IPSJ SIG Technical Report

Vol.2010-AL-129 No.5
2010/3/5

Enumerating Colored and Rooted Outerplanar Graphs

Jiexun Wang[†1] and Hiroshi Nagamochi[†1]

An outerplanar graph is a graph that admits a planar embedding such that all vertices appear on the boundary of its outer face. Given a positive integer n and a color set $\mathcal{C}$ with $K \geq 1$ colors, we consider the problem of enumerating all colored and rooted outerplanar graphs with at most n vertices without repetition. We design an efficient algorithm that can generate all required graphs in $O(1)$ time per each and in $O(n)$ space.

1. Introduction

The problem of enumerating (or listing) graphs of specific classes without repetition is one of fundamental and important problems in computer science. Various types of graphs has been studied such as trees[7)–10)] and series-parallel graphs[5)]. The graphical enumeration has been served as a very useful tool for solving problems in various field such as chemistry[2),6)] and biology[3)].

The goal of graphical enumeration problems is to efficiently enumerate graphs in classes without duplications in terms of running time and space. The running time (resp., space) of an enumeration algorithm is the time (resp., computer memory) required to compute the total amount of changes in the data structures, but not the time (resp., memory) required to print out all graphs. Particularly, many researchers are interested in the computation time per each graph, i.e., the *delay* between two successive outputs[7),9)]. Note that graphical symmetry leads to isomorphic duplications which prevent from designing efficient enumeration algorithms.

In literature, there are various approaches developed for graphical enumeration problems[1),4),5)]. The common idea behind most of efficient algorithms is to first define a unique embedding for each graph to be enumerated as its canonical representation, and then define a parent-child relationship among all canonical representations such that the difference between parent-child embeddings is small, which is implicitly represented as the *family tree* each of whose nodes corresponds to the canonical representation of a graph. This indicates that all canonical representations can be enumerated one by one according to the depth-first traversal of the family tree.

In this paper, we consider the problem of enumerating all colored and rooted outerplanar graphs with at most given n (≥ 1) vertices without repetition. To our best knowledge, few papers have been worked for this problem. The reasons are twofolds: 1) few researchers notice potential applications of the exhaustive list of outerplanar graphs, and 2) it is difficult to design efficient algorithms because free symmetry at cut-vertices and reflectional symmetry at rooted blocks lead to isomorphic duplications. We design an efficient algorithm that can enumerate all required outerplanar graphs in $O(1)$ per each and in $O(n)$ space. The algorithm proposed in this paper may have potential applications in various areas such as chemoinformatics, medicine and computer science.

2. Preliminaries

Throughout this paper, a graph stands for a simple connected undirected graph. The set of vertices and the set of edges of a graph H are denoted by $V(H)$ and $E(H)$, respectively.

A graph is called *planar* if its vertices and edges can be drawn as points and curves on the plane so that no two curves intersect except for their endpoints. An *outerplanar* graph is a planar graph that admits a plane graph such that all vertices appear on the boundary of its outer face.

A *rooted* graph is a graph with an arbitrary vertex designated as the root. For each block B of a graph rooted at a vertex r, the *root* $r(B)$ of B is defined to be the unique vertex $v \in V(B)$ closest to r. Let $V'(B)$ denote $V(B) - \{r(B)\}$. A block B is called the *parent-block* of all other vertices in $V'(B)$. A block B with $r(B) = v$ is a *child-block* of v. The *depth* $d(B)$ of a rooted block B is defined by the number of blocks which edge sets intersect with a simple path from a vertex in $V'(B)$ to the root r. Let B_r be an imaginary block which is the parent-block of r_G, and define depth $d(B_r) = 0$. For two blocks B and B' with $r(B') \in V'(B)$, we

†1 Department of Applied Mathematics and Physics, Kyoto University

Vol.2010-AL-129 No.5
2010/3/5

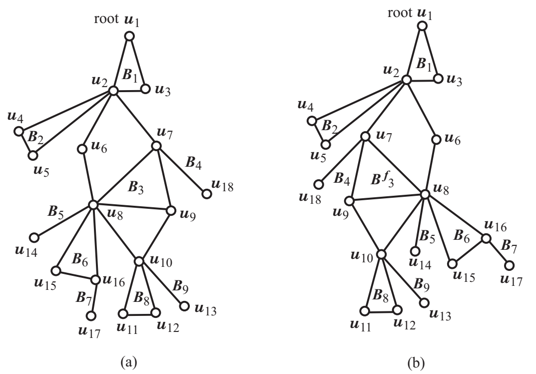

Fig. 1 (a) An embedding G of a rooted outerplanar graph; (b) The embedding G' obtained by flipping block B_3 of G.

say that B is the *parent-block* of B' and that B' is a *child-block* of B. Similarly, we define the *ancestor-blocks* and *descendant-blocks*.

Let $\mathcal{C}$ be a set of colors. A *colored* graph is a graph in which each vertex v is assigned with a color $c(v) \in \mathcal{C}$ (different vertices can receive the same color). Two colored and rooted graphs H_1 and H_2 are *rooted-isomorphic* if and only if their vertex sets admit a bijection by which the root, the color classes, and the incidence-relation between vertices and edges in H_1 correspond to those in H_2. Note that two distinct embeddings of a colored and rooted graph are rooted-isomorphic.

A colored and rooted outerplanar graph H can have several different embeddings in the plane, where each embedding is determined by choosing one of the two ways of embeddings of each block and choosing one of the orderings of child-blocks of each cut-vertex. We let $G(B)$ denote the embedding of that consists of embeddings of B and all descendant-blocks of B. For an embedding G, let G^f denote the *flipped embedding* of G that is obtained by reversing the embedding G on the plane. For example, Fig. 1(a) shows an embedding G of a rooted outerplanar graph, and Fig. 1(b) shows the embedding G' obtained by flipping block B_3 in G.

3. Rooted Outerplanar Graphs

Let G be a colored and rooted outerplanar embedding, and let r_G denote the root of G. We define the depth $d(r_G) = 0$ for the root r_G, and depth of other vertices in G recursively based on the following decomposition of blocks.

