

Article

Enumerating Tree-Like Graphs and Polymer Topologies with a Given Cycle Rank

Naveed Ahmed Azam *, Aleksandar Shurbevski and Hiroshi Nagamochi

Department of Applied Mathematics and Physics, Kyoto University, Kyoto 606-850, Japan; shurbevski@amp.i.kyoto-u.ac.jp (A.S.); nag@amp.i.kyoto-u.ac.jp (H.N.)

* Correspondence: azam@amp.i.kyoto-u.ac.jp

Received: 26 October 2020; Accepted: 9 November 2020; Published: 13 November 2020



Abstract: Cycle rank is an important notion that is widely used to classify, understand, and discover new chemical compounds. We propose a method to enumerate all non-isomorphic tree-like graphs of a given cycle rank with self-loops and no multiple edges. To achieve this, we develop an algorithm to enumerate all non-isomorphic rooted graphs with the required constraints. The idea of our method is to define a canonical representation of rooted graphs and enumerate all non-isomorphic graphs by generating the canonical representation of rooted graphs. An important feature of our method is that for an integer $n \geq 1$, it generates all required graphs with n vertices in $\mathcal{O}(n)$ time per graph and $\mathcal{O}(n)$ space in total, without generating invalid intermediate structures. We performed some experiments to enumerate graphs with a given cycle rank from which it is evident that our method is efficient. As an application of our method, we can generate tree-like polymer topologies of a given cycle rank with self-loops and no multiple edges.

Keywords: chemical graph; polymer topology; cycle rank; enumeration; canonical representation

1. Introduction

The problem of enumerating discrete structures has several applications in applied fields such as graph theory, chemoinformatics, bioinformatics, and material informatics [1–11]. In particular, the enumeration of chemical compounds is widely used in the discovery of novel drugs [12–17] and structural elucidation [18]. It is necessary for an enumeration method to generate all possible required structures without duplication in low computational complexity, due to which designing an enumeration method is not an easy task.

The problem of the enumeration of chemical compounds with given constraints is often modeled as the problem of the enumeration of graphs. Several chemical compound generation methods have been proposed [3–9], where some methods [3,4] focus on general chemical compounds, while the other methods [5–8] deal with restricted chemical graphs. These methods are mainly based on the branching algorithm paradigm; the required chemical compounds appear at the leaves of a computation tree. However, these algorithms generate many invalid intermediate structures that appear at the non-leaf nodes of the computation tree [9]. Due to this fact, these methods are inefficient to generate chemical compounds with more than 20 non-hydrogen atoms. Thus, it is natural to explore and develop such methods that can enumerate chemical compounds without generating invalid intermediate structures. Jin et al. [9] proposed one such chemical compound generation method based on the junction tree and the variational autoencoder.

For a chemical compound C , the *polymer topology* is a connected multi-graph where all vertices have degree at least three obtained by iteratively removing all vertices of degree at most two from C [19]. For example, the polymer topologies of the chemical compounds remdesivir $C_{27}H_{35}N_6O_8P$ (Figure 1a) and dexamethasone $C_{22}H_{29}F_0_5$ (Figure 1b) are illustrated in Figure 1c,d, respectively. Observe that

two different chemical compounds can have the same polymer topology, and the categorization of polymer topologies can play an important role in understanding and studying the synthetic pathways of macro-chemical compounds [20]. Polymer topologies P are often classified with respect to their *cycle rank*, which is the number of edges that are necessary to remove to get a spanning tree of P . Haruna et al. [19] developed an enumeration method to generate all polymer topologies with a given rank by using frontier-based search and zero-suppressed decision diagrams. As a result, they enumerated all polymer topologies with cycle rank at most 6. For a multi-graph G , we define the *skeleton* to be the simple graph obtained by removing all self-loops and multiple edges from G . Notice that the class of graphs with a tree skeleton, $\Delta \geq 0$ self-loops, and no multiple edges contains all tree-like polymer topologies with cycle rank Δ , and therefore, it is an interesting problem to enumerate all such graphs. Figure 2 illustrates examples of chemical compounds that have tree-like polymer topologies with self-loops and no multiple edges. Recently, Azam et al. [21] proposed a method to count all trees with given numbers of vertices and self-loops by using dynamic programming. As a result, they gave the upper bound and the lower bound on the number of tree-like mutually non-isomorphic polymer topologies with a given rank.

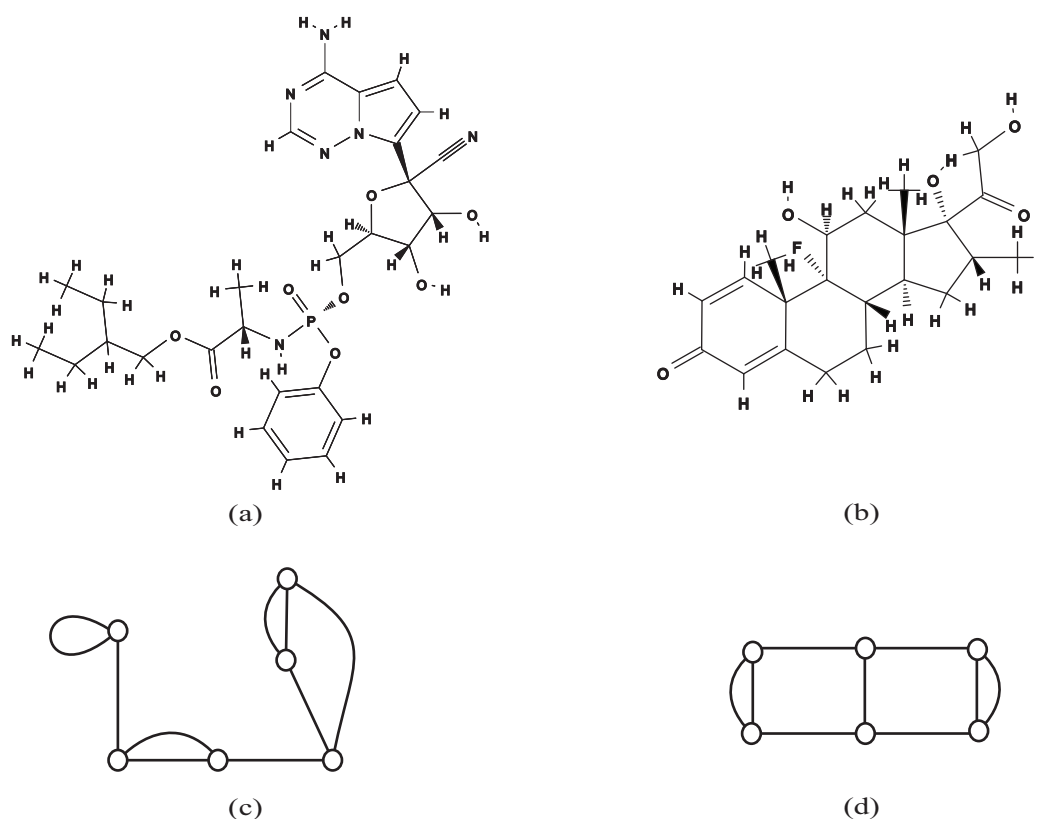


Figure 1. The chemical compounds remdesivir $C_{27}H_{35}N_6O_8P$ and dexamethasone $C_{22}H_{29}FO_5$ and their polymer topologies: (a) the chemical structure of remdesivir $C_{27}H_{35}N_6O_8P$ taken from PubChem; (b) the chemical structure of dexamethasone $C_{22}H_{29}FO_5$ taken from PubChem; (c) the polymer topology of remdesivir with cycle rank 4; and (d) the polymer topology of dexamethasone with cycle rank 4.

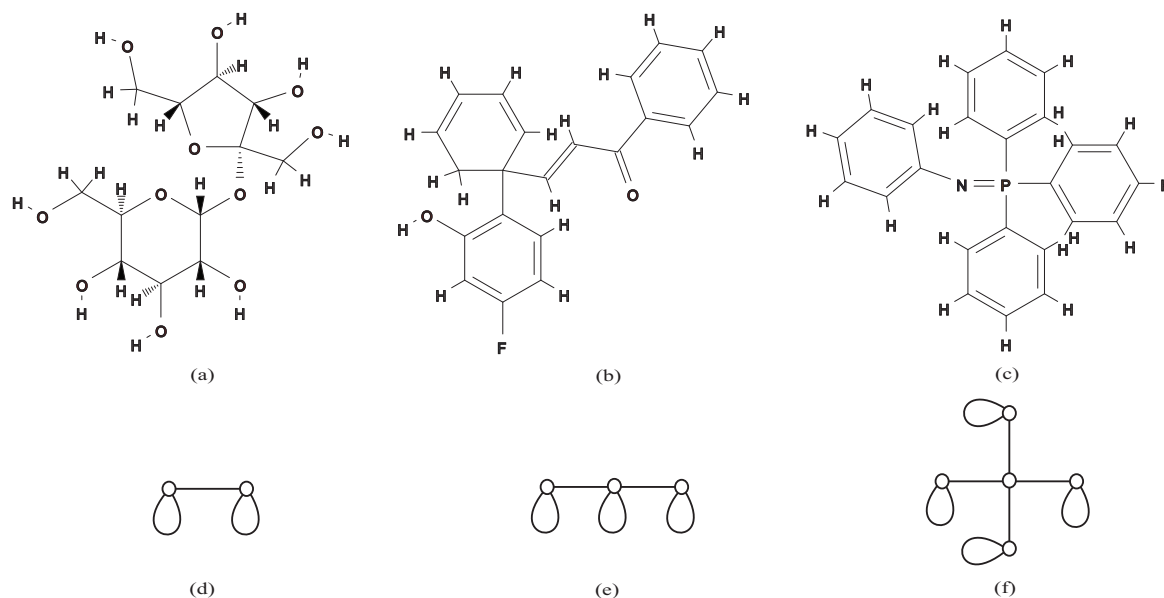


Figure 2. Three chemical compounds with their tree-like polymer topologies containing self-loops: (a–c) the chemical structures of $C_{12}H_{22}O_{11}$, $C_{21}H_{17}FO_2$, and $C_{24}H_{20}NP$ with CIDs 5988, 137,321,354, 75,352, respectively, obtained from the PubChem database; (d–f) the polymer topologies of the chemical structures in (a–c) with cycle ranks 2–4, respectively.

This article aims to develop an efficient method to enumerate all mutually non-isomorphic graphs with a tree skeleton, n vertices, Δ self-loops, and no multiple edges without generating invalid intermediate structures. The idea of our method is to define a canonical representation of rooted graphs with the said structures and then enumerate these graphs by generating their canonical representations. As a consequence of our method, we can get all polymer topologies with a tree skeleton, a given cycle rank, self-loops, and no multiple edges. We organize the paper as follows: In Section 2, we discuss some preliminaries. In Section 3, we first prove the mathematical properties based on which we develop our enumeration method. We discuss experimental results and an application of our enumeration method to generate all polymer topologies with a tree skeleton and a given cycle rank in Section 4. We conclude and discuss some future directions in Section 5.

