



ELSEVIER

Available online at www.sciencedirect.com ScienceDirect

Journal of Combinatorial Theory, Series A 114 (2007) 931–956

Journal of
Combinatorial
Theory

Series A

www.elsevier.com/locate/jcta

Bijjective counting of Kreweras walks and loopless triangulations

Olivier Bernardi

LaBRI, Université Bordeaux 1, 351 cours de la Libération, 33405 Talence cedex, France

Received 15 May 2006

Available online 4 December 2006

Abstract

We consider lattice walks in the plane starting at the origin, remaining in the first quadrant $i, j \geq 0$ and made of West, South and North-East steps. In 1965, Germain Kreweras discovered a remarkably simple formula giving the number of these walks (with prescribed length and endpoint). Kreweras' proof was very involved and several alternative derivations have been proposed since then. But the elegant simplicity of the counting formula remained unexplained. We give the first purely combinatorial explanation of this formula. Our approach is based on a bijection between Kreweras walks and triangulations with a distinguished spanning tree. We obtain simultaneously a bijective way of counting loopless triangulations.

© 2006 Elsevier Inc. All rights reserved.

Keywords: Planar walk; Kreweras walk; Planar map; Triangulation; Cubic map; Bijection; Counting

1. Introduction

We consider lattice walks in the plane starting from the origin $(0, 0)$, remaining in the first quadrant $i, j \geq 0$ and made of three kind of steps: West, South and North-East. These walks were first studied by Kreweras [4] and inherited his name. A Kreweras walk ending at the origin is represented in Fig. 1.

These walks have remarkable enumerative properties. Kreweras proved in 1965 that the number of walks of length $3n$ ending at the origin is:

$$k_n = \frac{4^n}{(n+1)(2n+1)} \binom{3n}{n}. \quad (1)$$

E-mail address: bernardi@labri.fr.

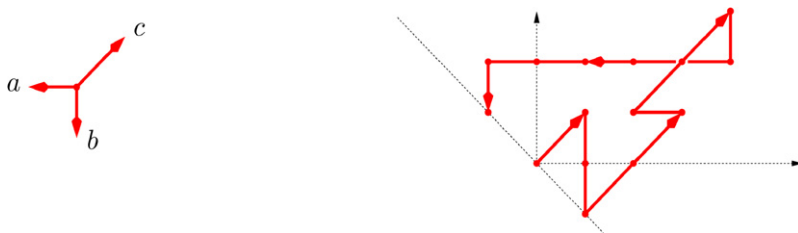


Fig. 2. An excursion.

$i + j \geq 0$. We borrow a terminology from probability theory and call these walks *meanders*. We call *excursion* a meander ending on the second diagonal (i.e. the line $i + j = 0$). An excursion is represented in Fig. 2.

Unlike Kreweras walks, excursions are easy to count. By applying the cycle lemma (see [9, Section 5.3]), we prove that the number of excursions of size n (length $3n$) is

$$e_n = \frac{4^n}{2n + 1} \binom{3n}{n}.$$

- In Section 4, we define a mapping Φ between excursions and cubic maps with a distinguished depth tree. In Section 5 we prove that the mapping Φ is an $(n + 1)$ -to-1 correspondence Φ between excursions (of size n) and bridgeless cubic maps (of size n) with a distinguished depth tree. As a consequence, the number of bridgeless cubic maps of size n with a distinguished depth tree is found to be:

$$d_n = \frac{e_n}{n + 1} = \frac{4^n}{(n + 1)(2n + 1)} \binom{3n}{n}.$$

- In Section 6, we prove that the correspondence Φ , restricted to Kreweras walks, induces a *bijection* between Kreweras walks (of size n) ending at the origin and bridgeless cubic maps (of size n) with a distinguished depth tree. As a consequence, we obtain:

$$k_n = d_n = \frac{4^n}{(n + 1)(2n + 1)} \binom{3n}{n},$$

where k_n is the number of Kreweras walks of size n ending at the origin. This gives a combinatorial proof of (1).

- In Section 7, we enumerate depth trees on cubic maps. We prove that the number of such trees for a cubic map of size n is 2^n . As a consequence, the number of cubic maps of size n is

$$c_n = \frac{d_n}{2^n} = \frac{4^n}{(n + 1)(2n + 1)} \binom{3n}{n}.$$

This gives a combinatorial proof of (2).

- In Section 8, we extend the mapping Φ to Kreweras walks ending at $(i, 0)$ and discuss some open problems.

3. Preliminaries

3.1. Planar maps and depth trees

3.1.1. Planar maps

A *planar map*, or *map* for short, is an embedding of a connected planar graph in the sphere without intersecting edges, defined up to orientation preserving homeomorphisms of the sphere. Loops and multiple edges are allowed. The *faces* are the connected components of the complement of the graph. By removing the midpoint of an edge we obtain two *half-edges*, that is, one-dimensional cells incident to one vertex. We say that each edge has two *half-edges*, each of them incident to one of the endpoints.

A map is *rooted* if one of its half-edges is distinguished as the *root*. The edge containing the root is the *root-edge* and its endpoint is the *root-vertex*. Graphically, the root is indicated by an arrow pointing to the root-vertex (see Fig. 3). All the maps considered in this paper will be rooted without further mention.

3.1.2. Growing maps

Our constructions lead us to consider maps with some *legs*, that is, half-edges that are not part of a complete edge. A *growing map* is a (rooted) map together with some legs, one of them being distinguished as the *head*. We require the legs to be (all) in the same face called *head-face*. The endpoint of the head is the *head-vertex*. Graphically, the head is indicated by an arrow pointing away from the head-vertex. The root of a growing map can be a leg or a regular half-edge. For instance, the growing map in Fig. 4 has 2 legs beside the head, and its root is not a leg.

3.1.3. Cubic maps

A map (or growing map) is *cubic* if every vertex has degree 3. It is *k*-near-cubic if the root-vertex has degree *k* and any other vertex has degree 3. For instance, the map in Fig. 3 is 2-near-cubic and the growing map in Fig. 4 is cubic. Observe that cubic maps are in bijection with 2-near-cubic maps not reduced to a loop by the mapping illustrated in Fig. 5.

The incidence relation between vertices and edges in cubic maps shows that the number of edges is always a multiple of 3. More generally, if *M* is a *k*-near-cubic map with *e* edges

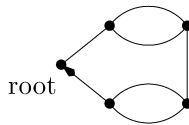


Fig. 3. A rooted map.

