A BIJECTION BETWEEN REALIZERS OF MAXIMAL PLANE GRAPHS AND PAIRS OF NON-CROSSING DYCK PATHS

NICOLAS BONICHON

ABSTRACT. Schnyder trees, or realizers, of maximal plane graphs, are widely used in the graph drawing domain. In this paper, a bijection between realizers and pairs of non-crossing Dyck paths is proposed. The transformation of a realizer into a pair of non-crossing Dyck paths and the opposite operation can be done in linear time. Applying this bijection, we enumerate the number of realizers of size n and we can efficiently generate all of them.

RÉSUMÉ. Les arbres de Schnyder, ou réaliseurs, d'un graphe maximal planaire, sont largement répandus dans le domaine du dessin de graphe. Nous proposons ici, une bijection entre les réaliseurs et les paires de chemins de Dyck qui ne se coupent pas. La transformation d'un réaliseur en une paire de chemin de Dyck et son inverse se font en temps linéaire. Utilisant cette bijection, nous pouvons énumérer les réaliseurs de taille n et nous pouvons les générer exhaustivement de manière efficace.

1. INTRODUCTION

Schnyder showed that every maximal plane graph admits a special decomposition of its interior edges into three trees (see Fig. 2), called a realizer [16, 17]. Such decomposition can be constructed in linear time [17]. Using realizers, it has been proved in [17] that every plane graph with $n \geq 3$ vertices has a planar straight-line drawing in a rectangular grid area $(n-2) \times (n-2)$.

Realizers are useful for many graph algorithms, of course for graph drawing [17, 4, 1, 13] but also for graph encoding [5]. They are linked to canonical orderings (or shelling orders) [9, 14], with 3-orientations [6], and with orderly spanning trees [4]. They can also be used to characterize planar graphs in terms of the order of their incidence, i.e., a graph G is planar iff the dimension of the incidence order of vertices and edges is at most 3 [16].

Realizers of the same graph have already been investigated [6, 3]. Suitable operations transforming a realizer of a graph to another realizer of the same graph have been introduced [3]. A particular normal form is also characterized. Moreover, the structure of the set of realizers of a given graph turns out to be a distributive lattice [6, 3]. Operations on realizers of same size have also been investigated. In [18] diagonal flip operations have been introduced. For all the maximal planar graphs in [2], colored diagonal flip operations on realizers have been proposed.

Here, we deal with realizers of size n, i.e. realizers of maximal plane graphs of size n. The main motivations are the following: how many realizers of size n are there and how can they be generated. To answer these two questions, a bijection between realizers of size n and pairs of non-crossing Dyck paths of size $2n-6$ is proposed.

A *Dyck path* of size $2n$ is a path in the discrete plane that starts from the point $(0,0)$ and ends at the point $(2n,0)$. It is composed of length $sqrt(2)$ elementary steps North-East and South-East such that it stays in the positive quarter of the plane. (f,g) is a *pair of non-crossing Dyck paths* if g never goes below f. Such paths have been studied by D. Gouyou-Beauchamps [10, 11]. Pairs of non-crossing Dyck paths are a particular case of *vicious walkers* [8, 7, 12, 15]. In [10], the number of pairs of non-crossing Dyck paths of length $2n$ is calculated: $|V_n| = C_{n+2}C_n - C_{n+1}^2$, where C_n is the Catalan number $\frac{(2n)!}{n!(n+1)!}$.

The principle of the bijection is the following. To each realizer R we can associate a particular realizer R_c, called *star realizer*. A star realizer is a realizer in which the third tree is a star i.e. all the inner vertices are children of the root. A realizer R is totally defined by its associated star realizer R_c and a particular sequence of flips, called a *prefix flip sequence*, which transforms R_c into R. The star realizer and the prefix flip sequence can be encoded by two non-crossing Dyck paths of size $2n-6$, where n is the size of the realizer. The star realizer is totally defined by its first tree T_0. T_0 is encoded by a Dyck path. The prefix flip sequence is encoded by a second Dyck path obtained from the first one by local transformations.