3.1 Structure of rooted blocks

We decompose a rooted block B into three parts: "core," "left wings" and "right wings," where the core is a subgraph which is reflectionally symmetric in the block B (except for an assignment of colors to the vertices in the core), left (resp., right) wings can be treated as rooted outerplanar embeddings on the left (resp., right) side of B.

Specifically, for a block B in G, the vertices in $V'(B)$ adjacent to $r(B)$ are called the *head-vertices* of B, and the edges in B incident to $r(B)$ are called the *head-edges* of B. Let $V_{\mathtt{head}}(B)$ denote the set of all head-vertices in B, and let $h = |V_{\mathtt{head}}(B)|$. We denote the head-vertices in $V_{\mathtt{head}}(B)$ by

$$x_1, x_2, \ldots, x_{(h-1)/2}, z, y_{(h-1)/2}, \ldots, y_2, y_1 \quad \text{(if } h \text{ is odd)}$$
$$x_1, x_2, \ldots, x_{h/2}, y_{h/2}, y_{h/2-1}, \ldots, y_2, y_1 \quad \text{(if } h \text{ is even)}$$

from left to right, where x_1 is the leftmost vertex in $V(B)$ adjacent to $r(B)$. See Fig. 2. Define depth of the above head-vertices to be

$$d(x_i) = d(y_i) = d(r(B)) + i, \quad d(z) = d(r(B)) + (h+1)/2.$$

We define "axial-faces," "axial-vertex" and "bottom" of block B as follows. Let h be odd. We call vertex z the *bottom vertex* of B and denote it by $bv(B)$. If $h = 1$, then no axial-face is defined for B. If $h \geq 3$, then an inner face of B containing edge $(r(B), z)$ is called an *axial-face* of B. Note that B has exactly two such faces.

Let h be even. The inner face f_1 of B containing both edges $(r(B), x_{h/2})$ and $(r(B), y_{h/2})$ is called the *first axial-face* of B. If f_1 consists of an odd number of edges, then f_1 has a unique edge e^1 farthest from $r(B)$, and the other inner face containing e^1 (if any) is defined to be the *second axial-face* f_2. For each axial-face f_i, $i \geq 2$, if f_i consists of an even number of edges, then f_i has a unique edge e^i farthest from $r(B)$, and the other inner face containing e^i (if any) is defined to be the $(i+1)$*st axial-face* f_{i+1}.

Given any block B, the non-head-vertices in all axial-faces are called the *axial-*

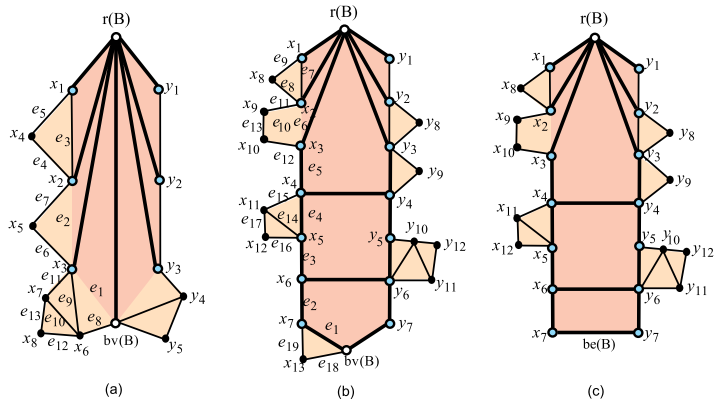

Fig. 2 Structure of a rooted block.

vertices, and the non-head-edges in all axial-faces are called the *axial-edges*. Let $V_{\mathtt{axis}}(B)$ denote the set of axial-vertices in B. Note that B has no axial-vertex if $|V_{\mathtt{head}}(B)|$ is odd. The depth $d(u)$ of an axial-vertex u is defined to be the number of edges in a shortest path from u to a head-vertex v plus $d(v)$. The last axial-face f_p has a unique vertex or edge farthest from $r(B)$, which is called the *bottom vertex* of B or *bottom edge* of B, and denote it by $bv(B)$ or $be(B)$, respectively. We let $bv(B)=\emptyset$ (resp., $be(B)=\emptyset$) mean that B has no bottom vertex (resp., edge). A head- or axial-vertex (if any) is called a *core-vertex* of B. Let $V_{\mathtt{core}}(B)$ denote the set of core-vertices of B.

A non-core-vertex in B is called a *wing-vertex* of B. Let $V_{\mathtt{wing}}(B)$ denote the set of wing-vertices in B. We define a unique numbering and depths for all wing-vertices as follows. Here we only explain the case where $|V_{\mathtt{head}}(B)|=h$ is even, because the case where h is odd is similar. Let

$$x_1, x_2, \ldots, x_p \text{ (resp., } x_1, x_2, \ldots, x_p, bv(B))$$

be the sequence of axial-vertices on the shortest path from x_1 to the bottom if $(x_p, y_p)=be(B)$ (resp., $bv(B)$ exists). We define $y_1, y_2, \ldots, y_p$ (resp., $y_1, y_1, \ldots, y_p, bv(B)$) symmetrically.

Let x and x' be two consecutive core-vertices in the sequence $x_1, x_2, \ldots, x_p, bv(B)$ (possibly $bv(B)=\emptyset$). Removal of these vertices from B leaves at most one subgraph B' which consists of wing-vertices. Let $B(x,x')$ denote such a subgraph B' if any. We define a unique numbering for the wing-vertices in $B(x,x')$ as the reverse order of the following vertex eliminations: recursively eliminating the wing-vertex of degree 2 visited *last* when traversing the boundary of $B(x,x')$ from x to x' until no wing-vertices in $B(x,x')$ remains (Fig. 2). Besides, we define a unique numbering π for all wing-vertices in $B(x_i, x_{i+1})$ for $i=1,2,\ldots,p-1$ (and in $B(x_p, bv(B))$) if $bv(B)$ exists). The depth of the jth wing-vertex w in π is defined to be $d(w)=d(r(B))+p+j$ (see Fig. 2).