2. Preliminaries

For a graph G , let $V(G)$ denote the vertex set and $E(G)$ denote the edge set. Let $n(G)$ denote $|V(G)|$ and $\text{self}(G)$ denote the number of self-loops in G . We define size $s(G)$ of graph G to be the sequence $(n(G), \text{self}(G))$. Let $s(v)$ denote the number of self-loops on $v \in V(G)$. For $v \in V(G)$, we denote by $N_G(v)$ the set of neighbors of v other than v . The *degree* $\text{deg}_G(v)$ of $v \in V(G)$ is defined to be the size of $N_G(v)$.

For a multi-graph G , we define the skeleton $\gamma(G)$ of G to be the simple graph obtained by removing all self-loops and multiple edges from the graph G . For a rooted graph G with root r_G , we define the *rooted skeleton* $\gamma(G)$ of G to be the rooted simple graph obtained by removing all self-loops from G with root r_G .

Let $n \geq 1$ and $\Delta \geq 0$ be two integers. We denote by $\mathcal{H}(n, \Delta)$ a maximal set of mutually non-isomorphic rooted graphs with a tree skeleton, n vertices, and Δ self-loops. Let H be a rooted graph in $\mathcal{H}(n, \Delta)$. For a vertex $v \in V(H)$, let H_v denote the subgraph of H rooted at v induced by v and its descendants in the rooted skeleton $\gamma(H)$. For a vertex $v \in N_H(r_H)$ of root r_H of H , we call the subgraph H_v a *root-subgraph* of H .

Let $n \geq 1$ and $\Delta \geq 0$ be two integers and H be a rooted graph in $\mathcal{H}(n, \Delta)$. An *ordered graph* (H, π) of H is defined to be the rooted graph H with a left-to-right ordering π on the children of each vertex of the rooted skeleton $\gamma(H)$. Let $K = (H, \pi)$ be an ordered graph of H . For a vertex $v \in V(K)$, we define

the ordered subgraph K_v of K to be a subgraph of K rooted at v induced by v and its descendants in the rooted skeleton $\gamma(K)$ with preserving the ordering π on the children of each vertex in $\gamma(K_v)$. For a vertex $v \in N_K(r_K)$, we call the ordered subgraph K_v an ordered root-subgraph of K .

For an ordered tree, we discuss two vertex orderings: depth first search (DFS) ordering [22] and sibling-depth first search (SDFS) ordering. In DFS ordering, we index the vertices of a given ordered tree starting from the root and visiting them from left to right. Masui et al. [23] introduced the SDFS ordering for simple ordered trees. For an ordered tree $T = (L, \pi)$ with n vertices and a left-to-right ordering π , the SDFS ordering is defined to be a vertex ordering obtained by indexing the vertices from the set $\{1, 2, \dots, n\}$ such that:

- (i) the root has index one;
- (ii) all siblings are indexed consecutively according to the left-to-right ordering π ; and
- (iii) all descendants of a vertex v are indexed consecutively with indices larger than that of v and smaller than the indices of the descendants of any vertex u , which is not a descendant of v with index larger than v .

Examples of an ordered tree and its vertex indexing in DFS and SDFS ordering are illustrated in Figure 3.

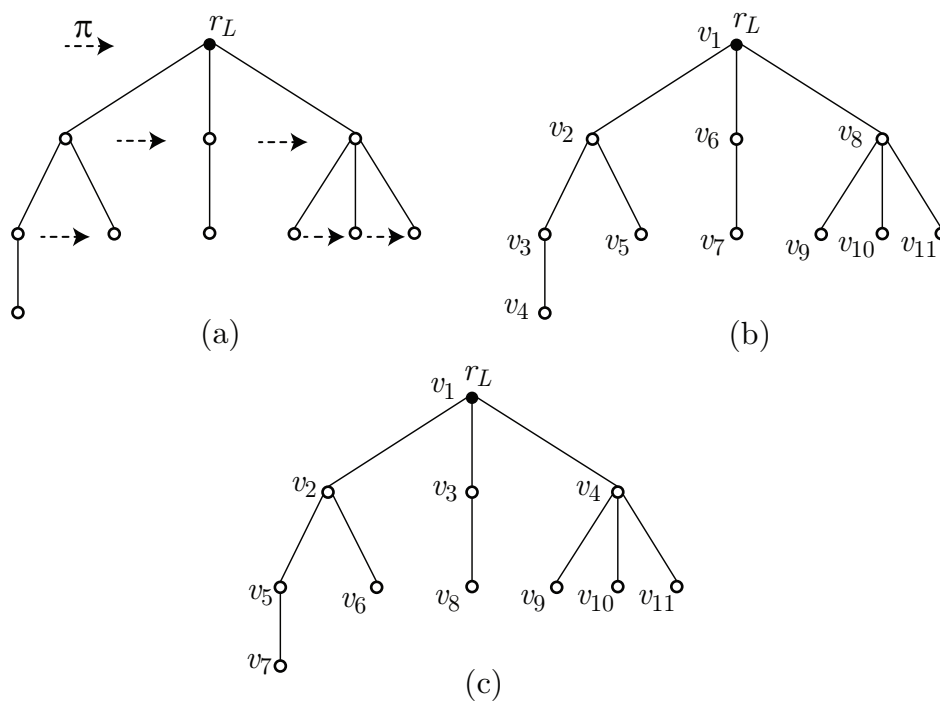


Figure 3. Examples of an ordered tree and vertex indexing: (a) an ordered tree $T = (L, \pi)$ with left-to-right ordering π indicated by the dashed arrow; (b) ordered tree $T = (L, \pi)$ from (a) with vertices indexed in depth first search (DFS) order; and (c) ordered tree $T = (L, \pi)$ from (a) with vertices indexed in sibling-depth first search (SDFS) order.

Let $A = (a_1, a_2, \dots, a_n)$ and $B = (b_1, b_2, \dots, b_m)$ be two sequences over integers. We say that the sequence A is lexicographically smaller $A \prec B$ than the sequence B if there exists an integer ℓ , $1 \leq \ell \leq \min\{n, m\}$, such that for each integer i , $1 \leq i \leq \ell$, it holds that $a_i = b_i$ and:

- (i) either $\ell = n$ with $n < m$ or
- (ii) $\ell < \min\{n, m\}$ with $a_{\ell+1} < b_{\ell+1}$.

In such a case, we say that the sequence B is lexicographically greater $B \succ A$ than the sequence A . We define the concatenation $A \oplus B$ of the sequences A and B to be the sequence $(a_1, \dots, a_n, b_1, \dots, b_m)$.

3. Enumeration of Graphs with a Tree Skeleton and a Given Number of Vertices and Self-Loops

For two integers $n \geq 1$ and $\Delta \geq 0$, the aim of this section is to present a method to generate all rooted graphs in $\mathcal{H}(n, \Delta)$. The idea of our enumeration method is to generate a rooted graph $H \in \mathcal{H}(n, \Delta)$ by generating a canonical ordered graph of H . To achieve this, we define a canonical graph of a rooted graph $H \in \mathcal{H}(n, \Delta)$ and represent the canonical ordered graph with a sequence by using its ordered subgraphs. Finally, generate the canonical ordered graph of a rooted graph by using the sequence representation of the canonical ordered graph.

We next present a canonical representation of rooted graphs in $\mathcal{H}(n, \Delta)$ based on a generalization of the canonical representation of simple rooted trees with n vertices introduced by Masui et al. [23]. Recall that $\mathcal{H}(n, 0)$ denote a maximal set of all mutually rooted non-isomorphic simple rooted trees with n vertices. Further, note that for $\Delta \geq 1$, it is necessary for a canonical representation of a rooted graph $H \in \mathcal{H}(n, \Delta)$ to contain the information of vertices and self-loops in H .

Let H be a rooted graph in $\mathcal{H}(n, \Delta)$ and r_H denote its root. Further, let $K = (H, \pi)$ be an ordered graph of H with a left-to-right ordering π . For an integer $i \in [1, n]$ and i -th vertex v_i of K following the SDFS ordering on the rooted skeleton $\gamma(K)$, let $K_{(i)}$ denote the ordered subgraph K_{v_i} of K for convenience.

We introduce a canonical representation of K by using the information of the number of vertices and self-loops in the ordered subgraphs of K . For the vertices $\{v_1, v_2, \dots, v_n\}$ of K indexed by SDFS ordering on the rooted skeleton $\gamma(K)$, we define the sequence representation $SR(K)$ of K to be a sequence of the size of each ordered subgraph $K_{(i)}$, integer $i \in [2, n]$, of K :

$$SR(K) \triangleq (s(K_{(2)}), s(K_{(3)}), \dots, s(K_{(n)})).$$

Examples of a rooted graph $H \in \mathcal{H}(11, 3)$, an ordered graph $K = (H, \pi)$ of H with a left-to-right ordering π , and vertices indexed in SDFS ordering and canonical representation $SR(K)$ of K are illustrated in Figure 4a–c.

The next lemma states that the sequence representation of an ordered graph K is a concatenation of:

- (i) a sequence of the size of the root-subgraphs of K in the left-to-right ordering and
- (ii) the sequence representation of all root-subgraphs of K following the left-to-right ordering.