The rest of this paper is organized as follows. In Section 2, realizers are presented and some of their properties are given. Star realizers and the prefix flip sequence are introduced in the section 3. The bijection between realizers and non-crossing Dyck paths is explained in section 4.

2. Realizers

2.1. **Definitions.** We assume that the reader is familiar with graph theory. In this paper we deal with simple and undirected graphs. A drawing of a graph is a mapping of each vertex to a point of the plane and of each edge to the continuous curve joining the two ends of this edge. A planar drawing, or *plane graph* is a drawing without crossing edges except, eventually, on a common extremity. A graph that has a planar drawing is a planar graph. A plane graph splits the plane into topologically connected regions, called *face regions*. A *face* is the counter-clockwise walk of the boundary of a face region. One of the regions is unbounded and its associated face is named the *external face* of the plane graph. The vertices and edges of this face are called *external vertices* and *external edges*. The other vertices are called *inner* ones. The *adjacency list* of a vertex u is the list of neighbors of u. In plane graphs, the neighbors of u are ordered in the clockwise order in the adjacency list of u.

A planar graph G is *maximal* (or *triangulated*) if all the other graphs with a same number of vertices that contain it are not planar. The faces of a maximal plane graph are triangular. In this case, we denote v_0, v_1, v_2 the three vertices of the external face of this plane graph.

Definition 1. *(Schnyder* [16]*)*

A realizer of a maximal plane graph G is a partition of the interior edges of G in three sets T_0, T_1, T_2 of directed edges such that for each interior vertex u there holds:

1. *u has out-degree exactly one in each of T_0, T_1, T_2.*
2. *The counter-clockwise order of the edges incident on u is: leaving in T_0, entering in T_2, leaving in T_1, entering in T_0, leaving in T_2 and entering in T_1.*

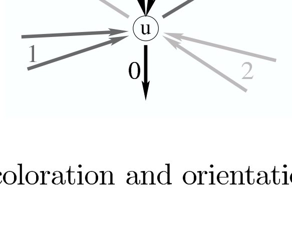


FIGURE 1. Edge coloration and orientation around a vertex

Schnyder show that, T_0, T_1 and T_2 were three *ordered rooted trees* where their edges are oriented toward their roots, which are the external vertices v_0, v_1, v_2. Each tree contains $n-2$ vertices.

An example of a graph, and a realizer of this graph are given in Figure 2.

FIGURE 2. An example of a realizer (a graph on the left side, and one of its realizers on the right side).

In the sequel, the edges of the tree T_i are colored with color i, where $i \in \{0,1,2\}$ such that the external edges $(v_i,\ v_{i+1})$ are of the color $i+1$.

$u_1 \xrightarrow{i} u_2$ denotes the path colored i from u_1 to u_2. We write $u_1 >^i_{ccw} u_2$ (resp. $u_1 >^i_{cw} u_2$) if u_1 is after u_2 in the *counter-clockwise preordering* (resp. *clockwise preordering*) of the tree T_i. The parent of u in the tree T_i is denoted by $P_i(u)$. Let $Ch_i(u)$ be the list of children of u in clockwise order. $Ch_i(u,k)$ denotes the k^th child of the vertex u in T_i. We denote by $deg_i(u)$ the number of ingoing edges (number of children), of u in T_i. If u is not the element of $Ch_i(P_i(u))$, its predecessor u' in $Ch_i(P_i(u))$ is the *left brother* u. The *right branch* of a vertex u in a tree T, is the path in T that joins the right most leaf of the subtree of u to u. The *length* of a right branch is the number of edges of the right branch.

2.2. Diagonal Flips on Realizers.

Let $\mathcal{R}_n$ be the set of realizers of graphs of size n.

In [18], R. Wagner proved that it is possible to obtain all maximal planar graphs of size n using a rewriting rule, called a diagonal flip. In this section, we extend this result to realizers using colored flips.