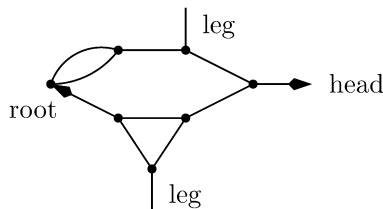


Fig. 4. A growing map.

and v vertices, the incidence relation reads: $3(v - 1) + k = 2e$. Equivalently, $3(v - k + 1) = 2(e - 2k + 3)$. The number $v - k + 1$ is non-negative for *non-separable* k -near-cubic maps (see definition below). (This property can be shown by induction on the number of edges by contracting the root-edge.) Hence, the number of edges has the form $e = 3n + 2k - 3$, where n is a non-negative integer. We say that a k -near-cubic map has *size* n if it has $e = 3n + 2k - 3$ edges (and $v = 2n + k - 1$ vertices). In particular, the mapping of Fig. 5 is a bijection between cubic maps of size n ($3n + 3$ edges) and 2-near-cubic maps of size $n + 1$ ($3n + 4$ edges).

3.1.4. Non-separable maps

A map is *non-separable* if its edge set cannot be partitioned into two non-empty sets such that only one vertex is incident to some edges in both sets. In particular, a non-separable map not reduced to an edge has no loop nor bridge (a *bridge* or *isthmus* is an edge whose deletion disconnects the map). For cubic maps and 2-near-cubic maps it is equivalent to be non-separable or bridgeless. The mapping illustrated in Fig. 5 establishes a bijection between bridgeless cubic maps and bridgeless 2-near-cubic maps not reduced to a loop.

Bridgeless cubic maps are interesting because their *duals* are the loopless triangulations. Recall that the dual M^* of a map M is the map obtained by putting a vertex of M^* in each face of M and by connecting the vertices by edges crossing the edges of M wherever possible, one edge of M^* across each edge of M . See Fig. 6 for an example.

3.1.5. Depth trees

A *tree* is a connected graph without cycle. A subgraph T of a graph G is a *spanning tree* if it is a tree containing every vertex of G . An edge of the graph G is said to be *internal* if it is in the spanning tree T and *external* otherwise. For any pair of vertices u, v of the graph G , there is a unique path between u and v in the spanning tree T . We call it the T -*path* between u and v . A map (or growing map) M with a distinguished spanning tree T will be denoted by M_T . Graphically, we shall indicate the spanning tree by thick lines as in Fig. 7. A vertex u of M_T is an *ancestor* of another vertex v if it is on the T -path between the root-vertex and v . In this case,

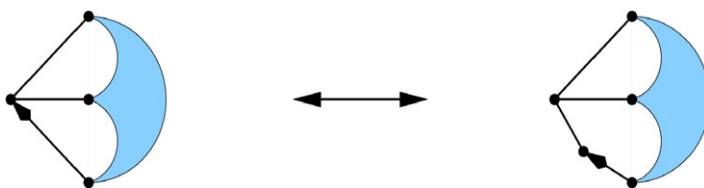


Fig. 5. Bijection between cubic maps and 2-near-cubic maps.

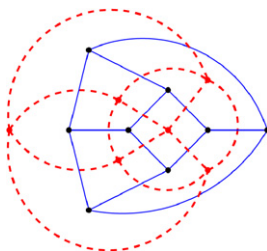


Fig. 6. A cubic map and the dual triangulation (dashed lines).

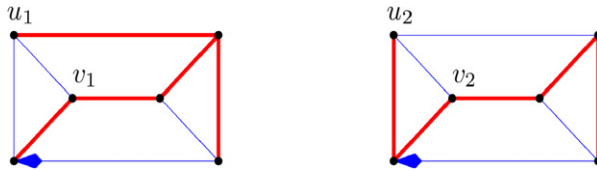


Fig. 7. A depth tree (left) and a non-depth tree (right).

v is called a *descendant* of u . Two vertices are *comparable* if one is the ancestor of the other. For instance, in Fig. 7, the vertices u_1 and v_1 are comparable whereas u_2 and v_2 are not.

A *depth tree* is a spanning tree such that any external edge joins comparable vertices. Moreover, we require the edge containing the root to be external. In Fig. 7, the tree on the left side is a depth tree but the tree on the right side is not a depth tree since the edge (u_2, v_2) breaks the rule. A *depth-map* is a map with a distinguished depth tree. A *marked-depth-map* is a depth-map with a marked external edge.

3.2. Kreweras walks and meanders

In what follows, Kreweras walks are considered as words on the alphabet $\{a, b, c\}$. The letter a (respectively b, c) corresponds to a West (respectively South, North-East) step. For instance, the walk in Fig. 1 is $cbccccbcaaaaabb$. The length of a word w is denoted by $|w|$ and the number of occurrences of a given letter α is denoted by $|w|_\alpha$. Kreweras walks are the words w on the alphabet $\{a, b, c\}$ such that any prefix w' of w satisfies

$$|w'|_a \leq |w'|_c \quad \text{and} \quad |w'|_b \leq |w'|_c. \tag{3}$$

Kreweras walks ending at the origin satisfy the additional constraint

$$|w|_a = |w|_b = |w|_c. \tag{4}$$

These conditions can be interpreted as a ballot problem with three candidates. This is why Kreweras walks sometimes appear under this formulation in the literature [6].

Similarly, the meanders, that is, the walks remaining in the half-plane $i + j \geq 0$, are the words w on $\{a, b, c\}$ such that any prefix w' of w satisfies

$$|w'|_a + |w'|_b \leq 2|w'|_c. \tag{5}$$

Excursions, that is, meanders ending on the second diagonal, satisfy the additional constraint

$$|w|_a + |w|_b = 2|w|_c. \tag{6}$$

Note that the length of any walk ending on the second diagonal is a multiple of 3. We define the *size* of such a walk of length $3n$ to be n . Note also that a walk ending at point $(i, 0)$ has a length of the form $l = 3n + 2i$ where n is a non-negative integer. We define a Kreweras walk of length $l = 3n + 2i$ ending at $(i, 0)$ to have *size* n .

Unlike Kreweras walks, the excursions are easy to count.

Proposition 3.1. *There are*

$$e_n = \frac{4^n}{2n + 1} \binom{3n}{n} \tag{7}$$

excursions of size n .

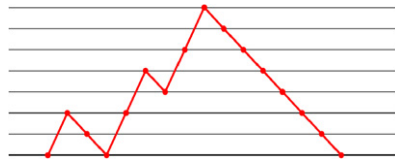


Fig. 8. The projected walk associated to the excursion of Fig. 2.