The sequence of core-vertices $x_1, x_2, \ldots, x_p$ and wing-vertices in the order π is called the *left side* of B. We define the right side of B in the same way (note that $bv(B)$ is not contained in the left or right side of B). Let $V^{\mathtt{L}}(B)$ and $V^{\mathtt{R}}(B)$ denote the sets of vertices in the left and right sides of B, respectively. A vertex $u \in V^{\mathtt{L}}(B)$ (resp., $V^{\mathtt{R}}(B)$) is called a *left* (*resp., right*) *vertex* of B. Also denote $V^{\mathtt{L}}(B) \cap V_{\mathtt{core}}(B)$ by $V^{\mathtt{L}}_{\mathtt{core}}(B)$. Similarly for $V^{\mathtt{R}}_{\mathtt{core}}(B)$, $V^{\mathtt{L}}_{\mathtt{head}}(B)$ and $V^{\mathtt{R}}_{\mathtt{head}}(B)$, $V^{\mathtt{L}}_{\mathtt{axis}}(B)$, $V^{\mathtt{R}}_{\mathtt{axis}}(B)$, $V^{\mathtt{L}}_{\mathtt{wing}}(B)$ and $V^{\mathtt{R}}_{\mathtt{wing}}(B)$. For a vertex u in the left side of B, let $P_{\mathtt{L}}(u;B)$ denote the boundary of the left side of B from u to the bottom of B (excluding the bottom edge), and $E_{\mathtt{L}}(u;B)$ denote the sequence of edges in the path $P_{\mathtt{L}}(u;B)$. We define $P_{\mathtt{R}}(u;B)$ and $E_{\mathtt{R}}(u;B)$ for the right side symmetrically with $P_{\mathtt{L}}(u;B)$ and $E_{\mathtt{L}}(u;B)$.

Let $\tilde{E}(B)$ denote the set of all edges (v,v') with $v, v \in V^{\mathtt{L}}(B) \cup \{bv(B)\}$ or $v, v \in V^{\mathtt{R}}(B) \cup \{bv(B)\}$, where we include edge (v,v') that appears as an edge when we remove the wing-vertex w adjacent to v and v' to define the ordering π, but (v,v') is not an edge in B. A left (resp., right) edge $e=(v,v')$ is an edge such that $\{v,v'\} \subseteq V^{\mathtt{L}}(B) \cup \{bv(B)\}$ (resp., $\{v,v'\} \subseteq V^{\mathtt{R}}(B) \cup \{bv(B)\}$).

We define depth $d(e)$ for all left edges $e \in \tilde{E}(B)$ as follows. Let $L_1 = |V^{\mathtt{L}}_{\mathtt{core}}(B) \cup \{bv(B)\}|$ (possible $bv(B)=\emptyset$), and $L_2 = |V^{\mathtt{L}}_{\mathtt{wing}}(B)|$. For the left wing-vertex w with the largest depth and the two edges e and e' incident to w, where e is closer to x_1 than e' along $P_{\mathtt{L}}(x_1;B)$, we let $d(e)=2L_2+L_1-1$ and $d(e')=2L_2+L_1-2$, and then remove w. We repeat this procedure of assigning pair of numbers $(2(L_2-1)+L_1-1, 2(L_2-1)+L_1-2)$, ..., (L_1+1, L_1) until no left wing-vertices remain. After removing all left wing-vertices, we assign $d(e_i)=i$ for the ith edge e_i along $P_{\mathtt{L}}(x_1;B')$ when we traverse $P_{\mathtt{L}}(x_1;B')$ reversely from the bottom to the first left head-vertex x_1 in the resulting block (see Fig. 2). We define depth $d(e)$ for all right edges $e \in \tilde{E}(B)$ similarly.

3.2 Tips of Rooted Blocks

We define the "tip" $t(B)$ of a block B as follows. If B consists of a single vertex u, then we define $t(B)$ to be the vertex u. Otherwise, let $\{x_i \mid i = 1, 2, \ldots, p_{\mathtt{L}}\}$ with $p_{\mathtt{L}} = |V^{\mathtt{L}}(B)|$ (resp., $\{y_j \mid j = 1, 2, \ldots, p_{\mathtt{R}}\}$ with $p_{\mathtt{R}} = |V^{\mathtt{R}}(B)|$) denote the set of vertices in the left (resp., right) side of B, where $d(x_i) = d(r(B)) + i$ and $d(y_j) = d(r(B)) + j$.

Case-1. $V^{\mathtt{R}}_{\mathtt{cut}}(B) \neq \emptyset$ (see Fig. 3(a)): Define $t(B)$ to be the right vertex $y \in V^{\mathtt{R}}_{\mathtt{cut}}(B)$ with the *largest* depth $d(y)$.

Case-2. $V^{\mathtt{R}}_{\mathtt{cut}}(B) = \emptyset$ and $V^{\mathtt{R}}_{\mathtt{wing}}(B) \neq \emptyset$ (see Fig. 3(b)): Define $t(B)$ to be the right wing-vertex $y \in V^{\mathtt{R}}_{\mathtt{wing}}(B)$ with the *largest* depth $d(y)$.

Case-3. $V^{\mathtt{R}}_{\mathtt{cut}}(B) = V^{\mathtt{R}}_{\mathtt{wing}}(B) = \emptyset$ and $V^{\mathtt{L}}_{\mathtt{cut}}(B) \neq \emptyset$, where possibly $V^{\mathtt{L}}_{\mathtt{wing}}(B) = \emptyset$ (see Fig. 3(c)-(d)): Define $t(B)$ to be the left vertex $x \in V^{\mathtt{L}}_{\mathtt{cut}}(B)$ with the *largest* depth $d(x)$.

Case-4. $V^{\mathtt{R}}_{\mathtt{cut}}(B) = V^{\mathtt{R}}_{\mathtt{wing}}(B) = V^{\mathtt{L}}_{\mathtt{cut}}(B) = \emptyset$ and $V^{\mathtt{L}}_{\mathtt{wing}}(B) \neq \emptyset$ (see Fig. 3(e)): Define $t(B)$ to be the left wing-vertex $x \in V^{\mathtt{L}}_{\mathtt{wing}}(B)$ with the *largest* depth $d(x)$.