Lemma 1. *Let K be an ordered graph with $n \geq 1$ vertices and $\Delta \geq 0$ self-loops. For integers $d = \deg_K(r_K)$ and $i \in [1, d]$, let K_i denote the i -th root-subgraph of K in the left-to-right ordering. Then, it holds that:*

$$SR(K) = (s(K_1), \dots, s(K_d)) \oplus SR(K_1) \oplus \dots \oplus SR(K_d).$$

Proof. We know that in SDFS ordering, the root vertex is indexed by one, and the siblings of a vertex are indexed consecutively. This implies that the subsequence of the first d entries of $SR(K)$ is equal to $(s(K_1), \dots, s(K_d))$. Furthermore, for an integer $i = 2$ (resp., $i \in [3, d + 1]$), the SDFS ordering assigns the index to the descendants of the i -th vertex consecutively and greater than the indices of the children (resp., descendants) of the $(i - 1)$ -th vertex. From this, it follows that for an integer $i = 2$ (resp., $i \in [3, d + 1]$), the entries of $SR(K_{(i)})$ appear consecutively after the entries of the sequence $s(K_1), \dots, s(K_d)$ (resp., $SR(K_{(i-1)})$) in $SR(K)$. This implies that for an integer $i \in [1, d]$, the subsequence of $SR(K)$ consisting of $n(K_i) - 1$ consecutive entries starting from $d + \sum_{1 \leq h \leq i-1} n(K_h) - (i - 1) + 1$ is actually $SR(K_i)$. Hence, it follows that $SR(K) = (s(K_1), \dots, s(K_d)) \oplus SR(K_1) \oplus \dots \oplus SR(K_d)$. \square

For the ordered graph K in Figure 4c, $SR(K) = ((4, 1), (2, 0), (4, 2), (1, 0), (1, 1), (1, 0), (1, 0), (2, 0), (1, 1), (1, 0))$ and $\deg_K(r_K) = 3$. This implies that K has three root-subgraphs K_1, K_2 and K_3 following the left-to-right ordering. From Figure 4c, we have $(s(K_1), s(K_2), s(K_3)) = ((4, 1), (2, 0), (4, 2))$, $SR(K_1) = ((1, 0), (1, 1), (1, 0))$, $SR(K_2) = ((1, 0))$, and $SR(K_3) = ((2, 0), (1, 1), (1, 0))$. Thus, we see that $SR(K) = (s(K_1), s(K_2), s(K_3)) \oplus SR(K_1) \oplus SR(K_2) \oplus SR(K_3)$.

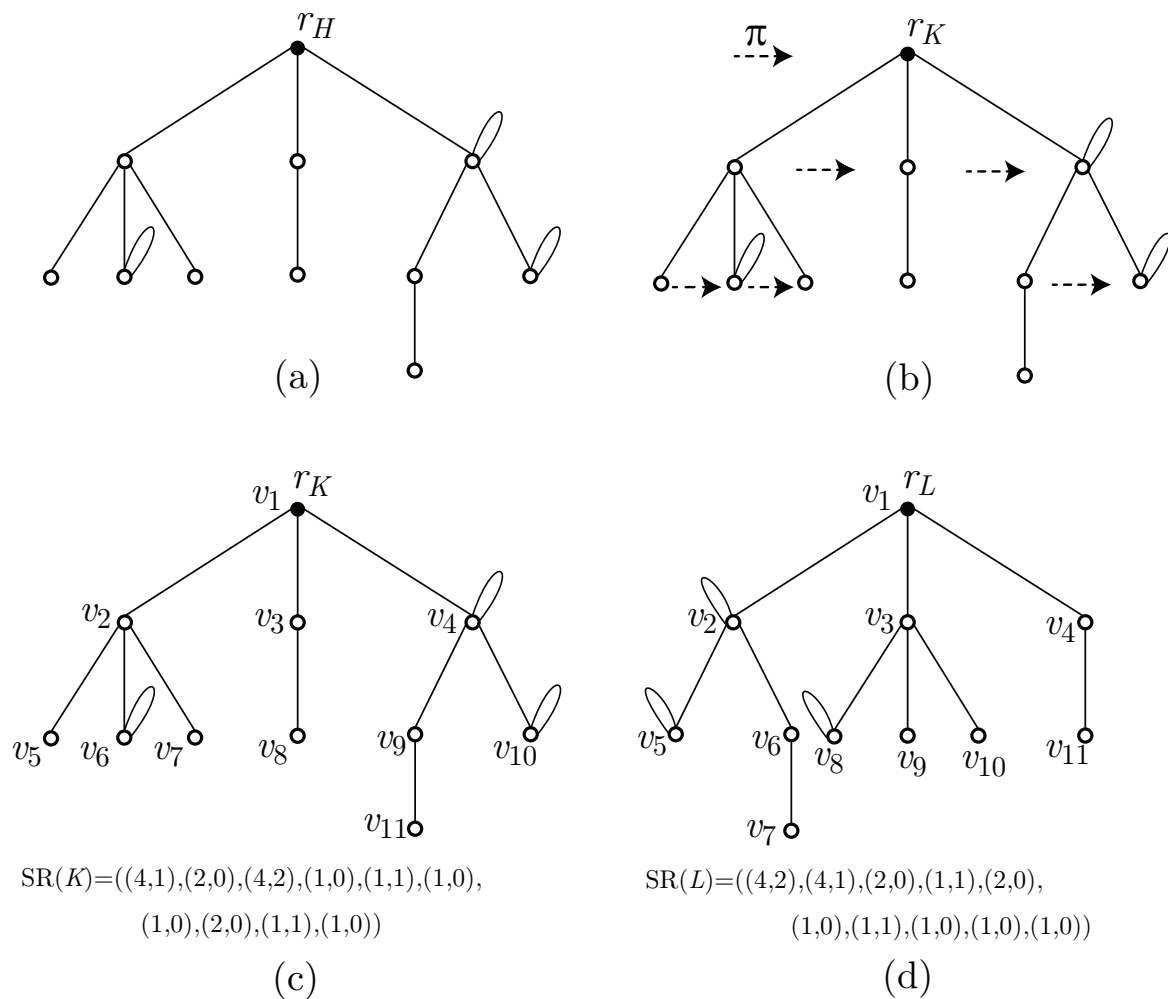


Figure 4. (a) A rooted graph $H \in \mathcal{H}(11, 3)$; (b) an ordered graph K of H with left-to-right ordering π on siblings; (c) the ordered graph K of H with SDFS vertex indexing and $SR(K)$; and (d) the canonical representation of H and the ordered graph with sequence representation equal to the canonical representation of H .

We rephrase the recursion in Lemma 1 for a sequence of pairs in the following paragraph and claim that this recursion is a sufficient condition for a sequence of pairs to be the sequence representation of some ordered graph. We prove this claim in Theorem 1.

Let $n \geq 1$ and $\Delta \geq 0$ be two integers and $M = ((a_1, b_1), (a_2, b_2), \dots, (a_{n-1}, b_{n-1}))$ be a sequence of pairs of integers with $a_i \geq 1$ and $b_i \geq 0$, integer $i \in [1, n - 2]$. We say that the sequence M is (n, Δ) -admissible if either $n = 1$ or:

- (i) there exists an integer $d \in [1, n - 1]$ such that $n - 1 = \sum_{1 \leq i \leq d} a_i$ with $\Delta \geq \sum_{1 \leq i \leq d} b_i$ and
- (ii) for each integer $i \in [1, d]$, the subsequence of M consisting of $a_i - 1$ consecutive entries starting from $d + \sum_{1 \leq h \leq i-1} a_h - (i - 1) + 1$ is (a_i, b_i) -admissible.

Theorem 1. Let $n \geq 1$ and $\Delta \geq 0$ be two integers and $M = ((a_1, b_1), (a_2, b_2), \dots, (a_{n-1}, b_{n-1}))$ be a sequence of pairs of integers with $a_i \geq 1$ and $b_i \geq 0$, integer $i \in [1, n - 2]$.

- (i) Sequence M is the sequence representation $SR(K)$ of some ordered graph K with n vertices and Δ self-loops if and only if M is (n, Δ) -admissible.
- (ii) Whether M is admissible or not can be tested in $\mathcal{O}(n)$ time.
- (iii) When M is (n, Δ) -admissible, the ordered graph K with $SR(K) = M$ can be constructed in $\mathcal{O}(n)$ time.

Proof. For sequence M with an integer $d \in [1, n - 1]$ such that $n - 1 = \sum_{1 \leq i \leq d} a_i$ and $\Delta \geq \sum_{1 \leq i \leq d} b_i$, let M_i denote the subsequence of M consisting of $a_i - 1$ consecutive entries starting from $d + \sum_{1 \leq h \leq i-1} a_h - (i - 1) + 1$. For an ordered graph K and integer $i \in [1, \deg_K(r_K)]$, let K_i denote the i -th root-subgraph of K in the left-to-right ordering.

(i) The if-part: Suppose that $M = \text{SR}(K)$ for some ordered graph K with n vertices and Δ self-loops. If $n = 1$, then M is $(1, \Delta)$ -admissible by the definition of admissibility. Let us assume that $n \geq 2$. Then, for $d = \deg_K(r_K)$, it holds that $n - 1 = \sum_{1 \leq i \leq d} n(K_i)$ and $\Delta \geq \sum_{1 \leq i \leq d} s(K_i)$. Further, by Lemma 1 for each integer $i \in [1, d]$, the subsequence M_i of M is equal to $\text{SR}(K_i)$. Thus, by recursively using Lemma 1 for $\text{SR}(K_i)$, $i \in [1, d]$, we see that the sequence representation $\text{SR}(K_i)$ is $(n(K_i), s(K_i))$ -admissible. Hence, it follows that the sequence representation $\text{SR}(K)$ is (n, Δ) -admissible.

The only-if part: We prove the converse of (i) by induction on n .

For $n = 1$, M is $(1, \Delta)$ -admissible by the definition of admissibility. Note that M is an empty sequence in this case. Let K be an ordered graph with $n = 1$ vertices and Δ self-loops. Then, $\text{SR}(K)$ is an empty sequence, and hence, $M = \text{SR}(K)$.

Suppose that the converse of (i) holds for any positive integer ℓ . We show that the converse holds for the integer $\ell + 1$. Let M be $(\ell + 1, \Delta)$ -admissible. Then, by the definition of admissibility, there exists an integer $d \in [1, \ell]$ such that $\ell = \sum_{1 \leq i \leq d} a_i$ and $\Delta \geq \sum_{1 \leq i \leq d} b_i$. This implies that for an integer $i \in [1, d]$, we have $a_i \leq \ell$. Further, for each integer $i \in [1, d]$, the subsequence M_i is (a_i, b_i) -admissible by the admissibility of M . This and the inductive hypothesis that the converse of (i) holds for any integer $\ell \geq 1$ imply that for each integer $i \in [1, d]$, there exists an ordered graph H with a_i vertices and b_i self-loops such that $M_i = \text{SR}(H)$. Let K denote the ordered graph with $\ell + 1$ vertices, Δ self-loops, $\deg_K(r_K) = d$, $\Delta - \sum_{1 \leq i \leq d} b_i$ self-loops on the root r_K , and the i -th root subgraph K_i of K be the ordered subgraph H such that $M_i = \text{SR}(H)$. Then, it immediately follows that:

$$\text{SR}(K) = (s(K_1), \dots, s(K_d)) \oplus M_1 \oplus M_2 \oplus \dots \oplus M_d.$$

This means that $M = \text{SR}(K)$ holds, since $((a_1, b_1), \dots, (a_d, b_d)) = (s(K_1), \dots, s(K_d))$, showing that the converse holds for the integer $\ell + 1$.

Hence, by mathematical induction, the converse of (i) holds for any integer $n \geq 1$.

(ii) We prove this result by induction on n .