Proof. We consider *projected walks*, that is, one-dimensional lattice walks starting and ending at 0, remaining non-negative and made of steps +2 and −1. (They correspond to projections of excursions on the first diagonal.) A projected walk is represented in Fig. 8. Projected walks can be seen as words w on the alphabet $\{\alpha, c\}$ with $|w|_\alpha = 2|w|_c$ and such that any prefix w' of w satisfies $|w'|_\alpha \leq 2|w'|_c$. The projected walks can be counted bijectively by applying the cycle lemma (see Section 5.3 of [9]): there are

$$p_n = \frac{1}{3n+1} \binom{3n+1}{2n+1} = \frac{1}{2n+1} \binom{3n}{n}$$

projected walks of size n (length $3n$).

Given an excursion, we obtain a projected walk by replacing the occurrences of a and b by α . Conversely, taking a projected walk of length $3n$ and replacing the $2n$ letters α by a sequence of letters in $\{a, b\}$ one obtains an excursion. This establishes a 4^n -to-1 correspondence between excursions (of size n) and projected walks (of size n). Thus, there are $4^n p_n$ excursions of size n . □

4. A bijection between excursions and cubic marked-depth-maps

In this section we define a mapping Φ between excursions and bridgeless 2-near-cubic *marked-depth-maps* (2-near-cubic maps with a distinguished depth tree and a marked external edge). We shall prove in Section 5 that the mapping Φ is a bijection between excursions and bridgeless 2-near-cubic marked-depth-maps. The general principle of the mapping Φ is to read the excursion from right to left and interpret each letter as an operation for constructing the map and the tree. This step-by-step construction is illustrated in Fig. 10. The intermediate steps are *tree-growing maps*, that is, growing maps together with a distinguished spanning tree (indicated by thick lines).

- We start with the tree-growing map M^0 consisting of one vertex and two legs. One of the legs is the root, the other is the head (see Fig. 9). The spanning tree is reduced to the unique vertex.
- We apply successively certain elementary mappings $\varphi_a, \varphi_b, \varphi_c$ (Definition 4.1) corresponding to the letters a, b, c of the excursion read *from right to left*.
- When the whole excursion is read, there is only one leg remaining beside the head. At this stage, we *close* the tree-growing map, that is, we join the head and the remaining leg into a *marked external edge* as shown in Fig. 11.

Let us enter into the details and define the mapping Φ . Consider a growing map M . We make a *tour* of the head-face if we follow its border in counterclockwise direction (i.e. the border of the



Fig. 9. The tree-growing map M_{\bullet}^0 .

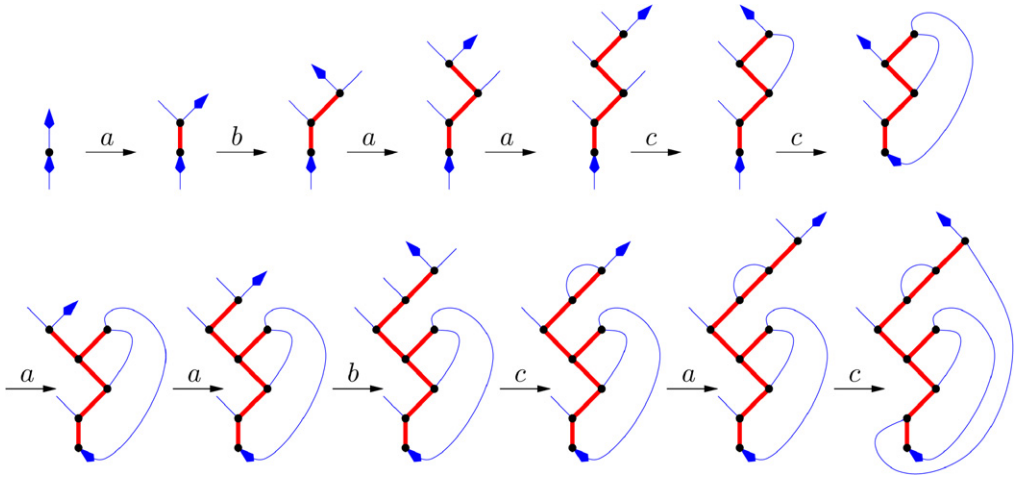


Fig. 10. Successive applications of the mappings $\varphi_a, \varphi_b, \varphi_c$ for the walk $cacbaaccaaba$ (read from right to left).

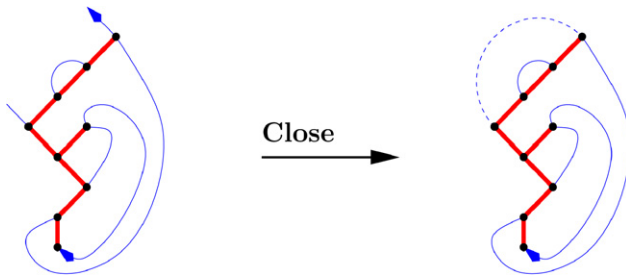


Fig. 11. Closing the map (the marked edge is dashed).

head-face stays on our left-hand side) starting from the head (see Fig. 12). This journey induces a linear order on the legs of M . We shall talk about the *first* and *last* legs of M .

We define three mappings $\varphi_a, \varphi_b, \varphi_c$ on tree-growing maps.

Definition 4.1. Let M_T be a tree-growing map (the map is M and the distinguished tree is T).

- The mappings φ_a and φ_b are represented in Fig. 13. The tree-growing map $M'_{T'} = \varphi_a(M_T)$ (respectively $\varphi_b(M_T)$) is obtained from M_T by replacing the head by an edge e together with a new vertex v incident to the new head and another leg at its left (respectively right). The tree T' is obtained from T by adding the edge e and the vertex v .
- The tree-growing map $\varphi_c(M_T)$ is only defined if the first and last legs exist (that is, if the head-face contains some legs beside the head) and have distinct and comparable endpoints. We

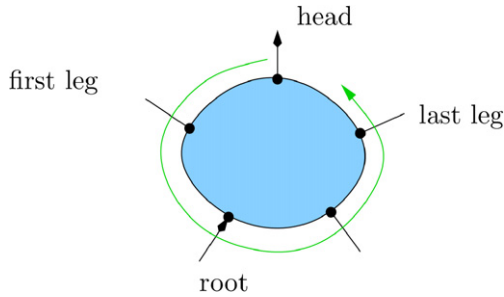


Fig. 12. Making the tour of the head-face.



Fig. 13. The mappings φ_a and φ_b .

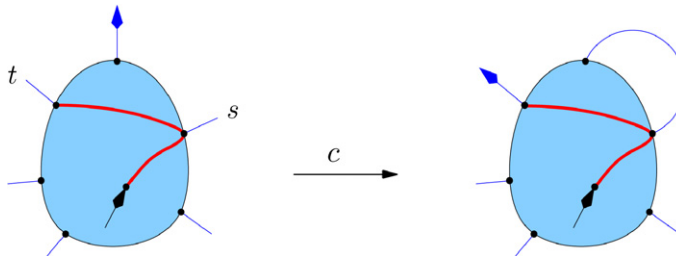


Fig. 14. The mapping φ_c .

call these legs s and t with the convention that the endpoint of s is an ancestor of the endpoint of t .