Case-5. $|V(B)| = 2$ or $V^{\mathtt{R}}_{\mathtt{cut}}(B) = V^{\mathtt{R}}_{\mathtt{wing}}(B) = V^{\mathtt{L}}_{\mathtt{cut}}(B) = V^{\mathtt{L}}_{\mathtt{wing}}(B) = \emptyset$, where possibly $\mathcal{B}(bv(B)) \neq \emptyset$ (see Fig. 3(f)-(g)): Define $t(B)$ to be the core-vertex $u \in V'(B)$ with the *largest* depth $d(u)$. Let $t(B)$ be the right endvertex of $be(B)$ if any.

The *spine* of G is defined to be the sequence of all successors starting from the rightmost block $B^1 \in \mathcal{B}(r_G)$ by $B^1, B^2, \ldots, B^p$, where B^1 is the rightmost block in $\mathcal{B}(r_G)$, and each B^i ($i \geq 2$) is the rightmost block in $\mathcal{B}(t(B^{i-1}))$ (Fig. 4). The tip $t(G)$ of G is defined to be the tip $t(B^p)$ of block B^p, and the last block B^p is called the *tip-block* of G.

4. Signatures of Embeddings

This section will explain how to encode each colored and rooted outerplanar embedding into a sequence, called *signature*, such that each embedding can be uniquely reconstructed from its signature. We will first define a parent-child relationship between two outerplanar embeddings, and then introduce the signature for an outerplanar embedding based on its parent-embedding.

Specifically, let G be an embedding with at least one vertex, and $t(G) = u$ be the tip of G. We define the *parent-embedding* $G' = P(G)$ by removing the vertex u. Accordingly, G is called a *child-embedding* of G', which can be obtained from G' by adding the vertex u. We first encode the operation that creates the vertex u into a sequence, denoted by $\gamma(u)$. Then we define the signature $\sigma(G)$ by the following recursive formula

$$\sigma(G) = [\sigma(G'), \gamma(u)].$$

We set $\sigma(G') = \emptyset$ if G' is an empty graph.

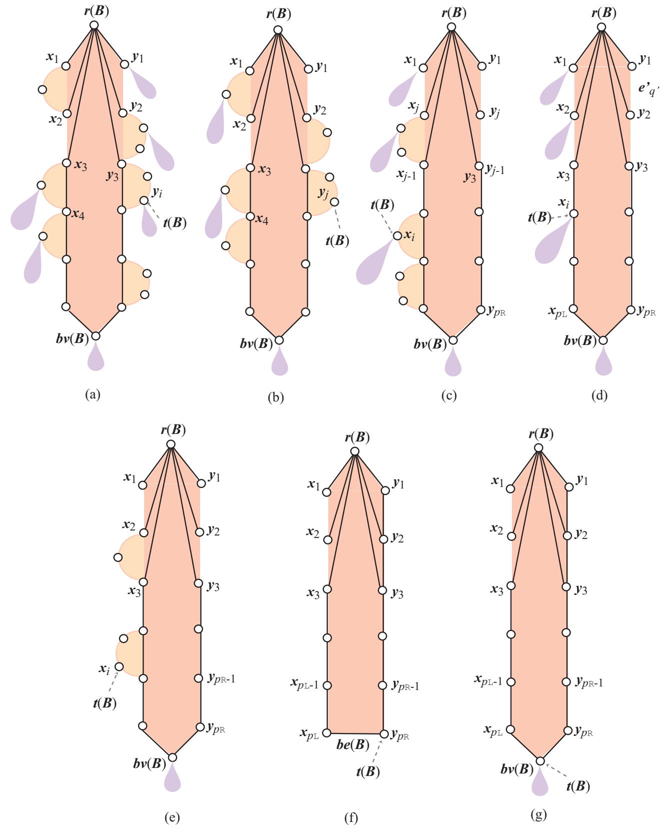

Fig. 3 Tip $t(B)$ of a rooted block B: (a) Case-1; (b) Case-2; (c) Case-3; (d) Case-3; (e) Case-4; (f) Case-5; and (d) Case-5.

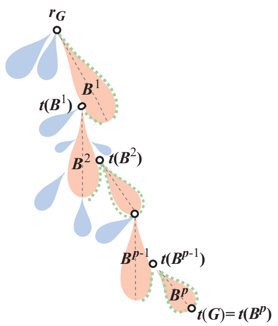

Fig. 4 An illustration for a sequence of blocks between r_G and $t(G)$, which forms a spine.

Now we explain how to define the code $\gamma(u)$ of the tip u of G. If $u = t_G$, i.e., G consists of a single vertex r_G, then we define the vertex code $\gamma(u) = c(u) \in \mathcal{C}$. Otherwise, we define a vertex code $\gamma(u)$ is a sequence

$$(d_1(u), \mathtt{at}(u), d_2(u), \mathtt{op}(u), c(u))$$

of five entries such that $d_1(u)$ and $d_2(u)$ are nonnegative integers, $c(u) \in \mathcal{C}$, $\mathtt{at}(u) \in \{\mathtt{h}^{\mathtt{L}}, \mathtt{w}^{\mathtt{L}}, \mathtt{h}^{\mathtt{R}}, \mathtt{w}^{\mathtt{R}}, *\}$, and $\mathtt{op}(u) \in \{\mathtt{new\text{-}block}, \mathtt{star}, \mathtt{triangle}, \mathtt{subdivide}\}$. Formally, we define the code $\gamma(u)$ of the tip u of G as follows. Let B be the parent-block of u, which is the tip-block of G.

(P-1) Let u be a head-vertex of B: Let $h = |V'(B)|$.

If $h = 1$, i.e., B consists a leaf edge $(v = r(B), u)$ of G, then for the block B' with $v \in V'(B')$, define

$$\gamma(u) = \begin{cases} (d(B'), \mathtt{h}^{\mathtt{L}}, d(v), \mathtt{new\text{-}block}, c(u)) & \text{if } v \text{ is a left vertex of } B' \\ (d(B'), \mathtt{h}^{\mathtt{R}}, d(v), \mathtt{new\text{-}block}, c(u)) & \text{if } v \text{ is a right vertex of } B' \\ (d(B'), *, d(v), \mathtt{new\text{-}block}, c(u)) & \text{otherwise,} \end{cases} \quad (1)$$

where $B' = B_r$ with $d(B_r) = 0$ if $v = r_G$.