For $n = 1$, the sequence M is an empty sequence and is $(1, \Delta)$ -admissible by the definition of admissibility. Therefore, it takes constant $\mathcal{O}(1)$ time to test admissibility in this case.

Suppose that for $n = \ell$, $\ell \geq 1$, the admissibility of sequence M can be tested in $\mathcal{O}(\ell)$ time. We show that the statement (ii) holds for $n = \ell + 1$. To show if M is $(\ell + 1, \Delta)$ -admissible, we need to find an integer $d \in [1, \ell]$ such that $\ell = \sum_{1 \leq i \leq d} a_i$ and $\Delta \geq \sum_{1 \leq i \leq d} b_i$. Such an integer d can be identified in $\mathcal{O}(d)$ time. Suppose that such an integer d exists for M . Then, for each integer $i \in [1, d]$, we next need to test if the subsequence M_i is (a_i, b_i) -admissible. Note that the size of the sequence M_i is $a_i - 1$, $i \in [1, d]$. By $\ell = \sum_{1 \leq i \leq d} a_i$, it holds that $a_i \leq \ell$, $i \in [1, d]$. This and the inductive hypothesis imply that for an integer $i \in [1, d]$, the admissibility of the sequence M_i can be tested in $\mathcal{O}(a_i)$ time. Thus, the time testing admissibility of M is $\mathcal{O}(d + \sum_{1 \leq i \leq d} a_i) = \mathcal{O}(d + \ell) = \mathcal{O}(\ell + 1)$, since $d \leq \ell$.

Hence, by mathematical induction, the admissibility of a sequence M of size $n - 1$, $n \geq 1$ can be tested in $\mathcal{O}(n)$ time.

(iii) We prove the claim in (iii) by induction on n .

For $n = 1$, M is $(1, \Delta)$ -admissible and is the sequence representation of the ordered graph K with only one vertex and Δ self-loops. This implies that K can be constructed in $\mathcal{O}(1)$ time.

Suppose that for $n = \ell$, $\ell \geq 1$, the statement (iii) holds. We show that the statement (iii) holds for $n = \ell + 1$. Let M be an $(\ell + 1, \Delta)$ -admissible sequence. Then, there exists an integer $d \in [1, \ell]$, such that $\ell = \sum_{1 \leq i \leq d} a_i$ and $\Delta \geq \sum_{1 \leq i \leq d} b_i$. By (i), there exists an ordered graph K with n vertices and Δ self-loops such that $\text{SR}(K) = M$. Thus, it holds that $\deg_K(r_K) = d$. Further, by Lemma 1, for each integer $i \in [1, d]$, the i -th root-subgraph K_i of K has a_i vertices and b_i self-loops. Observe that such an

integer d exists uniquely due to the admissibility of M . This implies that $\text{deg}_K(r_K)$ can be obtained in $\mathcal{O}(d)$ time.

By $\text{SR}(K) = M$ and Lemma 1, for each integer $i \in [1, d]$, it holds that $\text{SR}(K_i) = M_i$. Recall that the size of $\text{SR}(K_i)$ is $n(K_i) - 1$, which is equal to $a_i - 1$, $i \in [1, d]$. Further, by $\ell = \sum_{1 \leq i \leq d} a_i$, it follows that $a_i \leq \ell$, $i \in [1, d]$. Thus, by the inductive hypothesis for an integer $i \in [1, d]$, the subgraph K_i can be constructed from M_i in $\mathcal{O}(a_i)$ time. Since $d + \sum_{1 \leq i \leq d} a_i = d + \ell$ and $d \leq \ell$, K can be constructed in $\mathcal{O}(\ell + 1)$ time from M .

Hence, by mathematical induction, for integers $n \geq 1$ and $\Delta \geq 0$ and an (n, Δ) -admissible sequence M , the ordered graph K with $\text{SR}(K) = M$ can be constructed by M in $\mathcal{O}(n)$ time. \square

Let $M = ((a_1, b_1), (a_2, b_2), \dots, (a_{n-1}, b_{n-1}))$ be an (n, Δ) -admissible sequence. Then, there exists an integer $d \in [1, n - 1]$ such that $n - 1 = \sum_{1 \leq i \leq d} a_i$. Furthermore, by Theorem 1(i), there exists an ordered graph K with n vertices and Δ self-loops such that $\text{deg}_K(r_K) = d$ and $\text{SR}(K) = M$. Note that such an integer d and ordered graph K are unique. We call the integer d the root-degree of M and denote it by $d(M)$. Moreover, for each integer $i \in [1, d(M)]$, the subsequence of M consisting of $a_i - 1$ consecutive entries starting from $d(M) + \sum_{1 \leq h \leq i-1} a_h - (i - 1) + 1$ is equal to the sequence representation of the i -th root-subgraph of M . For an integer $i \in [1, d(M)]$, we call such a subsequence of M the i -th root-subsequence of M and denote it by $M(i)$.

By Theorem 1(i), it follows that an ordered graph K with $n \geq 1$ vertices and $\Delta \geq 0$ self-loops can be completely determined by $\text{SR}(K)$. Thus, we define a canonical representation of a rooted graph $H \in \mathcal{H}(n, \Delta)$ as follows. For a rooted graph $H \in \mathcal{H}(n, \Delta)$, we define the canonical representation to be an (n, Δ) -admissible sequence M such that M is lexicographically maximum among all (n, Δ) -admissible sequences that are the sequence representation of ordered graphs of H .

In Figure 4d, we show the canonical representation M of the rooted graph $H \in \mathcal{H}(11, 3)$ illustrated in Figure 4a. Further, we show the ordered graph L such that $\text{SR}(L) = M$.

To generate all rooted graphs in $\mathcal{H}(n, \Delta)$, it is enough to generate the canonical representation of each rooted graph $H \in \mathcal{H}(n, \Delta)$ by Theorem 1(i). For two integers $n \geq 1$ and $\Delta \geq 0$, let $\mathcal{M}(n, \Delta)$ denote the set of all (n, Δ) -admissible sequences that are canonical representation of graphs in $\mathcal{H}(n, \Delta)$. Note that the empty sequence is the only sequence in $\mathcal{M}(1, \Delta)$. In the next lemma, we give a characterization of sequences in $\mathcal{M}(n, \Delta)$.

Lemma 2. *Let $n \geq 2$ and $\Delta \geq 0$ be two integers. Let $M = ((a_1, b_1), (a_2, b_2), \dots, (a_{n-1}, b_{n-1}))$ be a sequence of integer pairs with an integer $d \in [1, n - 1]$ such that $n - 1 = \sum_{1 \leq i \leq d} a_i$. For an integer $i \in [1, d]$, let $M(i)$ denote the subsequence of M consisting of $a_i - 1$ consecutive entries starting from $d + \sum_{1 \leq h \leq i-1} a_h - (i - 1) + 1$. Then, $M \in \mathcal{M}(n, \Delta)$ if and only if the following hold:*

- (i) $a_i \geq 1, \forall i \in [1, d], \sum_{1 \leq i \leq d} a_i = n - 1$ and $a_i \geq a_{i+1}, \forall i \in [1, d - 1]$;
- (ii) $b_i \geq 0, \forall i \in [1, d], \sum_{1 \leq i \leq d} b_i \leq \Delta$ and for each integer $i \in [1, d - 1]$ such that $a_i = a_{i+1}$, it holds that $b_i \geq b_{i+1}$; and
- (iii) $M(i) \in \mathcal{M}(a_i, b_i), \forall i \in [1, d]$, and for each integer $i \in [1, d - 1]$ such that $a_i = a_{i+1}$ and $b_i = b_{i+1}$, it holds that $M(i) \succeq M(i + 1)$.

Proof. The if part: Let $M \in \mathcal{M}(n, \Delta)$. Then, by the definition of admissibility, it holds that $d = d(M)$. Let H denote the ordered graph with n vertices and Δ self-loops such that $\text{SR}(H) = M$.

- (i) By the admissibility of M , we have $a_i \geq 1, \forall i \in [1, d(M)], \sum_{1 \leq i \leq d(M)} a_i = n - 1$. Furthermore, M is the canonical representation of H , and therefore, for the sequence representation $((s_1, s'_1), (s_2, s'_2), \dots, (s_{n-1}, s'_{n-1}))$ of any ordered graph of H , it holds that $(a_1, \dots, a_{d(M)}) \succeq (s_1, \dots, s_{d(M)})$. This eventually implies that $a_i \geq a_{i+1}, \forall i \in [1, d - 1]$.
- (ii) By the admissibility of M , it holds that $b_i \geq 0, \forall i \in [1, d(M)], \sum_{1 \leq i \leq d(M)} b_i \leq \Delta$. Moreover, for the sequence representation $((s_1, s'_1), (s_2, s'_2), \dots, (s_{n-1}, s'_{n-1}))$ of any ordered graph of H such that $s_i = a_i, \forall i \in [1, d(M)]$, it holds that $(b_1, \dots, b_{d(M)}) \succeq (s'_1, \dots, s'_{d(M)})$ since M is the canonical

representation of H . This implies that for each integer $i \in [1, d(M) - 1]$ such that $a_i = a_{i+1}$, it holds that $b_i \geq b_{i+1}$.

(iii) We first prove that for each integer $i \in [1, d(M)]$, it holds that $M(i) \in \mathcal{M}(a_i, b_i)$.

For an integer $i \in [1, d(M)]$, let H_i denote the i -th root-subgraph of H following the left-to-right ordering. Then, by Lemma 1 for each integer $i \in [1, d(M)]$, it holds that $SR(H_i) = M(i)$.

Suppose on the contrary that there exists an integer $i \in [1, d(M)]$ such $M(i) \notin \mathcal{M}(a_i, b_i)$. This means that there exists an ordered graph L that is rooted isomorphic to H_i , and $SR(L) \succ M(i)$ holds. Let M' denote the sequence obtained from M by replacing the subsequence M_i with $SR(L)$. Clearly, M' is (n, Δ) -admissible, and $M' \succ M$ holds by the construction of M' . Let H' denote the ordered graph obtained by replacing H_i with L in H and preserving the ordering of the children of each vertex in L . Then, we see that H' is rooted isomorphic to H , and $SR(H') = M'$. This contradicts the fact that M is a sequence in $\mathcal{M}(n, \Delta)$. Hence, for each integer $i \in [1, d(M)]$, it holds that $M(i) \in \mathcal{M}(a_i, b_i)$.