In this case, the tree-growing map $M'_T = \varphi_c(M_T)$ is obtained from M_T by joining together the head and the leg s while the leg t becomes the new head (see Fig. 14). The spanning tree T is unchanged.

• For a word $w = a_1 a_2 \dots a_n$ on the alphabet $\{a, b, c\}$, we denote by φ_w the mapping $\varphi_{a_1} \circ \varphi_{a_2} \circ \dots \circ \varphi_{a_n}$.

Definition 4.2. The image of an excursion w by the mapping Φ is the map with a distinguished spanning tree and a marked external edge obtained by closing the tree-growing map $\varphi_w(M_\bullet^0)$, that is, by joining the head and the unique remaining leg into a marked edge.

The mapping Φ has been applied to the excursion $cacbaaccaaba$ in Figs. 10 and 11. Of course, we still need to prove that the mapping Φ is well defined.

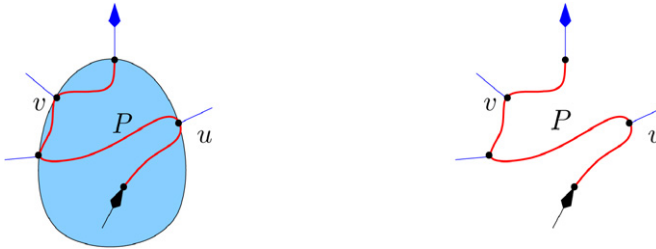


Fig. 15. The last vertex incident to a leg on the T -path from the root-vertex to the head-vertex is v .

Proposition 4.3. *The mapping Φ is well defined on any excursion w :*

- *It is always possible to apply the mapping φ_c when required.*
- *The tree-growing map $\varphi_w(M_\bullet^0)$ has exactly one leg beside the head. This leg and the head are both in the head-face, hence can be joined together.*

Before proving Proposition 4.3, we need three technical results.

Lemma 4.4. *Let w be a word on the alphabet $\{a, b, c\}$ such that $\varphi_w(M_\bullet^0)$ is well defined. Then, $\varphi_w(M_\bullet^0)$ is a tree-growing map.*

Proof. Let $M_T = \varphi_w(M_\bullet^0)$. It is clear by induction that T is a spanning tree. The only point to prove is that the legs of $\varphi_w(M_\bullet^0)$ are in the head-face. We proceed by induction on the length of w . This property holds for the empty word. If the property holds for $M_T = \varphi_w(M_\bullet^0)$ it clearly holds for $\varphi_a(M_T)$ and $\varphi_b(M_T)$. If φ_c can be applied, the head is joined either to the first or to the last leg of M_T . Thus, all the remaining legs (including the head of $\varphi_c(M_T)$) are in the same face. \square

We shall see shortly (Lemma 4.6) that whenever the tree-growing map $\varphi_w(M_\bullet^0)$ is well defined, the endpoints of any leg is an ancestor of the head-vertex. Observe that in this case the endpoints of the legs are comparable.

Lemma 4.5. *Let M_T be a tree-growing map. Suppose that the endpoint of any leg is an ancestor of the head-vertex. Suppose also that the first and last legs exist and have distinct endpoints. We call these endpoints u and v with the convention that u is an ancestor of v . Then, v is the last vertex incident to a leg on the T -path from the root-vertex to the head-vertex.*

Proof. The situation is represented in Fig. 15. We proceed by induction on the number of edges that are not in the T -path P from the root-vertex to the head-vertex. The property is clearly true if the tree-growing map is reduced to the path P plus some legs. If not, the deletion of a edge not in P does not change the order of appearance of the legs around the head-face. In particular, the first and last legs are unchanged. \square

Lemma 4.6. *Let w be a word on the alphabet $\{a, b, c\}$ such that $\varphi_w(M_\bullet^0)$ is defined. Then the endpoint of any leg of $\varphi_w(M_\bullet^0)$ is an ancestor of the head-vertex.*

Proof. We proceed by induction on the length of w . The property holds for the empty word. We suppose that it holds for $M_T = \varphi_w(M_\bullet^0)$. It is clear that the property holds for the tree-growing

maps $\varphi_a(M_T)$ and $\varphi_b(M_T)$. If φ_c can be applied, the endpoints of the first and last leg are distinct and comparable. We call these endpoints u and v with the convention that u is an ancestor of v . By the induction hypothesis, the conditions of Lemma 4.5 are satisfied by M_T . Therefore, the vertex v is the last vertex incident to a leg on the T -path from the root-vertex to the head-vertex. Hence, any endpoint of a leg of $\varphi_c(M_T)$ is an ancestor of v which is the head-vertex of $\varphi_c(M_T)$. \square

Proof of Proposition 4.3. Let w be an excursion. We consider a suffix w' of w and denote by $M'_T = \varphi_{w'}(M_\bullet^0)$ the corresponding tree-growing map (if it is well defined).

- If M'_T is well defined, it has $|w'|_a + |w'|_b - 2|w'|_c + 1$ legs besides the head. (Observe that, by (5) and (6), the quantity $|w'|_a + |w'|_b - 2|w'|_c$ is non-negative.)

We proceed by induction on the length of w' . The property holds for the empty word. Moreover, applying φ_a or φ_b increases by 1 the number of legs whereas applying φ_c decreases this number by 2. Thus, the property follows easily by induction.

- The tree-growing map M'_T is well defined.

We proceed by induction on the length of w' . The property holds for the empty word. We write $w' = \alpha w''$ and suppose that $M''_T = \varphi_{w''}(M_\bullet^0)$ is well defined. If $\alpha = a$ or b the tree-growing map $M'_T = \varphi_\alpha(M''_T)$ is well defined. We suppose now that $\alpha = c$. The tree-growing map M''_T has $|w''|_a + |w''|_b - 2|w''|_c + 1 = |w'|_a + |w'|_b - 2|w'|_c + 3 > 2$ legs besides the head. It is clear by induction that all these legs have distinct endpoints. Moreover, by Lemma 4.6, all the endpoints of these legs are ancestors of the head-vertex. Thus the endpoints of the legs are comparable. In particular, the endpoints of the first and last legs are comparable. Hence, the mapping φ_c can be applied.

- The tree-growing map $M_T = \varphi_w(M_\bullet^0)$ is well defined and has exactly one leg beside the head. This property follows from the preceding points since $|w|_a + |w|_b - 2|w|_c = 0$. \square

We now state the key result of this paper.