If $h = 2$, i.e., B consists of a triangle $(r(B), \ell v(B), u)$ of G and $(r(B), \ell v(B))$ is an edge in B, then define

$$\gamma(u) = (d(B), *, d(\ell v(B)), \mathtt{triangle}, c(u)).$$

For $h \geq 3$, let v and v' be the vertices in $V'(B)$ adjacent to u, and let $d(v') \geq d(v)$, where (v, v') is an edge e in $P(G)$. Define

$$\gamma(u) = (d(B), *, d(v'), \mathtt{star}, c(u)).$$

(P-2) Let u be an axial-vertex of B, where u is of degree 2 in G: Let v and v' be the vertices in $V'(B)$ adjacent to u, and let $d(v') \geq d(v)$, where (v, v') is an edge e in $P(G)$, but (v, v') can be an edge in B only when $|V(B)|$ is even. Define

$$\gamma(u) = \begin{cases} (d(B), *, d(v'), \mathtt{triangle}, c(u)) & \text{if } u, v \text{ and } v' \text{ form a triangle in } G \\ (d(B), *, d(v'), \mathtt{subdivide}, c(u)) & \text{if } v \text{ and } v' \text{ are not adjacent in } G. \end{cases}$$

(P-3) Let u be a left wing-vertex of B: Let x and x' be the two vertices in B adjacent to u, where (x, x') is an edge e in $P(G)$ and $e \in \tilde{E}(B)$ holds. Define

$$\gamma(u) = \begin{cases} (d(B), \mathtt{w}^{\mathtt{L}}, d(e), \mathtt{triangle}, c(u)) & \text{if } u, x \text{ and } x' \text{ form a triangle in } G \\ (d(B), \mathtt{w}^{\mathtt{L}}, d(e), \mathtt{subdivide}, c(u)) & \text{if } x \text{ and } x' \text{ are not adjacent in } G. \end{cases}$$

(P-4) Let u be a right wing-vertex of B: Let y and y' be the two vertices in B adjacent to u, where (y, y') is an edge e in $P(G)$ and $e \in \tilde{E}(B)$ holds. Define

$$\gamma(u) = \begin{cases} (d(B), \mathtt{w}^{\mathtt{R}}, d(e), \mathtt{triangle}, c(u)) & \text{if } u, y \text{ and } y' \text{ form a triangle in } G \\ (d(B), \mathtt{w}^{\mathtt{R}}, d(e), \mathtt{subdivide}, c(u)) & \text{if } y \text{ and } y' \text{ are not adjacent in } G. \end{cases}$$

By definition, we can see that each vertex code uniquely determines the resulting graph augmented with a new vertex.

Let an *element* denote a vertex or an edge. In case (P-1) with $h = 1$, we say that G is obtained from $P(G)$ by creating a new block at B with an application of $\gamma(u)$ to vertex $v = r(B)$ in $P(G)$. In case (P-1) with $h \geq 2$ and cases (P-2)-(P-4), we say that G is obtained from $P(G)$ by expanding block B with an application of $\gamma(u)$ to edge e in $P(G)$. Such a vertex v and an edge e in $P(G)$ are called *applicable* elements in $P(G)$.

5. Canonical Embeddings

We choose a specific embedding of a colored and rooted outerplanar graphs as *canonical* such that it facilitates identifying free symmetry at cut-vertices and reflectional symmetry along rooted blocks.

We will explain the choice of canonical embedding after giving necessary notations. For two sequences A and B, let $A > B$ mean that A is lexicographically larger then B, and let $A \geq B$ mean that $A > B$ or $A = B$. Let $A \sqsupset B$ mean that B is a prefix of A and $A \neq B$, and let $A \gg B$ mean that $A > B$ but B is not a prefix of A. Let $A \sqsupseteq B$ mean that $A \sqsupset B$ or $A = B$, i.e., B is a prefix of A.

For two embeddings G_1 and G_2 of a graph H, we compare two signatures $\sigma(G_1)$ and $\sigma(G_2)$ by comparing their codes lexicographically code-wise. We compare two vertex codes γ and γ' by comparing their entries lexicographically, treating

Vol.2010-AL-129 No.5
2010/3/5

colors and labels as negative integers such that

$$0 > c_K > c_{K-1} > \cdots > c_1 > * > \mathtt{w}^{\mathtt{L}} > \mathtt{h}^{\mathtt{L}} > \mathtt{w}^{\mathtt{R}} > \mathtt{h}^{\mathtt{R}}$$
$$> \mathtt{subdivide} > \mathtt{triangle} > \mathtt{star} > \mathtt{new\text{-}block}.$$

For each block $B \in \mathcal{B}(v)$, the signature $\sigma(G)$ of an embedding G contains a subsequence which consists of the codes of vertices in $V(G(B)) - \{v\}$, which we denote by $\sigma(G(B);G)$.

Left-sibling-heaviness An embedding G is called *left-sibling-heavy* at a block $B \in \mathcal{B}(v) = (B_1, B_2, \ldots, B_p)$ if $B = B_1$ or $\sigma(G) \geq \sigma(G')$ holds for the embedding G' obtained from G by exchanging the order of B_{i-1} and $B_i = B$ in $\mathcal{B}(v)$.

We can obtain the following result.