Definition 2. *Let G be an embedded graph. Let u_2, u_1, u_4 and u_3, u_4, u_1 be two adjacent faces where u_0 is not neighbor of u_3. A diagonal flip consists of removing the edge (u_1, u_4) and inserting the edge (u_2, u_3).*

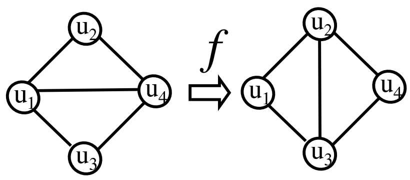


FIGURE 3. Diagonal flip operation

Theorem 1. *(Wagner [18])*

Let G_1 and G_2 be two maximal planar graphs with n vertices. There exists a sequence of diagonal flips that transforms G_1 into G_2.

2.3. **Generalization to realizers.** As shown in Figure 4, we propose colored diagonal flips for realizers using two kinds of flips: f_1^i and f_2^i. It is easy to see that the application of a diagonal flip f_1^i or f_2^i on a realizer gives another realizer.

The choice between f_1^i and f_2^i depends on the quadrilateral configuration. Note that if, the edge (u_2, u_1) is colored $i-1$ and oriented towards u_1, and if the edge (u_3, u_1) is colored $i+1$ and oriented towards u_1, then $f_1^i(u_1)$ or $f_2^i(u_1)$ can be applied.

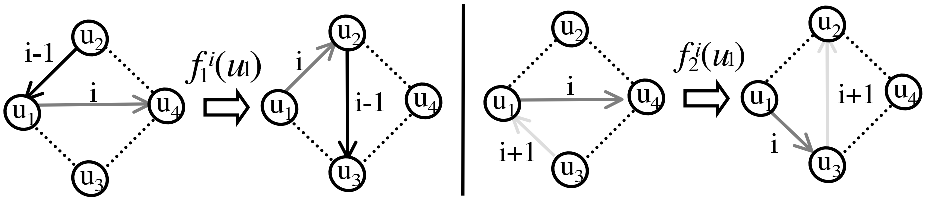


FIGURE 4. Flips on realizer

Unfortunately, these two operations are not always possible to apply. This occurs for the configuration of the quadrilateral of Figure 5.

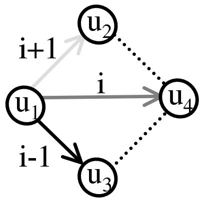


FIGURE 5. Configuration for which colored flip cannot be directly applied.

Theorem 2. [2] *There exists a sequence of colored flips that transforms any realizer R with n vertices into any other realizer R' with n vertices.*

3. Star realizers and prefix flip sequence

In this section, we present a particular class of realizers, called *star realizers*. Using these particular realizers, we construct in a unique way all realizers with a *prefix flip sequence.*

3.1. Star realizers.

Definition 3. *A star realizer $R_c = (T_0, T_1', C_{n-2})$ is a realizer where C_{n-2} is a star of size $n-2$ where all the edges are oriented toward the center of the star, i.e. C_{n-2} is a rooted tree of depth 1.*

In the first realizer of Figure 7, the vertex v_2 is a neighbor of all inner vertices of the graph. So this realizer is a star realizer.

Property 1. *Let T_0 be an ordered rooted tree of size $n-2$. There is a unique tree T_1' such that $R_c = (T_0, T_1', C_{n-2})$ is a star realizer.*

Proof. First, one can remark that there is only one way to connect T_0 and C_{n-2}: the clockwise prefix order in T_0 is the counter-clockwise order around v_2. Once T_0 and C_{n-2} are connected, we obtain a planar map. Let $F_k = (v_2, u_k, u_{k_1}, u_{k_2}, \dots, u_{k_p}, u_{k+1})$ be a face of this planar map (see Figure 6). The parent in T_1' of the vertices $u_k, u_{k_1}, u_{k_2}, \dots, u_{k_{p-1}}$ must be a vertex of this face. This is the only way to satisfy the second item of the definition 1 (see Figure 1). For the same reason the only vertex which can be the parent of $u_k, u_{k_1}, u_{k_2}, \dots, u_{k_{p-1}}$ is the vertex u_{k+}. For each vertex u_k, only one vertex can be the parent of u_k in T_1'. Hence, there is only one tree T_1' such that $R_c = (T_0, T_1', C_{n-2})$ is a realizer. ⊓