Theorem 4.7. *The mapping Φ is a bijection between excursions of size n and bridgeless 2-near-cubic marked-depth-maps of size n .*

The proof of Theorem 4.7 is postponed to the next section. For the time being we explore its enumerative consequences. We denote by d_n the number of bridgeless 2-near-cubic depth-maps of size n . Consider a 2-near-cubic map M of size n ($3n + 1$ edges, $2n + 1$ vertices) and a spanning tree T . Since T has $2n + 1$ vertices, M_T has $2n$ internal edges and $n + 1$ external edges. Hence, there are $(n + 1)d_n$ bridgeless 2-near-cubic marked-depth-maps. By Theorem 4.7, this number is equal to the number e_n of excursions of size n . Using Proposition 3.1, we obtain the following result.

Corollary 4.8. *There are*

$$d_n = \frac{e_n}{n + 1} = \frac{4^n}{(n + 1)(2n + 1)} \binom{3n}{n}$$

bridgeless 2-near-cubic depth-maps of size n .

Observe that d_n is also the number of bridgeless cubic depth-maps of size $n - 1$ since the bijection between cubic maps and 2-near-cubic maps represented in Fig. 5 can be turned into a bijection between cubic depth-maps and 2-near-cubic depth-maps.

5. Why the mapping Φ is a bijection

In this section, we prove that the mapping Φ is a bijection between excursions and bridgeless 2-near-cubic marked-depth-maps. We first prove that the image of any excursion by the mapping Φ is a bridgeless 2-near-cubic marked-depth-map (Proposition 5.1). Then we define a mapping Ψ from bridgeless 2-near-cubic marked-depth-maps to excursions (Definition 5.4) and prove that Φ and Ψ are inverse mappings (Propositions 5.7 and 5.9).

Proposition 5.1. *The image $\Phi(w)$ of any excursion w is a bridgeless 2-near-cubic marked-depth-map.*

Proof. Let w' be a suffix of w and let $M'_T = \varphi_{w'}(M^0_\bullet)$ be the corresponding tree-growing map.

- *The tree-growing map M'_T is 2-near-cubic.*

Applying φ_a or φ_b creates a new vertex of degree 3 and does not change the degree of the other vertices. Applying φ_c does not affect the degree of the vertices. The property follows by induction.

- *The head and the root of M'_T are distinct half-edges.*

The property holds for the empty word. We now write $w' = \alpha w''$. If $\alpha = a$ or b the property clearly holds for w' . Suppose now that $\alpha = c$. Let u and v be the vertices incident to the first and last legs of $M''_T = \varphi_{w''}(M^0_\bullet)$ with the convention that u is an ancestor of v . By definition, v is the head-vertex of $M'_T = \varphi_c(M''_T)$ and is a proper descendant of u . Hence, the head-vertex v and the root-vertex of M'_T are distinct.

- *The tree T is a depth tree of M'_T .*

The external edges are created by applying the mapping φ_c , that is, by joining the head to another leg. By Lemma 4.6, any vertex incident to a leg is an ancestor of the head-vertex. Hence, any external edge joins comparable vertices. Moreover, by the preceding point, if the root is part of a complete edge, then this edge is external (internal edges are created by the mappings φ_a or φ_b which replace the head by a complete edge).

- *Let u_0 be the first vertex of M'_T incident to a leg on the T -path from the root-vertex to the head-vertex. Any bridge of M'_T is in the T -path between u_0 and the head-vertex.*

We proceed by induction on the length of w' . The property holds for the empty word. We write $w' = \alpha w''$ and suppose that it holds for $M''_T = \varphi_{w''}(M^0_\bullet)$. If $\alpha = a$ or b the property clearly holds for $M'_T = \varphi_\alpha(M''_T)$. We suppose now that $\alpha = c$. We denote by u_1 the first vertex of M''_T incident to a leg on the T -path from the root-vertex to the head-vertex. Let u and v be the vertices incident to the first and last legs of M''_T with the convention that u is an ancestor of v . By Lemma 4.6, the vertices u_1, u and v are all ancestors of the head-vertex v_1 of M''_T . Hence, u and v are on the T -path between u_1 and v_1 . This situation is represented in Fig. 16. By definition, the tree-growing map M'_T is obtained from M''_T by creating an edge e_1 between u and v_1 while v becomes the new head-vertex. We denote by P_1 (respectively P_2) the T -path between u_1 and u (respectively u and v_1). We consider a bridge e of M'_T . The edge e is a bridge of M''_T (since M'_T is obtained from M''_T by deleting an edge). By the induction hypothesis, the bridge e is either in P_1 or in P_2 . The

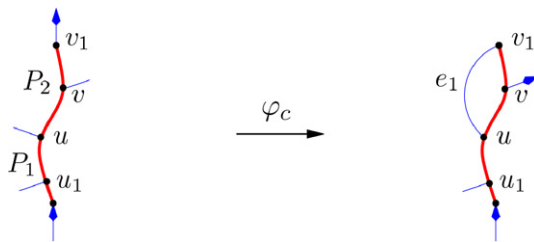


Fig. 16. Bridges are in the T -path between u_0 and the head-vertex.

edge e is not in the path P_2 since the new edge e_1 creates a cycle with P_2 . The bridge e is in P_1 , therefore the vertices u_1 and u are distinct. Hence $u_1 = u_0$ is the first vertex of M'_T incident to a leg on the T -path from the root-vertex to the head-vertex. Thus, the bridge e is in the T -path from u_0 to the head-vertex v of M'_T .

• The depth-map $\Phi(w)$ has no bridge.

By the preceding points, any bridge of $M_T = \varphi_w(M_\bullet^0)$ is on the T -path between the head-vertex and the endpoint of the only remaining leg. Hence, no bridge remains once the map closed. \square

We will now define a mapping Ψ (Definition 5.4) that we shall prove to be the inverse of Φ . The mapping Ψ destructs the tree-growing map that Φ constructs and recovers the walk. Looking at Fig. 10 from bottom-to-top and right-to-left we see how Ψ works.

We first define three mappings ψ_a, ψ_b, ψ_c on tree-growing maps that we shall prove to be the inverse of φ_a, φ_b and φ_c , respectively. We consider the following conditions for a tree-growing map M_T :

- (a) The head-vertex has degree 3 and is incident to an edge and a leg at the left of the head.
- (b) The head-vertex has degree 3 and is incident to an edge and a leg at the right of the head.
- (c) The head-vertex has degree 3 and is incident to 2 edges which are not bridges. Furthermore, the tree T is a depth tree.

The conditions (a)–(c) are the domain of definition of ψ_a, ψ_b, ψ_c , respectively. Before defining these mappings we need a technical lemma.