Lemma 1 An embedding G is left-sibling-heavy at a block $B_i \in \mathcal{B}(v) = (B_1, B_2, \ldots, B_p)$ with $i \geq 2$ if and only if $\sigma(G(B_{i-1});G) \geq \sigma(G(B_i);G)$ holds. ∎

Let $\hat{B} \in \mathcal{B}(r(B))$ denote the sibling preceding B, where we let $\hat{B} = \emptyset$ indicate that there is no such sibling (i.e., B is the leftmost block in $\mathcal{B}(r(B))$). We define the *sibling-state* $\mathtt{sbl}(B;G)$ of a block B in G as follows:

$$\mathtt{sbl}(B;G) = \begin{cases} \mathtt{stc} & \text{if } \hat{B} = \emptyset \text{ or } \sigma(G(\hat{B});G) \gg \sigma(G(B);G) \\ \mathtt{pfx} & \text{if } \hat{B} \neq \emptyset \text{ and } \sigma(G(\hat{B});G) \sqsupset \sigma(G(B);G) \\ \mathtt{eqv} & \text{if } \hat{B} \neq \emptyset \text{ and } \sigma(G(\hat{B});G) = \sigma(G(B);G). \end{cases} \quad (2)$$

Left-side-heaviness An embedding G is called *left-side-heavy* at a block $B \in \mathcal{B}(v)$ if $\sigma(G) \geq \sigma(G')$ holds for the embedding G' obtained from G by replacing B with B^f (thus flipping the embedding B along the axis through v and the bottom of B).

The code subsequence $\sigma(G(B);G)$ consists of six subsequences: the first consists of the codes of left or right core-vertices excluding $bv(B)$ (denoted by $\sigma_{\mathtt{core}}(G(B);G)$), the second consists of the code of the descendants of the bottom vertex $bv(B)$ if any (denoted by $\sigma_{\mathtt{b}}(G(B);G)$), the third consists of the code of left wing-vertices (denoted by $\sigma^{\mathtt{L}}_{\mathtt{wing}}(G(B);G)$), the fourth consists of the code of descendants of left vertices (denoted by $\sigma^{\mathtt{L}}_{\mathtt{dscd}}(G(B);G)$), the fifth consists of the code of right wing-vertices (denoted by $\sigma^{\mathtt{R}}_{\mathtt{wing}}(G(B);G)$), and the sixth consists of the code of descendants of right vertices (denoted by $\sigma^{\mathtt{R}}_{\mathtt{dscd}}(G(B);G)$).

Besides, let $\sigma_{\mathtt{L}}(G(B);G)$ (resp., $\sigma_{\mathtt{R}}(G(B);G)$) denoted the subsequence of $\sigma(G(B);G)$ consisting of $\sigma^{\mathtt{L}}_{\mathtt{wing}}(G(B);G)$ and $\sigma^{\mathtt{L}}_{\mathtt{dscd}}(G(B);G)$ (resp., $\sigma^{\mathtt{R}}_{\mathtt{wing}}(G(B);G)$ and $\sigma^{\mathtt{R}}_{\mathtt{dscd}}(G(B);G)$). Let $\sigma^{\mathtt{L}}_{\mathtt{core}}(G(B);G)$ (resp., $\sigma^{\mathtt{R}}_{\mathtt{core}}(G(B);G)$) denote the sequence obtained from $\sigma_{\mathtt{core}}(G(B);G)$ by eliminating the codes of right (resp., left) core-vertices and of the bottom vertex $bv(B)$ (if any) after deleting the first four entries of each code in $\sigma_{\mathtt{core}}(G(B);G)$, respectively. Thus, $\sigma^{\mathtt{L}}_{\mathtt{core}}(G(B);G)$ (resp., $\sigma^{\mathtt{R}}_{\mathtt{core}}(G(B);G)$) is the sequence of color entries of left (resp., right) core-vertices of B.

For each left wing-vertex u of B (resp., a child-vertex $u \in Ch(v)$ of a vertex v in the left side of B), we define the *flipped code* $\overline{\gamma}(u)$ of vertex code $\gamma(u)$ to be the code obtained from $\gamma(u)$ by replacing the second entry $\mathtt{w}^{\mathtt{L}}$ (resp., $\mathtt{h}^{\mathtt{L}}$) with $\mathtt{w}^{\mathtt{R}}$ (resp., $\mathtt{h}^{\mathtt{R}}$). Symmetrically, for each right wing-vertex u of B (resp., a child-vertex $u \in Ch(v)$ of a vertex v in the right side of B), we define the *flipped code* $\overline{\gamma}(u)$ of vertex code $\gamma(u)$ to be the code obtained from $\gamma(u)$ by replacing the second entry $\mathtt{w}^{\mathtt{R}}$ (resp., $\mathtt{h}^{\mathtt{R}}$) with $\mathtt{w}^{\mathtt{L}}$ (resp., $\mathtt{h}^{\mathtt{L}}$). For a notational convenience, we set $\overline{\gamma}(u) = \gamma(u)$ for the other vertices $u \in V(G(B)) - \{r(B)\}$.

Let $\overline{\sigma_{\mathtt{L}}}(G(B);G)$ (resp., $\overline{\sigma_{\mathtt{R}}}(G(B);G)$) denote the sequence obtained from $\sigma_{\mathtt{L}}(G(B);G)$ (resp., $\sigma_{\mathtt{R}}(G(B);G)$) by replacing each vertex code $\gamma(u)$ with $\overline{\gamma}(u)$.

We can gain the following sufficient and necessary condition for left-side-heaviness at a rooted block.

Lemma 2 An embedding G is left-side-heavy at a block $B \in \mathcal{B}(v)$ if and only if it holds $[\sigma^{\mathtt{L}}_{\mathtt{core}}(G(B);G), \sigma_{\mathtt{L}}(G(B);G)] \geq [\overline{\sigma^{\mathtt{R}}_{\mathtt{core}}}(G(B);G), \overline{\sigma_{\mathtt{R}}}(G(B);G)]$. ∎