Recall that M is the canonical representation of H . This implies that for the sequence representation $((s_1, s'_1), (s_2, s'_2), \dots, (s_{n-1}, s'_{n-1}))$ of any ordered graph of H such that $(s_i, s'_i) = (a_i, b_i), \forall i \in [1, d(M)]$, it holds that $((a_{d(M)+1}, b_{d(M)+1}), \dots, (a_{n-1}, b_{n-1})) \succeq ((s_{d(M)+1}, s'_{d(M)+1}), \dots, (s_{n-1}, s'_{n-1}))$. This implies that for each integer $i \in [1, d(M) - 1]$ such that $a_i = a_{i+1}$ and $b_i = b_{i+1}$, we have $M(i) \succeq M(i + 1)$.

The only-if part: Let M satisfy (i), (ii), and (iii). We show that $M \in \mathcal{M}(n, \Delta)$. To prove this, we show that M is a canonical representation of some graph in $\mathcal{H}(n, \Delta)$.

By (i) and (ii), we have $a_i \geq 1, b_i \geq 0, \forall i \in [1, d], n - 1 = \sum_{1 \leq i \leq d} a_i$ and $\Delta \geq \sum_{1 \leq i \leq d} b_i$. Furthermore, for each integer $i \in [1, d]$, the sequence M_i is (a_i, b_i) -admissible, since $M(i) \in \mathcal{M}(a_i, b_i)$ by (iii). This implies that M is (n, Δ) -admissible.

By Theorem 1(i), there exists a unique ordered graph $K = (H, \pi)$ such that $SR(K) = M$ for some $H \in \mathcal{H}(n, \Delta)$. This implies that $\deg_K(r_K) = d$, and for each integer $i \in [1, d]$, the i -th root subgraph of K has a_i vertices and b_i self-loops. This implies that any ordered graph L that is rooted isomorphic to K has x vertices and y self-loops such that $(x, y) = (a_i, b_i)$ for some $i \in [1, d]$.

The condition $a_i \geq a_{i+1}, \forall i \in [1, d - 1]$ in (i) implies that for the sequence representation $S = ((s_1, s'_1), (s_2, s'_2), \dots, (s_{n-1}, s'_{n-1}))$ of any ordered graph that is rooted isomorphic to K , it holds that $M \succeq S$ since $(a_1, \dots, a_d) \succeq (s_1, \dots, s_d)$. For the condition for each integer $i \in [1, d - 1]$ such that $a_i = a_{i+1}$, it holds that $b_i \geq b_{i+1}$ in (ii) implies that for the sequence representation $S = ((s_1, s'_1), (s_2, s'_2), \dots, (s_{n-1}, s'_{n-1}))$ of any ordered graph that is rooted isomorphic to K such that $s_i = a_i, \forall i \in [1, d]$, it holds that $M \succeq S$ since $(b_1, \dots, b_d) \succeq (s'_1, \dots, s'_d)$. Finally, for the condition for each integer $i \in [1, d - 1]$ such that $a_i = a_{i+1}$ and $b_i = b_{i+1}$, it holds that $M(i) \succeq M(i + 1)$ in (iii) implies that for the sequence representation $S = ((s_1, s'_1), (s_2, s'_2), \dots, (s_{n-1}, s'_{n-1}))$ of any ordered graph that is rooted isomorphic to K and $(s_i, s'_i) = (a_i, b_i), \forall i \in [1, d]$, it holds that $M \succeq S$ since $((a_{d+1}, b_{d+1}), \dots, (a_{n-1}, b_{n-1})) \succeq ((s_{d+1}, s'_{d+1}), \dots, (s_{n-1}, s'_{n-1}))$. This eventually implies that M is the canonical representation of H from which it follows that $M \in \mathcal{M}(n, \Delta)$. \square

We next give the structure of the sequences that are lexicographically minimum and maximum among all sequences in $\mathcal{M}(n, \Delta)$.

Lemma 3. *Let $n \geq 2$ and $\Delta \geq 0$ be two integers.*

- (i) *The sequences $N = ((1, 0), (1, 0), \dots, (1, 0))$ and $M = ((n - 1, \Delta), (n - 2, \Delta), \dots, (1, \Delta))$ each of length $n - 1$ are lexicographically minimum and maximum among all sequences in $\mathcal{M}(n, \Delta)$, respectively.*
- (ii) *Whether a sequence in $\mathcal{M}(n, \Delta)$ is lexicographically minimum or maximum among all sequences in $\mathcal{M}(n, \Delta)$ can be tested in $\mathcal{O}(n)$.*

Proof. (i) It is easy to observe that the sequence N is (n, Δ) -admissible. Furthermore, for two integers $i \geq 2$ and $j \geq 0$, the ranges of the first and the second entries in any sequence in $\mathcal{M}(i, j)$ are $[1, i - 1]$

and $[0, j]$, respectively. This implies that the sequence N is lexicographically minimum among all the sequences in $\mathcal{M}(n, \Delta)$. Moreover, the sequence $((1, \Delta))$ is admissible and lexicographically maximum among all the sequences in $\mathcal{M}(2, \Delta)$. From this, it follows that the sequence $((2, \Delta), (1, \Delta))$ is admissible and lexicographically maximum among all the sequences in $\mathcal{M}(3, \Delta)$. Thus, by using this inductive argument, we can conclude that the sequence M is admissible and lexicographically maximum among all the sequences in $\mathcal{M}(n, \Delta)$.

(ii) We know that a sequence S in $\mathcal{M}(n, \Delta)$ is of length $n - 1$, and therefore, by using a for-loop of size $n - 1$ and (i), we can test if S is lexicographically minimum or maximum among all sequences in $\mathcal{M}(n, \Delta)$ in $\mathcal{O}(n)$ time. \square

Let $S(n, \Delta)$ and $L(n, \Delta)$ denote the lexicographically minimum and maximum sequences among all sequences in $\mathcal{M}(n, \Delta)$, respectively.

For a sequence $M \in \mathcal{M}(n, \Delta)$ such that $M \neq S(n, \Delta)$, we define the predecessor $P(M)$ of M to be the sequence that is lexicographically maximum among all sequences that are lexicographically smaller than M , i.e., there does not exist a sequence $N \in \mathcal{M}(n, \Delta) \setminus \{P(M)\}$ such that $M \succ N \succ P(M)$ holds.

For a sequence $M \in \mathcal{M}(n, \Delta)$, we next give the structure of the predecessor $P(M)$, if it exists, of M .

Theorem 2. Let $n \geq 2$ and $\Delta \geq 0$ be two integers and $M = ((a_1, b_1), (a_2, b_2), \dots, (a_{n-1}, b_{n-1}))$ be a sequence in $\mathcal{M}(n, \Delta)$. Let d denote the root-degree of M . Then, for the predecessor $P(M) = ((x_1, y_1), (x_2, y_2), \dots, (x_{n-1}, y_{n-1}))$, if it exists, we have:

- (a) If $a_i = 1$ and $b_i = 0, \forall i \in [1, d]$, then $P(M)$ does not exist.
- (b) If $a_i \neq 1$ for some $i \in [1, d], b_j = 0$ and $M(j) = S(a_j, b_j), \forall j \in [1, d]$. Then, for the largest integer $k \in [1, d]$ such that $a_k \neq 1$, it holds that $d(P(M)) = k - 1 + \lceil (a_k + d - k) / (a_k - 1) \rceil, y_1 = \Delta, y_i = 0, \forall i \in [2, d(P(M))], x_i = a_i, \forall i \in [1, k - 1], x_i = a_k - 1, \forall i \in [k, d(P(M)) - 1], x_{d(P(M))} = a_k + d - k - \lfloor (a_k + d - k) / (a_k - 1) \rfloor (a_k - 1)$, and $P(M)(i) = L(x_i, y_i), \forall i \in [1, d(P(M))]$.
- (c) If $b_i \neq 0$ for some $i \in [1, d], M(j) = S(a_j, b_j), \forall j \in [1, d]$. For the largest integer $k \in [1, d]$ such that $b_k \neq 0$, let $p \triangleq \max\{i \leq d \mid a_k = a_{k+i}\}, q \triangleq p + 1$ if $b_k = 1$, and $q \triangleq \lfloor (\Delta - \sum_{1 \leq i \leq k-1} b_i) / (b_k - 1) \rfloor$ if $b_k \geq 2$ and $t \triangleq \min\{q, p + 1\}$. Then, it holds that $d(P(M)) = d, x_i = a_i, \forall i \in [1, d], y_i = b_i, \forall i \in [1, k - 1], y_k = b_k - 1$; for $k \neq d$, we have $y_{k+i} = b_k - 1, \forall i \in [1, t - 1], y_{k+t} = (\Delta - \sum_{1 \leq i \leq k-1} b_i) - t(b_k - 1), y_i = 0, \forall i \in [k + t + 1, d]$, and $P(M)(i) = L(x_i, y_i), \forall i \in [1, d]$.
- (d) Otherwise if $a_i \neq 1, b_j \neq 0, M(\ell) = S(a_\ell, b_\ell)$, for some $i, j, \ell \in [1, d]$. For the largest integer $k \in [1, d]$ such that $M(k) \neq S(a_k, b_k)$, let $p \triangleq \max\{i \leq d \mid a_k = a_{k+i} \text{ and } b_k = b_{k+1}\}$. Then, it holds that $d(P(M)) = d, (x_i, y_i) = (a_i, b_i), \forall i \in [1, d], P(M)(i) = M(i), \forall i \in [1, k - 1], P(M)(k + i) = P(M(k)), \forall i \in [0, p]$, and $P(M)(i) = L(a_i, b_i), \forall i \in [k + p + 1, d]$.

Proof. (a) If $a_i = 1$ and $b_i = 0, \forall i \in [1, d]$, then it holds that $d = n - 1$. Thus, by Lemma 3(i), it holds that $M = S(n, \Delta)$, and therefore, $P(M)$ does not exist.