Lemma 5.2. *If condition (c) holds for the tree-growing map M_T , then there exists a unique external edge e_0 incident to the head-face with one endpoint u ancestor of the head-vertex and one endpoint v_0 descendant of the head-vertex.*

Lemma 5.2 is illustrated by Fig. 17.

Proof of Lemma 5.2. We suppose that M_T satisfies condition (c). One of the two edges incident to the head-vertex is in the T -path from the root-vertex to the head-vertex. Denote it e . The edge e separates the tree T in two subtrees T_1 and T_2 . We consider the set E_0 of external edges having one endpoint in T_1 and the other in T_2 . Any edge satisfying the conditions of Lemma 5.2 is in E_0 . Since e is not a bridge, the set E_0 is non-empty. Moreover, any edge in E_0 has one endpoint that is a descendant of the head-vertex. Since T is a depth tree, the other endpoint is an ancestor of the head-vertex. It remains to show that there is a unique edge e_0 in E_0 incident to the head-face. By

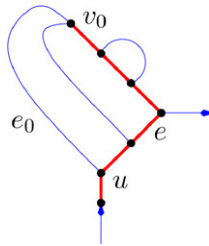


Fig. 17. The unique edge e_0 satisfying the conditions of Lemma 5.2.

contracting every edge in T_1 and T_2 we obtain a map with 2 vertices. The edges incident to both vertices are precisely the edges in $E_0 \cup \{e\}$. It is clear that exactly 2 of these edges are incident to the head-face. One is the internal edge e and the other is an external edge $e_0 \in E_0$. This edge e_0 is the only external edge satisfying the conditions of Lemma 5.2. \square

We are now ready to define the mappings ψ_a , ψ_b and ψ_c .

Definition 5.3. Let M_T be a tree-growing map.

- The tree-growing map $M'_{T'} = \psi_a(M_T)$ (respectively $\psi_b(M_T)$) is defined if condition (a) (respectively (b)) holds. In this case, the tree-growing map $M'_{T'}$ is obtained by suppressing the head-vertex v and the 3 incident half-edges. The other half of the edge incident to v becomes the new head.
- The tree-growing map $M'_{T'} = \psi_c(M_T)$ is defined if condition (c) holds. In this case, we consider the unique external edge e_0 with endpoints u, v_0 satisfying the conditions of Lemma 5.2. The edge e_0 is broken into two legs. The leg incident to v_0 becomes the new head (the former head becomes an anonymous leg).
- For a word $w = a_1 a_2 \dots a_n$ on the alphabet $\{a, b, c\}$, we denote by ψ_w the mapping $\psi_{a_n} \circ \psi_{a_{n-1}} \circ \dots \circ \psi_{a_1}$. Moreover, we say that the word w is *readable* on a tree-growing map M_T if the mapping ψ_w is well defined on M_T .

Remarks.

- Applying one of the mappings ψ_a , ψ_b or ψ_c to a 2-near-cubic map cannot delete the root (only half-edges incident to a vertex of degree 3 can disappear by application of ψ_a or ψ_b).
- The conditions (a), (b), (c) are incompatible. Thus, for any tree-growing map M_T , there is at most one readable word of a given length.
- Applying the mapping ψ_a , ψ_b or ψ_c decreases by one the number of edges. Therefore, the length of any readable word on a tree-growing map M_T is less than or equal to the number of edges in M_T .

We now define the mapping Ψ on bridgeless 2-near-cubic marked-depth-maps. Let M_T be such a map and let e be the marked (external) edge. Observe first that, unless M_T is reduced to a loop, the edge e has two distinct endpoints (or the endpoint of e would be incident to a bridge). We denote by u and v the endpoints of e with the convention that u is an ancestor of v . We *open* this map if we disconnect the edge e into two legs and choose the leg incident to v to be the head. We denote by $M_T^{-\dagger}$ the tree-growing map obtained by opening M_T . By convention, opening the

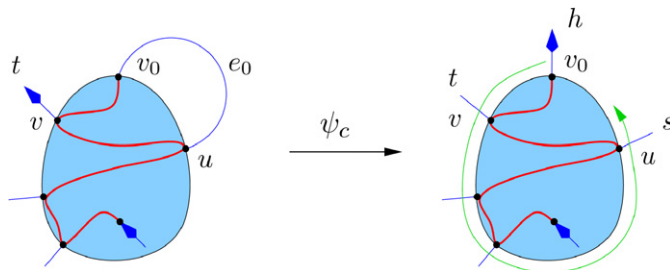


Fig. 18. The pair of first and last legs of the tree-growing map $\psi_c(M_T)$ is the pair $\{s, t\}$.

2-near-cubic marked-depth-map reduced to a loop gives M_{\bullet}^0 . Note that we obtain M_T by closing M_T^{pl} . We now define the mapping Ψ .

Definition 5.4. Let M_T be a bridgeless 2-near-cubic marked-depth-map. The word $\Psi(M_T)$ is the longest word readable on M_T^{pl} .

We want to prove that Φ and Ψ are inverse mappings. We begin by proving that the mapping ψ_α is the inverse of φ_α for $\alpha = a, b, c$.

We say that a tree-growing map satisfies condition (c') if it satisfies condition (c) and is such that the endpoint of every leg is an ancestor of the head-vertex.

Lemma 5.5.

- For $\alpha = a$ or b , the mapping $\psi_\alpha \circ \varphi_\alpha$ is the identity on all tree-growing maps and the mapping $\varphi_\alpha \circ \psi_\alpha$ is the identity on tree-growing maps satisfying condition (α) .
- The mapping $\psi_c \circ \varphi_c$ is the identity on tree-growing maps such that the endpoints of the first and last legs exist and are distinct ancestors of the head-vertex. The mapping $\varphi_c \circ \psi_c$ is the identity on tree-growing maps satisfying condition (c').

Before proving Lemma 5.5, we need the following technical result.

Lemma 5.6. Let M_T be a tree-growing map satisfying condition (c') and let e_0 be the edge with endpoints u, v_0 satisfying the conditions of Lemma 5.2. By definition, the tree-growing map $\psi_c(M_T)$ is obtained by breaking e_0 into two legs s and h incident to u and v_0 , respectively, while h becomes the new head. The pair of first and last legs of $\psi_c(M_T)$ is the pair $\{s, t\}$, where t is the head of M_T .

Lemma 5.6 is illustrated by Fig. 18.

Proof of Lemma 5.6.