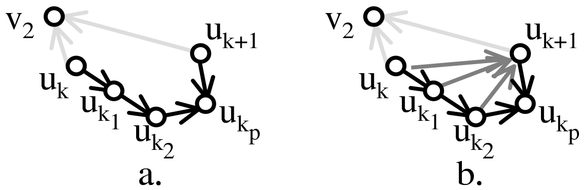


FIGURE 6. **a.** Face of the planar map obtained from the connection of T_0 and C_{n-2}. **b.** The same face, with the edges of T_1' inside

From the construction of the tree T_1' in the proof of property 1, the following property comes directly

Property 2. *Let $R_c = (T_0, T_1', T_2)$ be a star realizer. Let G_c the maximal plane graph of R_c. Let $(u_1, ..., u_{n-3})$ be the inner vertices of G_c in the clockwise prefix order of T_0. The number of children of u_k in T_1' is the length of the right branch of its left brother.*

3.2. **Prefix Flip sequence.** In this section, we will denote by $deg_1(u)$ the number of children of u in the tree T_1. Similarly, we will also denote by $deg_1'(u)$ the number of children of u in the tree T_1'.

Definition 4. *Let $R_c = (T_0, T_1', C_{n-2})$ be a star realizer. The inner vertices u_k of G are numbered respecting the prefix order of T_0. A Prefix Flip Sequence, or PFS, is a sequence of flips $(f_1^2(u_{k_1}), f_1^2(u_{k_2}), \dots, f_1^2(u_{k_p}))$ that can be applied to R_c such that $i < j \Rightarrow k_i \leq k_j$.*

In the sequel, a PFS will be represented by a list of $n-2$ numbers, specifying the number of flips to apply on each inner vertex. For example, the PFS $(f_1^2(u_3), f_1^2(u_4), f_1^2(u_4))$ is represented by $(0,0,1,2)$. $\#f(u_k)$ denotes the number of flips on u_k in the PFS. In the previous sequence, $\#f(u_4) = 2$. Figure 7 shows this PFS applied to a star realizer.

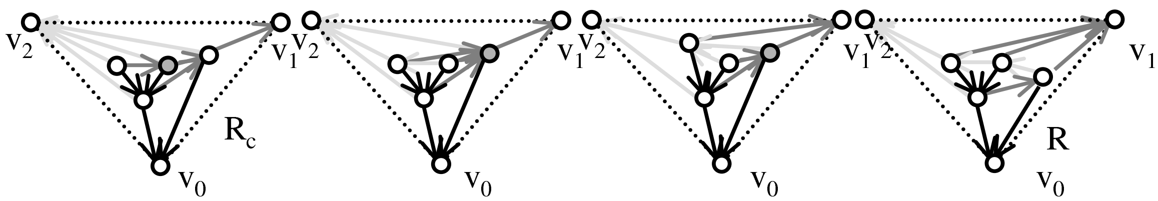


FIGURE 7. Example of prefix flip sequence $(0,0,1,2)$

Remark 1. *A prefix flip sequence does not change the tree T_0 of the star realizer.*

Property 3. *Let $R_c = (T_0, T_1', C_{n-2})$ be a star realizer. Let S be a prefix flip sequence and $R = (T_0, T_1, T_2)$ be the realizer obtained from R_c by S. For each inner vertex u_k, we have: $deg_1(u_k) = deg_1'(u_k) + \#f(u_{k-1}) - \#f(u_k)$*

Proof. The property can be reformulated in the following way: when a flip is applied on u_k in S, $deg_1(u_k)$ is decremented and $deg_1(u_{k+1})$ is incremented. Obviously, when a flip $f_1^2(u_k)$ is applied, $deg_1(u_k)$ is decremented. Let us show, that when a flip $f_1^2(u_k)$ is applied,

$deg_1(u_{k+1})$ is incremented. For this purpose, let us prove by induction on k that when a flip can be applied on u_k, u_{k+1} is just after $P_2(u_k)$ in the adjacency list of u_k.