(b) If $a_i \neq 1$ for some $i \in [1, d], b_j = 0$ and $M(j) = S(a_j, b_j), \forall j \in [1, d]$, then M is lexicographically minimum among all those sequences $((c_1, c'_1), \dots, (c_{n-1}, c'_{n-1})) \in \mathcal{M}(n, \Delta)$ for which it holds that $c_i = a_i, \forall i \in [1, d]$. Further, $a_i \neq 1$ for some $i \in [1, d]$ implies that $P(M)$ exists. By the definition of a predecessor, observe that $P(M)$ is lexicographically maximum among all those sequences $S = ((s_1, s'_1), \dots, (s_{n-1}, s'_{n-1})) \in \mathcal{M}(n, \Delta)$ for which it holds that $s_i = x_i, \forall i \in [1, d(P(M))]$. This implies that for each such sequence S , it holds that either $(y_1, y_2, \dots, y_{d(P(M))}) \succ (s'_1, s'_2, \dots, s'_{d(P(M))})$ or $y_i = s'_i$ and $P(M)(i) \succeq S(i), \forall i \in [1, d(P(M))]$. The former implies that $y_1 = \Delta, y_i = 0, \forall i \in [2, d(P(M))]$, while the latter implies that $P(M)(i) = L(x_i, y_i), \forall i \in [1, d(P(M))]$. Further, the sequence $(x_1, \dots, x_{d(P(M))})$ satisfies Lemma 2(i), and there does not exist a sequence L that satisfies Lemma 2(i) such that $(x_1, \dots, x_{d(P(M))}) \prec L \prec (a_1, \dots, a_d)$ holds by the definition of $P(M)$. This and the definition of k imply that $x_i = a_i, \forall i \in [1, k - 1]$; the sequence $(x_k, x_{k+1}, \dots, x_{d(P(M))})$ is a non-decreasing

sequence, and there does not exist a sequence $Z = (z_1, \dots, z_t)$ such that $z_i \in [1, a_k - 1], \forall i \in [1, t]$ and $\sum_{1 \leq i \leq t} z_i = \sum_{k \leq j \leq d} a_j$ for which it holds that $(x_k, \dots, x_{d(P(M))}) \prec Z \prec (a_k, \dots, a_d)$ holds. This eventually implies that $d(P(M)) = k - 1 + \lceil \sum_{k \leq i \leq d} a_i / (a_k - 1) \rceil, x_i = a_k - 1, \forall i \in [k, d(P(M)) - 1]$, and $x_{d(P(M))} = \sum_{k \leq i \leq d} a_i - \lfloor \sum_{k \leq i \leq d} a_i / (a_k - 1) \rfloor (a_k - 1)$. Thus, by the definition of k , we have $\sum_{k \leq i \leq d} a_i = a_k + d - k$, and therefore, we have the required result.

(c) If $b_i \neq 0$ for some $i \in [1, d]$ and $M(j) = S(a_j, b_j), \forall j \in [1, d]$, then M is lexicographically minimum among all those sequences $((c_1, c'_1), \dots, (c_{n-1}, c'_{n-1})) \in \mathcal{M}(n, \Delta)$ for which it holds that $(c_i, c'_i) = (a_i, b_i), \forall i \in [1, d]$. Since $b_i \neq 0$ for some $i \in [1, d]$, therefore $P(M)$ exists and is lexicographically maximum among all those sequences $((s_1, s'_1), \dots, (s_{n-1}, s'_{n-1})) \in \mathcal{M}(n, \Delta)$ for which it holds that $(s_i, s'_i) = (a_i, y_i), \forall i \in [1, d(P(M))]$. This implies that $x_i = a_i, \forall i \in [1, d(P(M))]$, and therefore, we have $d(P(M)) = d$. Furthermore, $(y_1, y_2, \dots, y_d) \succ (s'_1, s'_2, \dots, s'_d)$ and $P(M)(i) \succeq S(i), \forall i \in [1, d]$. This implies that for each integer $i \in [1, d]$, it holds that $P(M)(i) = L(a_i, y_i)$, and the sequence (y_1, y_2, \dots, y_d) is lexicographically minimum among all those sequences that satisfy Lemma 2(ii) and are lexicographically smaller than (b_1, b_2, \dots, b_d) . This and the definition of k imply that $y_i = b_i, \forall i \in [1, k - 1]$; by Lemma 2(ii), the sequence $(y_k, y_{k+1}, \dots, y_{k+p})$ is a non-decreasing sequence, and there does not exist a sequence $Z = (z_k, z_{k+1}, \dots, z_{k+p})$ such that $z_i \in [0, b_k - 1]$ and $\sum_{k \leq i \leq k+p} z_i = \min\{(p + 1)(b_k - 1), \Delta - \sum_{1 \leq i \leq k-1} b_i\}$ for which it holds that $(y_k, y_{k+1}, \dots, y_{k+p}) \prec Z \prec (b_k, b_{k+1}, \dots, b_{k+p})$. This eventually implies that $y_k = b_k - 1$ and for $k \neq d, y_{k+i} = b_k - 1, \forall i \in [1, t - 1]$. This and the minimality of (y_1, y_2, \dots, y_d) imply that $y_{k+t} = (\Delta - \sum_{1 \leq i \leq k-1} b_i) - t(b_k - 1)$ and $y_i = 0, \forall i \in [k + t + 1, d]$.

(d) The conditions of this case imply that M is not the lexicographically minimum among all those sequences $((c_1, c'_1), \dots, (c_{n-1}, c'_{n-1})) \in \mathcal{M}(n, \Delta)$ for which it holds that $(c_i, c'_i) = (a_i, b_i), \forall i \in [1, d]$. This implies that for $P(M)$, it holds that $d(P(M)) = d$ and $(x_i, y_i) = (a_i, b_i), \forall i \in [1, d]$, and there does not exist a sequence $S = ((s_1, s'_1), \dots, (s_{n-1}, s'_{n-1})) \in \mathcal{M}(n, \Delta)$ for which it holds that $(s_i, s'_i) = (a_i, b_i), \forall i \in [1, d]$ and $((x_1, y_1), \dots, (x_{n-1}, y_{n-1})) \prec S \prec M$. This implies that for each such sequence S , it holds that $P(M)(1) \oplus P(M)(2) \oplus \dots \oplus P(M)(d) \succ S(1) \oplus S(2) \oplus \dots \oplus S(d)$. This and the definition of k imply that $P(M)(i) = M(i), \forall i \in [1, k - 1]$. Furthermore, by Lemma 2(iii) and the definition of p , it holds that $P(M)(i) \succ P(M)(i + 1), \forall i \in [k, k + p - 1]$, and there does not exist a sequence $S = ((s_1, s'_1), \dots, (s_{n-1}, s'_{n-1})) \in \mathcal{M}(n, \Delta)$ such that $(s_i, s'_i) = (a_i, b_i), \forall i \in [1, d]$ and $S(i) \succ S(i + 1), \forall i \in [k, k + p - 1]$ for which it holds that $P(M)(k) \oplus P(M)(k + 1) \oplus \dots \oplus P(M)(K + 1) \prec S(k) \oplus S(k + 1) \oplus \dots \oplus S(k + p + 1) \prec M(k) \oplus M(k + 1) \oplus \dots \oplus M(k + p + 1)$. Since $M(k) \neq S(a_k, b_k)$, therefore $P(M)(k)$ is a lexicographically minimum sequence in $\mathcal{M}(a_k, b_k)$ for which it holds that $P(M)(k) \prec M(k)$. This implies that $P(M)(k + i) = P(M)(k), \forall i \in [0, p]$. Further, by the minimality of $P(M)$, $P(M)(i) = L(a_i, b_i), \forall i \in [k + p + 1, d]$.

Finally, one can easily verify that the sequence $((x_1, y_1), \dots, (x_{n-1}, y_{n-1}))$ obtained in each of the above cases satisfies Lemma 2(i)–(iii) by construction, and hence, $((x_1, y_1), \dots, (x_{n-1}, y_{n-1}))$ is an element of $\mathcal{M}(n, \Delta)$ that is $P(M)$, which completes the proof. \square

Lemma 4. Let $n \geq 2$ and $\Delta \geq 0$ be two integers and M be a sequence in $\mathcal{M}(n, \Delta)$. Then, the predecessor $P(M)$, if it exists, can be computed in $\mathcal{O}(n)$ time and $\mathcal{O}(n)$ space.

The proof of Lemma 4 follows from Algorithm 1 and Lemma 5.

We next present Algorithm 1 to compute the predecessor based on Theorem 2. In this algorithm, for a sequence $M \in \mathcal{M}(n, \Delta)$ with root-degree d and integer $i \in [1, d]$, the variable $M[i]$ stores the i -th root-subsequence $M(i)$ of M , and the variable $P[M]$ stores the predecessor, if it exists, of M .

Algorithm 1 Computing the predecessor of an admissible sequence.

Input: Two integers $n \geq 2$ and $\Delta \geq 0$ and an (n, Δ) -admissible sequence

$$M = ((a_1, b_1), \dots, (a_{n-1}, b_{n-1})).$$

Output: The predecessor M if it exists; The predecessor of M does not exist otherwise.

```

1:  $d :=$  The root-degree of  $M$ ;
2: if  $a_i = 1$  and  $b_i = 0, \forall i \in [1, d]$  then
3:   Output The predecessor of  $M$  does not exist /* Theorem 2(a) */
4: else /* The predecessor of  $M$  exists by Theorem 2 */
5:    $P[M] := ((x_1, y_1), \dots, (x_{n-1}, y_{n-1}))$ 
6:   if  $a_i \neq 1$  for some  $i \in [1, d]$ , and  $b_j = 0$  and  $M[j] = S(a_j, b_j), \forall j \in [1, d]$  then
   /* Theorem 2(b) */
7:      $k := \max\{i \mid a_i \neq 1\}; h := k - 1 + \lceil (a_k + d - k) / (a_k - 1) \rceil$ ; /* The root-degree
        $d(P(M))$  of  $P(M)$  */
8:      $y_1 := \Delta; y_i := 0, \forall i \in [2, h]; x_i := a_i, \forall i \in [1, k - 1]; x_i := a_k - 1, \forall i \in [k, h - 1];$ 
        $x_h := a_k + d - k - \lfloor (a_k + d - k) / (a_k - 1) \rfloor (a_k - 1); P[M][i] = L(x_i, y_i), \forall i \in [1, h].$ 
9:   else if  $b_i \neq 0$  for some  $i \in [1, d]$  and  $M[j] = S(a_j, b_j), \forall j \in [1, d]$  then
   /* Theorem 2(c) */
10:     $k := \max\{i \mid b_i \neq 0\}; p := \max\{i \leq d \mid a_k = a_{k+i}\}; q := p + 1$  if  $b_k = 1$ ;
        $q := \lfloor (\Delta - \sum_{1 \leq i \leq k-1} b_i) / (b_k - 1) \rfloor$  if  $b_k \geq 2; t := \min\{q, p + 1\}$ ;
11:     $x_i := a_i, \forall i \in [1, d]; y_i := b_i, \forall i \in [1, k - 1]; y_k := b_k - 1$ ;
12:    if  $k \neq d$  then
13:       $y_{k+i} := b_k - 1, \forall i \in [1, t - 1]; y_{k+t} := (\Delta - \sum_{1 \leq i \leq k-1} b_i) - t(b_k - 1);$ 
        $y_i = 0, \forall i \in [k + t + 1, d]$ 
14:    end if;
15:     $P[M][i] := L(x_i, y_i), \forall i \in [1, d]$ 
16:  else /* If  $a_i \neq 1, b_j \neq 0$  and  $M[\ell] \neq S(a_\ell, b_\ell)$ , for some  $i, j, \ell \in [1, d]$  */
   /* Theorem 2(d) */
17:     $k := \max\{i \mid M[i] \neq S(a_i, b_i)\}$ ;
18:     $p := \max\{i \leq d \mid a_k = a_{k+i} \text{ and } b_k = b_{k+1}\}$ ;
        $(x_i, y_i) := (a_i, b_i), \forall i \in [1, d]; P[M][i] := M[i], \forall i \in [1, k - 1];$ 
19:     $P[M][k] := \text{Algorithm 1}(a_k, b_k, M[k]);$ 
20:     $P[M][k + i] := P[M][k], \forall i \in [1, p]; P[M][i] := L(a_i, b_i), \forall i \in [k + p + 1, d]$ 
21:  end if
22:  Output  $P[M]$  as the predecessor of  $M$ ;
23: end if.
```

Lemma 5. For two integers $n \geq 2$ and $\Delta \geq 0$ and an (n, Δ) -admissible sequence M , Algorithm 1 outputs the predecessor $P(M)$, if it exists, in $\mathcal{O}(n)$ time and $\mathcal{O}(n)$ space.