- Let v be the head-vertex of M_T (i.e. the endpoint of t). By condition (c'), the endpoint of any leg of M_T is an ancestor of v . Therefore, in the tree-growing map $\psi_c(M_T)$, the vertex v is the last vertex incident to a leg on the T -path from the root-vertex to the head-vertex v_0 . Hence, by Lemma 4.5, the leg t is either the first or the last leg of $\psi_c(M_T)$.
- No leg lies between s and h on the tour of the head-face of $\psi_c(M_T)$ since this leg would have been inside a non-head face of M_T . Thus the leg s is either the first or the last leg of $\psi_c(M_T)$. \square

Proof of Lemma 5.5.

- For $\alpha = a$ or b , it is clear from the definitions that $\varphi_\alpha \circ \psi_\alpha$ is the identity mapping on all tree-growing maps and that $\varphi_\alpha \circ \psi_\alpha$ is the identity on tree-growing maps satisfying condition (α) .

- Consider a tree-growing map M_T such that the endpoints of the first and last legs exist and are distinct ancestors of the head-vertex v_0 . We call these legs s and t with the convention that the endpoint u of s is an ancestor of the endpoint v of t . By definition, $\varphi_c(M_T)$ is obtained by joining the head of M_T to s while t becomes the new head. Let e_0 be the external edge created by joining the head to s . The head-vertex v of the tree-growing map $\varphi_c(M_T)$ is on the cycle made of e_0 and the T -path between its two endpoints u and v_0 , thus $\varphi_c(M_T)$ satisfies condition (c) . Moreover, the external edge e_0 satisfies the conditions of Lemma 5.2. Thus, $\psi_c \circ \varphi_c(M_T) = M_T$.

- We consider a tree-growing map M_T satisfying condition (c') . We consider the edge e_0 with endpoints u, v_0 satisfying the conditions of Lemma 5.2. By definition, $\psi_c(M_T)$ is obtained by breaking e_0 into two legs s and h incident to u and v_0 , respectively, while h becomes the new head. By Lemma 5.6, the pair of first and last legs of $\psi_c(M_T)$ is $\{s, t\}$. Moreover, the endpoint u of s is an ancestor of the endpoint v of t (by definition of e_0, u, v_0 in Lemma 5.2). Therefore, the identity $\varphi_c \circ \psi_c(M_T) = M_T$ follows from the definitions. \square

Proposition 5.7. *The mapping $\Psi \circ \Phi$ is the identity on excursions.*

Proof.

- For any word w on the alphabet $\{a, b, c\}$ such that the tree-growing map $\varphi_w(M_\bullet^0)$ is well defined, the word w is readable on $\varphi_w(M_\bullet^0)$ and $\psi_w \circ \varphi_w(M_\bullet^0) = M_\bullet^0$.

We proceed by induction on the length of w . The property holds for the empty word. We write $w = \alpha w'$ with $\alpha = a, b$ or c and suppose that it holds for w' . Let $M'_T = \varphi_{w'}(M_\bullet^0)$. If $\alpha = c$, the endpoints of the first and last legs of M'_T are distinct and comparable (since φ_c is defined on M'_T). Moreover, we know by Lemma 4.6 that these endpoints are ancestors of the head-vertex. Thus, for $\alpha = a, b$ or c , Lemma 5.5 ensures that $\psi_\alpha \circ \varphi_\alpha(M'_T) = M'_T$. Therefore,

$$\psi_{\alpha w'} \circ \varphi_{\alpha w'}(M_\bullet^0) = \psi_{w'} \circ \psi_\alpha \circ \varphi_\alpha \circ \varphi_{w'}(M_\bullet^0) = \psi_{w'} \circ \psi_\alpha \circ \varphi_\alpha(M'_T) = \psi_{w'}(M'_T),$$

and $\psi_{w'}(M'_T) = M_\bullet^0$ by the induction hypothesis.

- For any excursion w , we have $\Psi \circ \Phi(w) = w$.

By definition, the map $M_T = \Phi(w)$ is obtained by closing $\varphi_w(M_\bullet^0)$. In order to conclude that $M_T^{+-} = \varphi_w(M_\bullet^0)$, we only need to check that the head of M_T^{+-} is the head of $\varphi_w(M_\bullet^0)$ (and the non-head leg of M_T^{+-} is the non-head leg of $\varphi_w(M_\bullet^0)$). This is true since the endpoint of the non-head leg of $\varphi_w(M_\bullet^0)$ is an ancestor of the head-vertex by Lemma 4.6. By the preceding point, the word w is readable on $M_T^{+-} = \varphi_w(M_\bullet^0)$ and $\psi_w(M_T) = \psi_w \circ \varphi_w(M_\bullet^0) = M_\bullet^0$. Since no letter is readable on M_\bullet^0 , the longest word readable on M_T^{+-} is w . Thus, $\Psi \circ \Phi(w) = \Psi(M_T) = w$. \square

It remains to show that $\Phi \circ \Psi$ is the identity mapping on bridgeless 2-near-cubic marked-depth-maps. We first prove that the image of a bridgeless 2-near-cubic marked-depth-map by Ψ is an excursion.

Proposition 5.8. *For any bridgeless 2-near-cubic marked-depth-map M_T , the longest word w readable on M_T^{+-} is an excursion. Moreover, the tree-growing map $\psi_w(M_T^{+-})$ is M_\bullet^0 .*

Proof. If M_T is the map reduced to a loop the result is trivial. We exclude this case in what follows. Let w be a word readable on M_T^{\pm} and let $N_T = \psi_w(M_T^{\pm})$. We denote by u_0 the first vertex of N_T incident to a leg on the T -path from the root-vertex to the head-vertex.

- Any bridge of N_T is in the T -path between u_0 and the head-vertex.

We proceed by induction on the length of w . Suppose first that w is the empty word. Let e_0 be the marked edge of M_T . By definition, the tree-growing map $N_T = M_T^{\pm}$ is obtained from M_T by breaking e_0 into two legs: the head and another leg incident to u_0 . Let e be a bridge of N_T and let N_1, N_2 be the two connected submaps obtained by deleting e . Since e is not a bridge of M_T , the edge e_0 joins N_1 and N_2 . Therefore, the root-vertex and head-vertex are not in the same submap. Thus, the bridge e is in any path between u_0 and the head-vertex, in particular, it is in the T -path.