For simplicity, denote $\sigma^{\mathtt{L}}_{\mathtt{core}}(G(B);G)$ by $\sigma^{\mathtt{L}}_{\mathtt{core}}$. Similarly for $\overline{\sigma^{\mathtt{R}}_{\mathtt{core}}}$, $\sigma_{\mathtt{R}}$ and $\sigma_{\mathtt{L}}$. We define the *side-state* $\mathtt{sd}(B;G)$ of a block B in G as follows:

$$\mathtt{sd}(B;G) = \begin{cases} \mathtt{stc} & \text{if ``}[\sigma^{\mathtt{L}}_{\mathtt{core}}, \sigma_{\mathtt{L}}] \gg [\overline{\sigma^{\mathtt{R}}_{\mathtt{core}}}, \overline{\sigma_{\mathtt{R}}}] \\ \mathtt{nil} & \text{if } \sigma^{\mathtt{L}}_{\mathtt{core}} = \overline{\sigma^{\mathtt{R}}_{\mathtt{core}}} \text{ and } \sigma_{\mathtt{R}} = \emptyset \\ \mathtt{pfx} & \text{if } \sigma^{\mathtt{L}}_{\mathtt{core}} = \overline{\sigma^{\mathtt{R}}_{\mathtt{core}}} \text{ and } \sigma_{\mathtt{L}} \sqsupset \overline{\sigma_{\mathtt{R}}} \neq \emptyset \\ \mathtt{eqv} & \text{if } \sigma^{\mathtt{L}}_{\mathtt{core}} = \overline{\sigma^{\mathtt{R}}_{\mathtt{core}}} \text{ and } \sigma_{\mathtt{L}} = \overline{\sigma_{\mathtt{R}}} \neq \emptyset. \end{cases} \quad (3)$$

An embedding G is called *canonical* if it is left-sibling-heavy and left-side-heavy at all blocks in G. We find the following property of canonical embeddings.

Lemma 3 Let G be an embedding of a colored and rooted outerplanar graph H. Then G is canonical if and only if $\sigma(G)$ is lexicographically maximum among all $\sigma(G')$ of embeddings $G' \in \xi(H)$. ∎

Lemma 4 For a canonical embedding G with $|V(G)| \geq 2$, its parent-embedding $P(G)$ is a canonical embedding. ∎

Family Tree Based on Lemma 4, we construct a rooted tree whose nodes represent canonical embeddings to be generated, called *family tree*, where the root-node is an empty graph, and other nodes are canonical embeddings of all colored and rooted outerplanar graphs. This family tree indicates that all colored and rooted outerplanar graphs may be generated by traversing all nodes in some order such as depth-first searching.

6. Generating Canonical Child-embeddings

This section explains how to systematically generate all canonical child-embddings from any given canonical embedding G without repetition and without testing duplication. Based on Sections 4 and 5, an embedding G' is a canonical child-embedding of G if and only if G' is obtained from G by attaching a new vertex v to an element ε (i.e., vertex or edge) in $V(G) \cup E(G)$ with a vertex code γ and G' satisfies the left-heavy properties. The systematical generation of all canonical child-embeddings of G depends on the determination of all possible elements ε of G (arranged as a sequence $\mathcal{E}^*(G)$) and all possible vertex codes (denoted by Γ). We can see that if all these valid elements and vertex codes can be automatically gained, then all canonical child-embeddings of G can be generated systematically without repetition.

To obtain $\mathcal{E}^*(G)$ and Γ, we first identify all the elements ε in $V'(B) \cup E(B)$ and vertex code set $\Gamma(\varepsilon)$ that admit a vertex code $\gamma \in \Gamma(\varepsilon)$ such that $G' = G + \gamma(u_{N+1})$ remains left-side-heavy at a block B based on the same five cases used to define the tip of a block. We give the set of such elements in $V'(B) \cup E(B)$ as a sequence of these elements, called the *element sequence* $\mathcal{E}(B)$. Let $\mathcal{E}(B) = \emptyset$ if $\mathtt{sd}(B;G) = \mathtt{eqv}$, since no application of $\gamma(u_{N+1})$ is applied any element in $V'(B) \cup E(B)$ without violating left-side-heaviness of B.

Then we identify the condition for $G + \gamma(u_{N+1})$ to remain canonical, i.e., left-sibling-heavy and left-side-heavy at all blocks. For the spine $B^1, B^2, \dots, B^p$ of G, define sequences

$$\mathcal{E}(G) = [r_G, \mathcal{E}(B^1), \mathcal{E}(B^2), \dots, \mathcal{E}(B^p)]$$

and

$$s(G) = [s_1 = \mathtt{sbl}(B^1;G), s_2 = \mathtt{sd}(B^1;G), s_3 = \mathtt{sbl}(B^2;G), s_4 = \mathtt{sd}(B^2;G), \dots, s_{2p-1} = \mathtt{sbl}(B^p;G), s_{2p} = \mathtt{sd}(B^p;G)].$$

A canonical child-embedding $G' = G + \gamma(u_{N+1})$ of G is generated by applying a code $\gamma(u_{N+1}) = (d_1, \mathtt{at}, d_2, \mathtt{op}, c) \in \Gamma(\varepsilon)$ to an element $\varepsilon \in \mathcal{E}(G)$.

Let B^h be the block in the spine with $\varepsilon \in V'(B^h) \cup E(B^h)$, i.e., the block to which a new vertex u_{N+1} is introduced, where B^h is determined by $d(B^h) = d_1$. Let B' denote the new block created by $\gamma(u_{N+1})$ (if ε is a vertex), where B' is the $(h+1)$st block in the spine of G'. Observe that G' is also left-sibling-heavy and left-side-heavy at any block not in the spine of G or at any block B^i with $i > h$ in the spine of G. Thus, to know when G' is canonical, we only need to examine states $s_1, s_2, \dots, s_{2h-1}, s_{2h}$ in G' and the new sibling-state $s_{2h+1} = \mathtt{sbl}(B';G')$ (if ε is a vertex) (recall that the side-state $s_{2h+2} = \mathtt{sbl}(B';G')$ is always $\mathtt{nil}$).