Proof. Correctness: The correctness of Algorithm 1 immediately follows from Theorem 2.

Complexity analysis: By the definition of the root-degree, we can compute d at Line 1 in $\mathcal{O}(n)$ time.

We can test if $a_i \neq 1$ and $b_j \neq 0$ hold for some $i, j \in [1, d]$ in $\mathcal{O}(n)$ time. Similarly, we can test if $M(j) = S(a_j, b_j)$, for some $j \in [1, d]$ in $\mathcal{O}(n)$ time, since the length of $M(j)$ is $a_j - 1$, and $\sum_{1 \leq i \leq d} a_i = n - 1$. Hence, we can test the conditions at Lines 2, 6, 9, and 16 in $\mathcal{O}(n + n) = \mathcal{O}(n)$ time. This implies that we can check if the predecessor of M exists in $\mathcal{O}(n)$ time. We next discuss the time complexity of computing the predecessor in each of the cases at Lines 6, 9, and 16.

When the conditions at Line 6 hold, then k and h can be computed in $\mathcal{O}(n)$ time, since $k \leq d$ and $h \leq d + 1$. This implies that $((x_1, y_1), \dots, (x_h, y_h))$ can be computed in $\mathcal{O}(n)$ time. Furthermore, we can

compute $L(x_i, y_i), \forall i \in [1, h]$ in $\mathcal{O}(x_i)$ time by Lemma 3(i). Recall that for $P[M]$, it holds that $\sum_{1 \leq i \leq d} x_i = n - 1$. Thus, $P[M]$ can be computed in $\mathcal{O}(n)$ time.

When the conditions at Line 9 hold, then k, p, q , and t can be computed in $\mathcal{O}(n)$ time by these definitions. Thus, $((x_1, y_1), \dots, (x_d, y_d))$ can be computed from Line 11 to Line 14 in $\mathcal{O}(n)$ time. Furthermore, $L(x_i, y_i), \forall i \in [1, d]$ can be computed at Line 15 in $\mathcal{O}(n)$ time, as discussed above. This implies that $P[M]$ can be computed in $\mathcal{O}(n)$ time in this case.

Finally, when the conditions at Line 16 hold, then once again, we can compute k in $\mathcal{O}(n)$ time, since $\sum_{1 \leq i \leq d} a_i = n - 1$. Further, p can be computed in $\mathcal{O}(n)$ time by the definition of p . However, $P[M][k]$ can be obtained by recursively running Algorithm 1 on a_k, b_k and $M[k]$. Note that this operation is repeated at most the length of the sequence $P[M][k]$, which is $a_k - 1$ in this case, and hence, $P[M][k]$ can be computed in $\mathcal{O}(n)$ time. Once again, the computation at Line 20 can be done in $\mathcal{O}(n)$ time since it holds that $\sum_{1 \leq i \leq d} a_i = n - 1$. Hence, $P[M]$ can be computed in $\mathcal{O}(n)$ time.

Note that the $\mathcal{O}(n)$ space is sufficient to store $P[M]$, if it exists, which completes the proof. \square

Note that we can generate all rooted graphs in $\mathcal{H}(n, \Delta)$ by generating their canonical representation by repeatedly using Algorithm 1 starting from the lexicographically maximum sequence $L(n, \Delta)$ in $\mathcal{M}(n, \Delta)$ in $\mathcal{O}(n)$ time per graph and $\mathcal{O}(n)$ space in total.

Theorem 3. *Let $n \geq 2$ and $\Delta \geq 0$ be two integers. Then, all mutually non-isomorphic graphs with n vertices, Δ self-loops, and a tree skeleton can be generated in $\mathcal{O}(n)$ time per graph and $\mathcal{O}(n)$ space in total.*

Proof. A tree can be viewed as a rooted tree by considering its centroid as the root [24]. We know that when n is odd, then there are only trees with unicentroids; however, when n is even, then there are trees with unicentroids.

By the definition of a unicentroid, all mutually non-isomorphic graphs with n vertices, Δ self-loops, and a tree skeleton with a unicentroid can be enumerated by generating all graphs H in $\mathcal{H}(n, \Delta)$ such that each root subgraph of H has at most $\lfloor (n - 1)/2 \rfloor$ number of vertices, i.e., by generating all sequences $M = ((a_1, b_1), \dots, (a_{n-1}, b_{n-1})) \in \mathcal{M}(n, \Delta)$ such that $a_i \in [1, \lfloor (n - 1)/2 \rfloor], \forall i \in [1, d(P(M))]$. Let S denote the sequence that is lexicographically maximum among all those sequences in $\mathcal{M}(n, \Delta)$ that represent a tree with a unicentroid. When n is even, then it holds that $S = ((\lfloor (n - 1)/2 \rfloor, \Delta), (\lfloor (n - 1)/2 \rfloor, 0), (1, 0)) \oplus L(\lfloor (n - 1)/2 \rfloor, \Delta) \oplus L(\lfloor (n - 1)/2 \rfloor, 0) \oplus L(1, 0)$. Recall that $L(1, 0)$ is an empty sequence, and therefore, we have $S = ((\lfloor (n - 1)/2 \rfloor, \Delta), (\lfloor (n - 1)/2 \rfloor, 0), (1, 0)) \oplus L(\lfloor (n - 1)/2 \rfloor, \Delta) \oplus L(\lfloor (n - 1)/2 \rfloor, 0)$. However, when n is odd, then it holds that $S = ((\lfloor (n - 1)/2 \rfloor, \Delta), (\lfloor (n - 1)/2 \rfloor, 0)) \oplus L(\lfloor (n - 1)/2 \rfloor, \Delta) \oplus L(\lfloor (n - 1)/2 \rfloor, 0)$. Hence, we can generate all sequences in $\mathcal{M}(n, \Delta)$ that represent a graph $H \in \mathcal{H}(n, \Delta)$ such that the skeleton of H has a unicentroid by repeatedly using Theorem 2 starting from the sequence S . This implies that we can generate all such sequences in $\mathcal{O}(n)$ time per sequence and $\mathcal{O}(n)$ space by using Algorithm 1.

When n is even, then all mutually non-isomorphic graphs with n vertices, Δ self-loops, and a tree skeleton with a bicentroid can be enumerated by generating all sequences $M = ((a_1, b_1), \dots, (a_{n-1}, b_{n-1})) \in \mathcal{M}(n, \Delta)$ such that the root-degree $d(M) = 2, a_1 = a_2 = n/2$ and $b_1 + b_2 = \Delta$ with $b_1 \geq b_2$. In such a case, the sequences $((n/2, \Delta), (n/2, 0)) \oplus L(n/2, \Delta) \oplus L(n/2, 0)$ and $((n/2, \lceil \Delta/2 \rceil), (n/2, \lfloor \Delta/2 \rfloor)) \oplus L(n/2, \lceil \Delta/2 \rceil) \oplus L(n/2, \lfloor \Delta/2 \rfloor)$ are lexicographically maximum and minimum, respectively, among all those sequences in $\mathcal{M}(n, \Delta)$ that represent a graph with a bicentroid. Let $M = ((a_1, b_1), \dots, (a_{n-1}, b_{n-1})) \in \mathcal{M}(n, \Delta)$ be a sequence that represents a graph $H \in \mathcal{H}(n, \Delta)$ such that the skeleton of H has a bicentroid and $M \neq ((n/2, \lceil \Delta/2 \rceil), (n/2, \lfloor \Delta/2 \rfloor)) \oplus L(n/2, \lceil \Delta/2 \rceil) \oplus L(n/2, \lfloor \Delta/2 \rfloor)$. When $M(i) = S(a_j, b_j), \forall i \in \{1, 2\}$, then it holds that $P(M) = ((a_1, b_1 - 1), (a_2, b_2 + 1)) \oplus L(a_1, b_1 - 1) \oplus L(a_2, b_2 + 1)$. However, in the case otherwise, i.e., when $M(i) \neq S(a_j, b_j)$, for some $i \in \{1, 2\}$, then $P(M)$ can be generated by using Theorem 2(d). Clearly, in both of these cases, $P(M)$ can be generated in $\mathcal{O}(n)$ time and $\mathcal{O}(n)$ space. This eventually implies that all sequences that represent a graph in $\mathcal{H}(n, \Delta)$ with a bicentroid can be generated in $\mathcal{O}(n)$ time per sequence and $\mathcal{O}(n)$ space by repeatedly computing the predecessor of sequences M as described above

starting from $((n/2, \Delta), (n/2, 0)) \oplus L(n/2, \Delta) \oplus L(n/2, 0)$ until $M = ((n/2, \lceil \Delta/2 \rceil), (n/2, \lfloor \Delta/2 \rfloor)) \oplus L(n/2, \lceil \Delta/2 \rceil) \oplus L(n/2, \lfloor \Delta/2 \rfloor)$.

Hence, we can generate all non-isomorphic graphs with n vertices, Δ self-loops, and a tree skeleton with a un centroid or bicentroid in $\mathcal{O}(n)$ time per graph and $\mathcal{O}(n)$ space in total, which completes that proof. \square

4. Results and Discussion

We computed all graphs with n vertices, Δ self-loops, and a tree skeleton for different values of n and Δ to test the efficiency of our algorithm, and the results are listed in Table 1. These experiments were performed on a PC with an Intel Core i7-500 processor, running at 2.70 GHz, 16 GB of memory, and Windows 10. From the experimental results, it is evident that the proposed method is computationally efficient.

Table 1. Experimental results of the enumeration method.