First we can remark, that in a the star realizer R_c, for each k, u_{k+1} is just after $P_2(u_k)$ in the adjacency list of u_k.

Assume that after applying the flips of S on the $k-1$ first vertices, u_{i+1} is just after $P_2(u_i)$ in the adjacency list of u_i for all $i \geq k$. After applying $f_1^2(u_k)$, u_{k+1} is still just after $P_2(u_k)$ in the adjacency list of u_k (see Fig. 4). So for the $\#f(u_k)$ flips on u_k, u_{k+1} is just after $P_2(u_k)$ in the adjacency list of u_k.

Moreover, the modifications made by the flips $f_1^2(u_k)$ are enclosed in the region $(v_2, u_{k+1}, u_{k+1} \underrightarrow{0} v_0)$. Hence, $P_2(u_{k+1})$ in the adjacency list of u_{k+1} and for each $i > k+1$, the adjacency list of u_i is unchanged.

Hence, in a prefix flip sequence, each time we operate a flip $f_1^2(u_k)$, the number of children of u_k in T_1 is decremented and the number of children of u_{k+1} is incremented. □

Remark 2. *The property 3 can be also expressed:* $\#f(u_k) = deg_1'(u_k) + \#f(u_{k-1}) - deg_1(u_k)$.

Lemma 1. *Let* $R = (T_0, T_1, T_2)$ *be a realizer and* $R_c = (T_0, T_1', C_{n-2})$ *be its star realizer. There exists a unique prefix flip sequence* S_cw *that transforms* R_c *into* R.

Proof. Existence: Let R be a realizer. Let us consider the following algorithm:

for each vertex u_k in counter prefix order of T_0 **do**
  while u_k is not a neighbor of v_2 **do**
    Make the flip $f_2^1(P_2(u_k))$
  end while
end for

We cannot operate an infinite number of times the flip $f_2^1(P_2(u_k))$. Hence the algorithm terminates. When this algorithm ends, a star realizer is obtained, since all the inner vertices are neighbors of v_2. The reverse of the flip $f_2^1(P_2(u_k))$ is the flip $f_1^2(u_k)$ (see Figure 4). The reverse of the sequence of flips built by the previous algorithm is a prefix flip sequence. Hence, for every realizer R, there exists a flip sequence that transforms the star realizer R_c of R into R.

Unicity: Two realizers with two different star realizers cannot be identical since, they will have different trees T_0. Let R_c be a star realizer. Let S_{cw1} and S_{cw2} be two prefix flip sequences. Let R_1 (resp. R_2) be the realizer obtained from R_c by the flip sequence S_{cw1} (resp. S_{cw1}). Let k be the first index where the two sequences are different. The $deg_1(u_k)$ in R_1 is different from $deg_1(u_k)$ in R_2 (see property 3). Hence, a star realizer with two different sequences cannot produce the same realizer. □

Remark 3. *If the flips can be operated in any order, there are several ways to transform* R_c *into* R. *For example, the sequence* $(f_1^2(u_4), f_1^2(u_3))$ *transforms the star realizer* R_c *into the realizer* R *of Figure 7.*

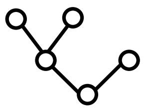
4. Encoding and Decoding Realizers

In this section the bijection between realizers and pairs of non-crossing Dyck words is presented. This bijection is described as an encoding scheme. A first Dyck word is used to encode a star realizer. Precisely, this word encodes the tree T_0 of the star realizer. Then, a second Dyck word, which is used to encode the prefix flip sequence, is obtained from the first one by applying some permutations. In the decoding process, the first word is used to reconstruct the tree T_0 and the star realizer. The difference between the two words encodes the PFS.

