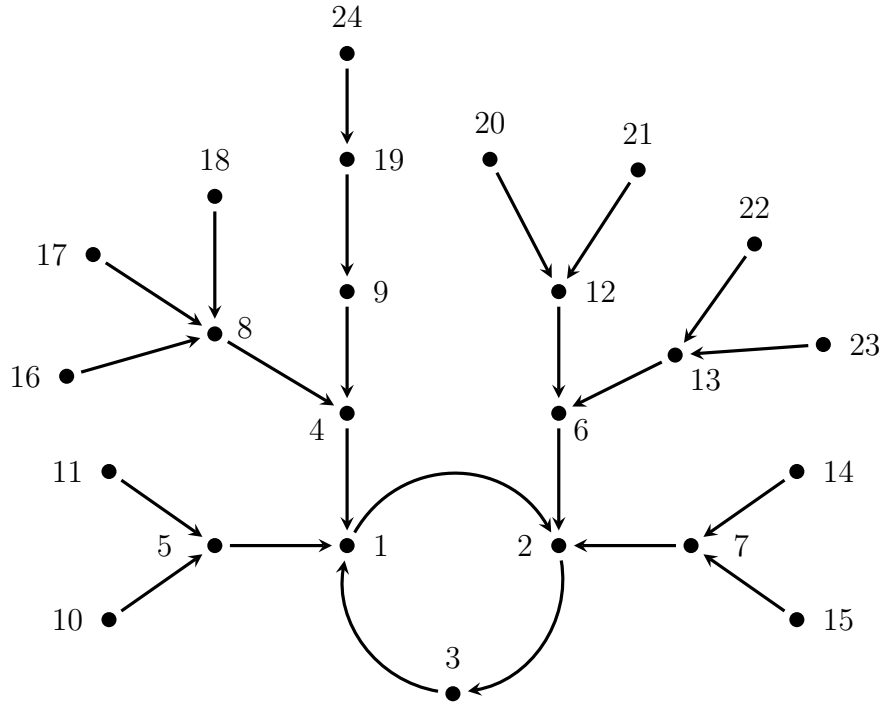


FIGURE 3. The corresponding minimal representative.

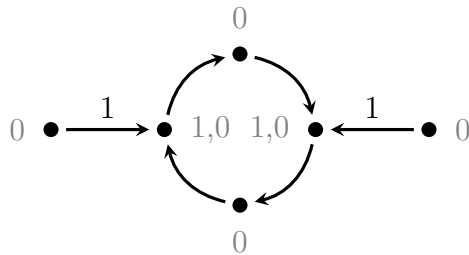


FIGURE 4. Rotational symmetry in a connected component.

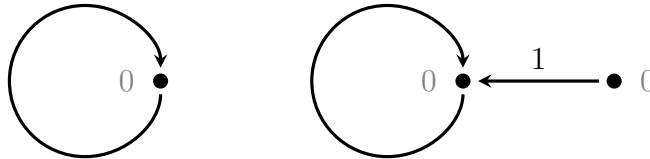


FIGURE 5. Ordering connected components.

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 complexity.** 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  $a_1, \dots, a_m$  be sequences over an  $r$ -letter alphabet and of total combined length  $n$ . Then  $a_1, \dots, a_m$  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  $O(n)$ .  $\square$

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  $c_i$ . To locate 1 on the cycle of  $U_i$ , we need to find a largest element among  $c(U_i) \leq c_i$  concatenated structural labels, each of length  $\leq c_i$ . By Lemma 2.11, this can be done in  $O(c_i^2)$ . Since  $\sum c_i = n$ , we have  $\sum c_i^2 \leq n^2$  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 complexity is the same as the lexicographic order), which can be done in  $O(n)$  by Lemma 2.11.

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

### 3. CONCLUDING REMARKS

A special case of the algorithm occurs when  $t \in T_n$  is a permutation. The standard way of finding a 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  $k$ .

Our algorithm can 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 subtrees can be permuted accordingly, giving rise to an automorphism of  $G$ . While locating the position of 1 on the cycle, every tie among the  $n$  concatenated sequences of structural vertex labels gives rise to an automorphism of  $G$  corresponding to a rotational symmetry. 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  $G$ .

The algorithm has been implemented in GAP and can be obtained from the webpage of the second author, presently at <http://www.math.du.edu/~petr>.

#### ACKNOWLEDGMENT

We thank the referee for several comments that improved the presentation of the paper as well as for pointing out additional literature on the topic of canonical and minimal representatives.

#### 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] N.G. de Bruijn, *Enumeration of mapping patterns*, J. Combinatorial Theory Ser. A **12** (1972), 14–20.
- [8] C. Jefferson, E. Jonauskyte, M. Pfeiffer and R. Waldecker, *Minimal and canonical images*, J. Algebra **521** (2019), 481–506.
- [9] J. Ježek, *Enumerating left distributive groupoids*, Czechoslovak Math. J. **47(122)** (1997), no. **4**, 717–727.
- [10] T. Junttila and P. Kaski, *Conflict propagation and component recursion for canonical labeling*, in TAPAS 2011, 151–162.
- [11] 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.
- [12] B.D. McKay and A. Piperno, *Practical graph isomorphism, II*, J. Symbolic Comput. **60** (2014), 94–112.
- [13] B.D. McKay and A. Piperno, *nautyTraces*, software distribution webpage <http://cs.anu.edu.au/~bdm/nauty> and <http://pallini.di.uniroma1.it>.
- [14] N.J.A. Sloane and The OEIS Foundation Inc., *The on-line encyclopedia of integer sequences*, <http://oeis.org>.
- [15] L.H. Soicher, *GRAPE*, graph algorithms using permutation groups, version 4.9.0, <https://gap-packages.github.io/grape>, December 2022, refereed GAP package.

(Mukherjee, Vojtěchovský) DEPARTMENT OF MATHEMATICS, UNIVERSITY OF DENVER, 2390 S. YORK ST., DENVER, CO 80208, USA

*Email address*, Mukherjee: [sujoymukherjee.math@gmail.com](mailto:sujoymukherjee.math@gmail.com), [sujoy.mukherjee@du.edu](mailto:sujoy.mukherjee@du.edu)

*Email address*, Vojtěchovský: [petr@math.du.edu](mailto:petr@math.du.edu)