(n, Δ)	# of Generated Graphs	Time (s)
(5, 9)	856	0.278
(5, 10)	1186	0.213
(5, 30)	50,596	4.354
(6, 9)	4270	0.992
(6, 10)	6333	1.571
(6, 30)	619,431	141.334
(7, 9)	20,548	5.084
(7, 10)	32,337	7.047
(8, 9)	95,357	17.444
(8, 10)	159,058	31.755
(9, 9)	429,496	88.899
(9, 10)	756,045	185.823
(10, 9)	1,882,764	528.286
(10, 10)	3,488,567	914.806
(17, 2)	25,939,679	3911.33
(18, 0)	123,867	34.189
(20, 0)	823,065	334.357
(22, 0)	5,623,756	1807.53
(24, 0)	39,299,897	8042.88

An application to the generation of polymer topologies with a tree skeleton: Observe that a graph with a tree skeleton, Δ self-loops, and no multiple edges has cycle rank Δ . Therefore, the class of such graphs contains all polymer topologies of cycle rank Δ with a tree skeleton. However, it is a natural question to search for a relationship between the number n of vertices and the number Δ of self-loops such that there exists a polymer topology with a tree skeleton, n vertices, and cycle rank Δ . Clearly, for $n = 1$ and $\Delta \geq 2$, there exists exactly one polymer topology with a tree skeleton, n vertices, and Δ self-loops. Let $\mathcal{P}(n, \Delta)$ denote a maximal set of mutually non-isomorphic polymer topologies with a tree skeleton, n vertices, Δ self-loops, and no multiple edges. Azam et al. [21] proved the following necessary condition on Δ to have a polymer topology with a tree skeleton and n vertices.

Lemma 6 ([21]). *If $n \geq 1$ and $\Delta \geq \lceil \frac{n}{2} \rceil + 1$, then it holds that $\mathcal{P}(n, \Delta) \neq \emptyset$.*

Let $\mathcal{G}(n, \Delta)$ denote a maximal set of mutually non-isomorphic graphs with a tree skeleton, n vertices, and Δ self-loops. For an integer $r \geq 1$, let $\mathcal{P}(r)$ denote a maximal set of mutually non-isomorphic polymer topologies with a tree skeleton, n vertices, and r self-loops. By Lemma 6, it holds that:

$$\mathcal{P}(r) = \bigcup_{n \in \mathbb{Z}^+ : \lceil \frac{n}{2} \rceil + 1 \leq r} \mathcal{P}(n, r) \subseteq \bigcup_{n \in \mathbb{Z}^+ : \lceil \frac{n}{2} \rceil + 1 \leq r} \mathcal{G}(n, r) \tag{1}$$

Thus, by using Equation (1) and identifying the degree of each vertex in the graphs in $\mathcal{G}(n, r)$ from their canonical representations, we can compute all polymer topologies in $\mathcal{P}(r)$. We applied our method to generate all polymer topologies in $\mathcal{P}(r)$ for rank $r = 2, 3, \dots, 9$, and the results are listed in Table 2.

Table 2. The number of polymer topologies with a tree skeleton, r self-loops, and no multiple edges.

n	Rank r								
	2	3	4	5	6	7	8	9	
1	1	1	1	1	1	1	1	1	
2	1	1	2	2	3	3	4	4	
3	–	1	2	4	6	9	12	16	
4	–	1	3	6	13	21	35	51	
5	–	–	2	7	18	40	77	136	
6	–	–	1	6	23	61	147	300	
7	–	–	–	3	20	76	223	559	
8	–	–	–	1	14	74	288	868	
9	–	–	–	–	5	54	291	1128	
10	–	–	–	–	2	29	241	1212	
11	–	–	–	–	–	10	145	1057	
12	–	–	–	–	–	2	68	733	
13	–	–	–	–	–	–	19	390	
14	–	–	–	–	–	–	4	151	
15	–	–	–	–	–	–	–	38	
16	–	–	–	–	–	–	–	6	
Total	2	4	11	30	105	308	1555	6650	

5. Conclusions

We proposed an efficient method to enumerate all mutually non-isomorphic graphs with a tree skeleton, a given number of vertices, and the number of self-loops. The idea of this method is to generate rooted graphs with n vertices and Δ self-loops by generating their canonical representation. We defined the canonical representation of a rooted graphs H with n vertices and Δ self-loops based on the ordered graphs of H . The proposed method generates all graphs with a tree skeleton, n vertices, and Δ self-loops in $\mathcal{O}(n)$ time per tree and $\mathcal{O}(n)$ space in total. As an application, we can generate all polymer topologies with a tree skeleton, self-loops, no multiple edges, and a given cycle rank.

An interesting future research direction is to design a method that can directly count and enumerate all mutually non-isomorphic polymer topologies with a given cycle rank.

Author Contributions: Conceptualization, N.A.A. and H.N.; funding acquisition, N.A.A.; methodology, N.A.A. and H.N.; software, N.A.A.; supervision, H.N.; validation, N.A.A., A.S., and H.N.; writing, original draft, N.A.A.; writing, review and editing, N.A.A. and A.S. All authors read and agreed to the published version of the manuscript.

Funding: This research is partially funded by JSPS KAKENHI Grant Number 18J23484.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Azam, N.A.; Shurbevski, A.; Nagamochi, H. A method for enumerating pairwise compatibility graphs with a given number of vertices. In *Discrete Applied Mathematics*; Elsevier: Amsterdam, The Netherlands, 2020.
2. Azam, N.A.; Shurbevski, A.; Nagamochi, H. On the Enumeration of Minimal Non-pairwise Compatibility Graphs. In *Proceedings of the International Computing and Combinatorics Conference, Atlanta, GA, USA, 29–31 August 2020*; Springer: New York, NY, USA, 2020; pp. 372–383.
3. Gugisch, R.; Kerber, A.; Kohnert, A.; Laue, R.; Meringer, M.; Rücker, C.; Wassermann, A. MOLGEN 5.0, a molecular structure generator. In *Advances in Mathematical Chemistry and Applications*; Elsevier: Amsterdam, The Netherlands, 2015; pp. 113–138.

4. Peironcely, J.E.; Rojas-Chertó, M.; Fichera, D.; Reijmers, T.; Coulier, L.; Faulon, J.L.; Hankemeier, T. OMG: Open molecule generator. *J. Cheminf.* **2012**, *4*, 21.
5. Fujiwara, H.; Wang, J.; Zhao, L.; Nagamochi, H.; Akutsu, T. Enumerating treelike chemical graphs with given path frequency. *J. Chem. Inf. Model.* **2008**, *48*, 1345–1357.
6. Ishida, Y.; Zhao, L.; Nagamochi, H.; Akutsu, T. Improved algorithms for enumerating tree-like chemical graphs with given path frequency. *Genome Inf.* **2008**, *21*, 53–64.
7. Suzuki, M.; Nagamochi, H.; Akutsu, T. Efficient enumeration of monocyclic chemical graphs with given path frequencies. *J. Cheminf.* **2014**, *6*, 31.
8. Suzuki, M.; Nagamochi, H.; Akutsu, T. A 2-phase algorithm for enumerating tree-like chemical graphs satisfying given upper and lower bounds. *IPSI SIG Tech. Rep.* **2012**, *28*, 1–8.
9. Jin, W.; Barzilay, R.; Jaakkola, T. Junction tree variational autoencoder for molecular graph generation. *arXiv* **2018**, arXiv:1802.04364.
10. Nakano, S.I.; Uno, T. *A Simple Constant Time Enumeration Algorithm for Free Trees*; PSJ SIGNotes ALgorithms; PSJ: Tokyo, Japan, 2003.
11. Hayat, U.; Umer, M.; Gutman, I.; Davvaz, B.; de Celis, Á.N. A novel method to construct NSSD molecular graphs. *Open Math.* **2019**, *17*, 1526–1537.
12. Blum, L.C.; Raymond, J.L. 970 million druglike small molecules for virtual screening in the chemical universe database GDB-13. *J. Am. Chem. Soc.* **2009**, *131*, 8732–8733.
13. Azam, N.A.; Chiewvanichakorn, R.; Zhang, F.; Shurbevski, A.; Nagamochi, H.; Akutsu, T. A method for the inverse QSAR/QSPR based on artificial neural networks and mixed integer linear programming. In Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies—Volume 3: BIOINFORMATICS, Valletta, Malta, 24–26 February 2020.
14. Ito, R.; Azam, N.A.; Wang, C.; Shurbevski, A.; Nagamochi, H.; Akutsu, T. A Novel Method for the Inverse QSAR/QSPR to Monocyclic Chemical Compounds Based on Artificial Neural Networks and Integer Programming. In *Advances in Computer Vision and Computational Biology*; Research Book Series; Springer Nature: Berlin, Germany, 2020.
15. Zhu, J.; Wang, C.; Shurbevski, A.; Nagamochi, H.; Akutsu, T. A Novel Method for Inference of Chemical Compounds of Cycle Index Two with Desired Properties Based on Artificial Neural Networks and Integer Programming. *Algorithms* **2020**, *13*, 124.
16. Méndez-Lucio, O.; Baillif, B.; Clevert, D.A.; Rouquié, D.; Wichard, J. De novo generation of hit-like molecules from gene expression signatures using artificial intelligence. *Nat. Commun.* **2020**, *11*, 1–10.
17. Lim, J.; Hwang, S.Y.; Moon, S.; Kim, S.; Kim, W.Y. Scaffold-based molecular design with a graph generative model. *Chem. Sci.* **2020**, *11*, 1153–1164.
18. Meringer, M.; Schymanski, E.L. Small molecule identification with MOLGEN and mass spectrometry. *Metabolites* **2013**, *3*, 440–462.
19. Haruna, T.; Horiyama, T.; Shimokawa, K. *On the Enumeration of Polymer Topologies*; 2017-AI-162; IPSJ SIG Technical Report; Information Processing Society of Japan: Tokyo, Japan, 2017.
20. Tezuka, Y.; Oike, H. Topological polymer chemistry. *Prog. Polym. Sci.* **2002**, *27*, 1069–1122.
21. Azam, N.A.; Shurbevski, A.; Nagamochi, H. An Efficient Algorithm to Count Tree-Like Graphs with a Given Number of Vertices and Self-Loops. *Entropy* **2020**, *22*, 923.
22. Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C. *Introduction to Algorithms*; MIT Press: Cambridge, MA, USA, 2009.

23. Masui, R.; Shurbevski, A.; Nagamochi, H. *Enumeration of Unlabeled Tree by Dynamic Programming*; Technical Report No. 2019-003; Department of Applied Mathematics and Physics, Kyoto University: Kyoto, Japan, 2009. Available online: <http://www.amp.i.kyoto-u.ac.jp/tecrep/ps-file/2019/2019-003.pdf> (Accessed on October 2020).
24. Jordan, C. Sur les assemblages de lignes. *J. Reine Angew. Math* **1869**, 70, 81.

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).