4.1. **Non-crossing Dyck paths.** We use a finite set called *alphabet* where the elements are called *letters*. Here, we will use the alphabet $A = \{'(',')'\}$. A *word* is a finite sequence of letters denoted by $f = f_1 f_2 \dots f_n$. The set A^* of all words on the alphabet A is equipped with the concatenation. The length of a word, denoted by $|f|$, is the number of letters of f. For a letter x, $|f|_x$ denotes the number of letters x in the word f. A word f' is a left factor of f if there exists a word f'' such that $f = f'f''$. A morphism δ from A^* to $\mathbb{N}$ is defined by: $\delta('(') = 1$, $\delta(')') = -1$ and $\delta(f'f'') = \delta(f') + \delta(f'')$. The *Dyck language* is the following: $D = \{f \in A^* | \delta(f) = 0 \text{ and } \forall f' \text{ left factor of } f, \delta(f') \geq 0\}$. We denote by $D_n = D \cap A^{2n}$. We denote by $open(k, f)$ the position of the k^{th} opening parenthesis in f.

Dyck paths are paths coded by Dyck words. A step North-East is coded by $'('$ and a step South-East is coded by $')'$. These paths start from the point $(0, 0)$, never go bellow the x-axis and end on the x-axis. Dyck words of $2n - 2$ length are classically used to encode ordered rooted trees of size n. Figure 8 a. shows an ordered rooted tree and its coding with a Dyck word.

The pair (g, h) of $D_n \times D_n$ are *non-crossing Dyck words* if for all g' (resp. h') left factor of g (resp. h) such that $|g'| = |h'|$, $\delta(h') \geq \delta(g')$. V_n denotes the set of pairs of non-crossing words of $D_n \times D_n$. Obviously, a pair of non-crossing Dyck words encodes a pair of non-crossing Dyck paths. Figure 8 b. shows an example of non-crossing Dyck paths.



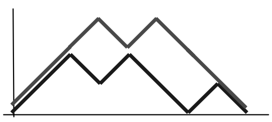


FIGURE 8. **a.** Encoding an ordered rooted tree with a Dyck word. **b.** Example of non-crossing Dyck paths

Theorem 3. *(Gouyou-Beauchamps* [10]*)*
$|V_n| = C_{n+2}C_n - C_{n+1}^2$, *where* C_n *is the Catalan number* $\frac{(2n)!}{n!(n+1)!}$.

The first values of $|V_n|$ are $1, 1, 3, 14, 84, 594, 4719, \dots$

Algorithm 1 Encoding algorithm

Build the corresponding star realizer R_c of R
Code the tree T_0 by a Dyck word g.
$h \leftarrow g$
for each vertex u_k in the prefix order of T_0 **do**
  $\#f(u_k) \leftarrow deg_1'(u_k) - deg_1(u_k) + \#f(u_{k-1})$
  Move $open(k, h)$ of $\#f(u_k)$ ranks to the left in h.
end for

4.2. **Encoding.**

Property 4. *In the algorithm 1, the number of flips on* u_k *is less than or equal to the number of consecutive closing brackets just before* $open(k, h)$ *in* h.

Proof. When no flips are previously made, $h = g$. The number of consecutive closing brackets just before $open(k, h)$ in h is exactly the length of the right branch of its left brother. $deg'_1(u_k)$ is equal to the length of the right branch of its left brother (see property 2). As $\#flips(u_k) \leq deg'_1(u_k)$, the property is satisfied in this case.

Suppose that the property is verified for $i \leq k-1$. The number of consecutive closing brackets just before $open(k,h)$ is $deg'_1(u_k) + \#flips(u_{k-1})$. As $\#flips(u_k) \leq deg'_1(u_k) + \#flips(u_{k-1})$ (see Remark 2) the property is still satisfied for $i = k$. □

Lemma 2. *The previous algorithm encodes a realizer R of size n with a pair of non-crossing Dyck words of lengths $2n-6$. This algorithm is linear time.*

Proof. First we can notice that g and h are non-crossing Dyck words.