We define *copy-state* $cs(G)$ of G to be the state $s_{i^*} \in \{\mathtt{eqv}, \mathtt{pfx}\}$ with the minimum index i^* in $s(G)$, and the block B^ℓ attaining $s_{i^*} = \mathtt{sbl}(B^\ell;G)$ or $s_{i^*} = \mathtt{sd}(B^\ell;G)$ is called the *dominating block* of G; let $cs(G) = \mathtt{stc}$ and $i^* = \infty$ otherwise (i.e., each state in $s(G)$ is $\mathtt{stc}$ or $\mathtt{nil}$). Then we can characterize the element sequence $\mathcal{E}^*(G)$ by the copy-state $cs(G)$ of G. Specifically, if $cs(G) = \mathtt{stc}$, then $\mathcal{E}^*(G) = \mathcal{E}(G)$, if $cs(G) = \mathtt{eqv}$, then $\mathcal{E}^*(G)$ is the sequence of element obtained from $\mathcal{E}(G)$ by deleting the elements contained in a block B^i with $i \geq \ell$ in the spine, and if $cs(G) = \mathtt{pfx}$, then $\mathcal{E}^*(G)$ is the sequence of elements obtained from $\mathcal{E}(G)$ by deleting the elements preceding a unique and specific element ε^* (excluding ε^*), where ε^* can be computed in $O(1)$ time based on constant-size of the information about the dorminating block B^ℓ of G. The vertex code set $\Gamma = \cup_{\varepsilon \in \mathcal{E}^*(G)} \Gamma(\varepsilon)$.

7. Algorithm

Recall that all canonical embeddings with at most n vertices to be enumerated are arranged in the family tree. Starting from a graph consisting of a single vertex, we recursively enumerate its first canonical child-embedding by appending a new vertex to the current graph until reaching an embedding that has no child-embedding; and then backtracking to the most recent embedding which has child-embeddings not being enumerated yet. Note that during the enumeration, we only output the constant-size difference between two consecutive embeddings.

The above idea of enumeration is presented as the following Algorithm GENERATE, where "/*... */" indicates a comment.

Algorithm GENERATE($n, \mathcal{C}$)

Input: An integer $n \geq 1$ and a set $\mathcal{C} = (c_1, c_2, \ldots, c_K)$ of K colors.
Output: All canonical embeddings of colored and rooted outerplanar graphs with at most n vertices.

begin
  for each $c \in \mathcal{C}$ **do**
    Let G be the graph consisting of a single vertex $u_1 (= r_G)$ with $c(u_1) = c$;
    Let B_r be the imaginary parent-block of the root r_G;
    Output $c(u_1) = c$;
    $\varepsilon_1 := u_1$; $\gamma_1 := (0, *, 0, \texttt{new-block}, c_1)$;
    GEN$(G, B_r, \varepsilon_1, \gamma_1)$
  endfor
end.

Given an embedding G with N vertices, a block B, an element $\varepsilon \in \mathcal{E}(B)$ and a code $\gamma \in \Gamma(\varepsilon)$, Procedure GEN$(G, B, \varepsilon, \gamma)$ recursively generates all descendant-embeddings of G with at most n vertices, which is given as follows.

Procedure GEN$(G, B, \varepsilon, \gamma)$
/* Let $N = |V(G)| \in [1, n-1]$, and u_{N+1} be a new vertex that will be created. */
begin
  G' :=APPEND$(G, B, \varepsilon, \gamma)$; /* Compute a child-embedding G' of G. */
  if N is odd **then** Output $\gamma(u_{N+1}) = \gamma$ **endif**;
  if $N + 1 < n$ **then**
    $\varepsilon_1 := r_G$; $\gamma_1 := (0, *, 0, \texttt{new-block}, c_1)$; GEN$(G', B_r, \varepsilon_1, \gamma_1)$
  endif;
  if N is even **then** Output $\gamma(u_{N+1}) = \gamma$ **endif**;
  REMOVETIP(G'); /* Compute G from G' by removing the tip of G'. */
  $[B', \varepsilon', \gamma']$:=NEXTCODE$(B, \varepsilon, \gamma; G)$; /* Calculate three parameters B', ε' and γ' to generate the next child-embedding of G */
  if $[B', \varepsilon', \gamma'] \neq \emptyset$ **then** GEN$(G, B', \varepsilon', \gamma')$ **endif**
end.

Theorem 5 Given an integer $n \geq 1$ and a set $\mathcal{C} = (c_1, c_2, \ldots, c_K)$ of $K \geq 1$ colors, the proposed algorithm enumerates all colored and rooted outerplanar graphs with at most n vertices without repetition in $O(1)$ time per each and in $O(n)$ space. ∎

8. Concluding Remarks

This paper has proposed an efficient algorithm for generating all colored and rooted outerplanar graphs with at most given number n (≥ 1) vertices without repetition in $O(1)$ time per each and $O(n)$ space.

References

1) Colbourn C. J. and Read R. C.: Orderly Algorithms for Graph Generation, *International Journal of Computer Mathematics. Section A: Computer Systems: Theory*, vol. 7, 167-172, 1979.
2) Fujiwara H., Wang J., Zhao L., Nagamochi H., and Akutsu T.: Enumerating Tree-like Chemical Graphs with Given Path Frequency, *Journal of Chemical Information and Modeling*, vol. 48, pp. 1345-1357, 2008.
3) Georgii E., Dietmann S., Uno T., Pagel P., and Tsuda K.: Enumeration of Condition-dependent Dense Modules in Protein Interaction Networks, *Bioinformatics*, vol. 25, no. 7, pp. 933-940, 2009.
4) Goldberg L. A.: Polynomial Space Polynomial Delay Algorithms for Listing Families of Graphs, in *proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, San Diego, California, United States, pp. 218-225, 1993.
5) Kawano S. and Nakano S.: Generating All Series-parallel Graphs, *IEICE Transactions*, vol. 88-A, no. 5, pp. 1129-1135, 2005.
6) Ishida Y., Zhao L., Nagamochi H., and Akutsu T.: Improved Algorithms for Enumerating Tree-like Chemical Graphs with Given Path Frequency, *Genome Informatics*, vol. 21, pp. 53-64, 2008.
7) Li G. and Ruskey F.: The Advantage of Forward Thinking in Generating Rooted and Free Trees, in *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 939-940, 1999.
8) Nakano S. and Uno T.: Efficient Generation of Rooted Trees, NII Technical Report (NII-2003-005), 2003, available on http://research.nii.ac.jp/TechReports/03-005E.html.
9) Nakano S. and Uno T.: Generating Colored Trees, *Lecture Notes in Computer Science*, vol. 3787, pp. 249-260, 2005.
10) Ruskey F. and Hu T. C.: Generating Binary Trees Lexicographically, *SIAM Journal of Computing*, vol. 6, pp. 745-758, 1977.