Injectivity: Let $R = (T_0, T_1, T_2)$ and $R' = (T'_0, T'_1, T'_2)$ be two different realizers. Let (g, h) (resp. (g', h')) the pair of non-crossing Dyck words obtained by the previous algorithm from R (resp. R'). If $T_0 \neq T'_0$, then $g \neq g'$. Let S_f (resp. S'_f) be the PFS associated to R (resp. R'). Let k be the first index such that $\#f(u_k) \neq \#f'(u_k)$. After the k_{th} step in the loop, $open(k,h) \neq open(k,h')$. During the rest of the algorithm $open(k,h)$ and $open(k,h')$ are unchanged, so $h \neq h'$. Hence two different realizers are encoded with two different pairs of non-crossing words.

Complexity: To construct the star realizer, all vertices of T_0 are connected with an out-going edge, colored 2, to v_2 and in-going edges, colored 1, to all the vertices of the right branch of its left brother. This construction can be done in linear time. The encoding of T_0 with the traditional algorithm is also done in linear time. The treatment of each vertex u_k is done in constant time. Hence the encoding algorithm computes in linear time. □

Example: to encode the realizer R of Figure 7, we can encode its star realizer R_c and the PFS $(0,0,1,2)$. The tree T_0 of R is the one of Figure 8 a. It can be encoded by $g = (()())()$. To encode the flip sequence, we need to move the third opening bracket of one step to the left and the fourth one to two steps to the left. So $h = ((()()))$. Hence, the realizer R of Figure 7 is encoded by the pair of non-crossing Dyck words (g, h).

4.3. **Decoding.** Let (g, h) be a pair of non-crossing words.

The function *Concat(L_1, L_2)* append the list L_2 at the end of the list L_1 and returns this new list. The function *Split(L, i)* removes the last i elements of L and returns a list which contains these i elements. The procedure *AddFirst(L, E)* added the element E at the begining of the list L. Naturally, *Del(E, i)* removes the i^{th} element of the list L.

Algorithm 2 Decoding algorithm

Build the tree T_0 from g
Build the star realizer $R_c = (T_0, T'_1, C_{n-2})$
$R = (T_0, T_1, T_2) \leftarrow R_c$
for each vertex u_k in the prefix order of T_0 **do**
  $\#f(u_k) \leftarrow open(g,k) - open(h,k)$
  $L \leftarrow Split(Ch_1(u_k), \#f(u_k))$
  $Ch_1(u_{k+1}) \leftarrow Concat(Ch_1(u_{k+1}), L)$
  $Del(Ch_2(P_2(u_k)), u_k)$
  $AddFirst(Ch_2(Ch_1(u_{k+1}, 0)), u_k)$
end for

Lemma 3. *The algorithm 2 computes in linear time a realizer R of size n from a pair of non-crossing Dyck words of lengths $2n-6$.*

Proof. Validity: as $h \geq g$, $0 \leq \#f(u_k) \leq deg'_1(u_k) + \#flips(u_{k-1})$ encodes a star Realizer and a valid PFS. Moreover the algorithm 2 constructs the realizer encoded by the algorithm 1.

Complexity: as in the encoding algorithm, the construction of the star realizer can be operated in linear time. The algorithm uses chained lists to store the list of children of vertices in each tree. The split operation in the loop takes $O(deg_1(u_k))$ operations. Globally, it takes $O(m) = O(n)$ operations. The other operations in the loop take $O(1)$ operations. So globally it takes $O(n)$ operations. □

The following theorem comes directly from lemma 2 and lemma 3:

Theorem 4. *There is a bijection between realizers of size n and pairs of non-crossing Dyck paths of lengths* $2n - 6$.

Corollary 5. *The number of realizer of size* n *is* $|R_n| = |V_{n-3}| = C_{n-3}C_{n-1} - C^2_{n-2}$.

Acknowledgments

The author thanks Philippe Duchon, Mireille Bousquet-Mélou and Xavier Gérard Viennot for fruitful discussions and suggestions. Much thanks to Mohamed Mosbah for his help with writing the paper.

References

[1] G. Di Battista, R. Tamassia, and L.Vismara. Output-sensitive reporting of disjoint paths. In *Proc. 2nd Int. Conf. Computing and Combinatorics*, number 1090, pages 81–91. Springer-Verlag, 1996.
[2] N. Bonichon, B. Le Saëc, and M. Mosbah. Diagonal flip operations on realizers and their application to wagner's theorem on realizers. In *Applied Graph Transformation 2002 (AGT2002), Grenoble 12-13th April, France*, 2002.
[3] E. Brehm. *3-orientations and schnyder 3-Tree-Decompositions.* PhD thesis, FB Mathematik und Informatik, Freie Universität Berlin, 2000.
[4] Yi-Ting Chiang Ching-Chi and Hsueh-I Lu. Orderly spanning trees with applications to graph encoding and graph drawing. In *Proc. 12th Symp. Discrete Algorithms*, pages 506–515. ACM and SIAM, 2001.
[5] Richie Chih-Nan Chuang, Ashim Garg, Xin He, Ming-Yang Kao, and Hsueh-I Lu. Compact encodings of planar graphs via canonical ordering and multiple parentheses. In *Proc. 25th International Colloquium on Automata, Languages, and Programming (ICALP'98)*, volume 1443, pages 118–129, 1998.
[6] P. Ossona de Mendez. *Orientations bipolaires.* PhD thesis, Ecole des Hautes Etudes en Sciences Sociales, 1994.
[7] J.W. Essam and A.J. Guttmann. Vicious walkers and directed polymer networks in general dimension. *Phys. Rev. E 52*, pages 5849–5862, 1995.
[8] M. E. Fisher and M. P. Gelfand. The reunions of three dissimilar vicious walkers. *J. Stat. Phys 53*, pages 175–189, 1988.
[9] H. De Frayseix, J. Pach, and J. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10:41–51, 1990.
[10] D. Gouyou-Beauchamps. Chemins sous-diagonaux et tableaux de young. In *Colloque de Combinatoire Enumrative, Montral, UQAM 1985, Lecture Notes in Mathematics*, volume 1234, pages 112–125, 1986.
[11] D. Gouyou-Beauchamps. Standard young tableaux of height 4 and 5. *Europ. J. Combinatorics*, 10:69–82, 1989.
[12] A. J. Guttmann, A. L. Owczarek, and X. G. Viennot. Vicious walkers and young tableaux i: Without walls. *J. Phys. A; Math. Gen.*, 31:8123–8135, 1998.
[13] G. Kant. Drawing planar graphs using the lmc-ordering. In *Proc. 33th Ann. IEEE Symp. on Found. of Comp. Science*, pages 101–110, 1992.
[14] G. Kant. Drawing planar graphs using the canonical ordering. *Algorithmica*, 16:4–32, 1996.
[15] C. Krattenthaler, A. J. Guttmann, and X. G. Viennot. Vicious walkers, friendly walkers and young tableaux ii: with a walls. *J. Phys. A; Math. Gen.*, 33:8123–8135, 2000.
[16] W. Schnyder. Planar graphs and poset dimension. *Order*, 5:323–343, 1989.
[17] W. Schnyder. Embedding planar graphs on the grid. *Proc. 1st ACM-SIAM Symp. Discrete Algorithms*, pages 138–148, 1990.

[18] K. Wagner. Bemerkungen zum vierfarbenproblem. In *Jahresber. Deutsche Math. -Verein.*, volume 46, pages 26–32, 1936.

LaBRI-Université Bordeaux 1, 351 Cours de la Libération, 33405 Talence, France
E-mail address: bonichon@labri.fr