We now write $w = \alpha w'$ with $\alpha = a, b$ or c and suppose, by the induction hypothesis, that the property holds for w' . We denote by u'_0 the first vertex of $N'_T = \psi_{w'}(M_T^{\pm})$ incident to a leg on the T -path from the root-vertex to the head-vertex. Suppose first that $\alpha = a$ or b . The edge incident to the head-vertex of N'_T is a bridge hence, by the induction hypothesis, it is in the T -path between u'_0 and the head-vertex v'_0 of N'_T . Hence, $u'_0 \neq v'_0$. Thus, $u_0 = u'_0$ and every bridge of $N_T = \psi_\alpha(N'_T)$ is in the T -path between u_0 and the head-vertex. Suppose now that $\alpha = c$. Since w is readable on M_T^{\pm} , the tree-growing map $N'_T = \psi_{w'}(M_T^{\pm})$ satisfies condition (c). We consider the edge e_0 with endpoints u, v_0 satisfying the conditions of Lemma 5.2. The map $N_T = \psi_c(N'_T)$ is obtained from N'_T by breaking e_0 into two legs. By definition, the head-vertex of N_T is v_0 . Moreover, the vertex u_0 is either u'_0 or u if u is an ancestor of u'_0 . We consider a bridge e of N_T . If e is a bridge of N'_T , it is in the T -path between u'_0 to the head-vertex of N'_T which is included in the T -path between u_0 and v_0 . If e is not a bridge of N'_T , we consider the two connected submaps N_1, N_2 obtained from N_T by deleting the bridge e . Since e is not a bridge of N'_T , the edge e_0 joins N_1 and N_2 . Hence, the endpoints u and v_0 of e_0 are not in the same submap. Thus, the bridge e is in every path of N_T between u and the head-vertex v_0 , in particular, it is in the T -path between u_0 and v_0 .

- The tree-growing map N_T has at least one leg beside the head.

We proceed by induction. The property holds for the empty word. We now write $w = \alpha w'$ with $\alpha = a, b$ or c and suppose that the property holds for w' . Suppose first that $\alpha = a$ or b . Since condition (α) holds, the edge incident to the head-vertex v'_0 of the tree-growing map $N'_T = \psi_{w'}(M_T^{\pm})$ is a bridge. By the preceding point, this edge is on the T -path between u'_0 and v'_0 , where u'_0 be the first vertex of N'_T incident to a leg on the T -path from the root-vertex to the head-vertex. Thus $u'_0 \neq v'_0$ and $N_T = \psi_\alpha(N'_T)$ has at least one leg (the one incident to u'_0) beside the head. In the case $\alpha = c$, the tree-growing map $N_T = \psi_c(N'_T)$ has one more legs than N'_T , hence it has at least one leg beside the head.

- The head and root of N_T are distinct half-edges.

By definition, the map M_T^{\pm} has one leg beside the head whose endpoint is a proper ancestor of the head-vertex. Hence, the head-vertex and root-vertex are distinct. We suppose now that $w = \alpha w'$ with $\alpha = a, b$ or c . If $\alpha = a$ or b the head of N_T is an half-edge of $N'_T = \psi_{w'}(M_T^{\pm})$ which is part of an internal edge. Hence it is not the root. If $\alpha = c$, the head of N_T is part of an external edge e of $N'_T = \psi_{w'}(M_T^{\pm})$. The edge is broken into the head of N_T and another leg whose endpoint is a proper ancestor of the head-vertex. Hence, the head-vertex and root-vertex of N_T are distinct.

- If w is the longest readable word, then $N_T = M_{\bullet}^0$.

We first prove that the root-vertex and the head-vertex of N_T are the same. Suppose they are distinct. In this case, the head-vertex has degree 3 and is incident to at least one edge. If it is incident to one edge, then one of the conditions (a) or (b) holds and w is not the longest readable word. Hence the head-vertex is incident to two edges e_1 and e_2 . One of these edges, say e_1 , is in the T -path from the root-vertex to the head-vertex and the other e_2 is not. By a preceding point, the edge e_2 is not a bridge. Therefore, e_1 is not a bridge either (e_1 and e_2 have the same ability to disconnect the map). In this case, condition (c) holds (since T is a depth tree) and w is not the longest readable word. Thus, the root-vertex and the head-vertex of N_T are the same. Therefore, the root-vertex has degree 2 and is incident to the head and the root. The head and the root are distinct (by the preceding point). Moreover, the root is a leg. Indeed, if the root was not a leg it would be part of an external edge which is a bridge (which is impossible since the tree T is spanning). Hence the root-vertex is incident to two legs: the root and the head. Thus, $N_T = M_{\bullet}^0$.

- The tree-growing map N_T has $2|w|_c - |w|_a - |w|_b + 1$ legs beside the head.

The tree-growing map $M_T^{\pm\pm}$ has one leg beside the head. Moreover, applying mapping ψ_a or ψ_b decreases by one the number of legs whereas applying mapping ψ_c increases this number by two. Hence the property follows easily by induction.

- The longest word w readable on $M_T^{\pm\pm}$ is an excursion.

By the preceding points, any prefix w' of w satisfies $2|w'|_c - |w'|_a - |w'|_b + 1 \geq 1$ (since this quantity is the number of non-head legs of $\psi_{w'}(M_T^{\pm\pm})$). Moreover, since $\psi_w(M_T^{\pm\pm}) = M_{\bullet}^0$ has one leg beside the root, we have $2|w|_c - |w|_a - |w|_b + 1 = 1$. These properties are equivalent to (5) and (6), hence w is an excursion. \square

Proposition 5.9. *The mapping $\Phi \circ \Psi$ is the identity on bridgeless 2-near-cubic marked-depth-maps.*

Proof. Let M_T be a bridgeless 2-near-cubic marked-depth-map.

- For any word w readable on $M_T^{\pm\pm}$, the endpoints of any leg of $\psi_w(M_T^{\pm\pm})$ is an ancestor of the head-vertex.

We proceed by induction on the length of w . The property holds for the empty word. We now write $w = \alpha w'$ with $\alpha = a, b$ or c and suppose that it holds for w' . For $\alpha = a$ or b , the property clearly holds for w . Suppose now that $\alpha = c$. Since w is readable, the tree-growing map $N'_T = \psi_{w'}(M_T^{\pm\pm})$ satisfies condition (c). We consider the edge e_0 with endpoints u, v_0 satisfying the conditions of Lemma 5.2. By definition, the head-vertex v_0 of $N_T = \psi_c(N'_T)$ is a descendant of the head-vertex v of N'_T . By the induction hypothesis, the endpoint of any leg of N'_T is an ancestor of v . Hence, the endpoint of any leg of N_T is an ancestor of the head-vertex v_0 .

- For any word w readable on $M_T^{\pm\pm}$, we have $\varphi_w \circ \psi_w(M_T^{\pm\pm}) = M_T^{\pm\pm}$.

We proceed by induction. The property holds for the empty word. We now write $w = \alpha w'$ with $\alpha = a, b$ or c and suppose that the property holds for w' . If $\alpha = a$ or b the induction step is given directly by Lemma 5.5 (since condition (α) holds for $M'_T = \psi_{w'}(M_T^{\pm\pm})$). If $\alpha = c$, that is, condition (c) holds for $M'_T = \psi_{w'}(M_T^{\pm\pm})$, we must prove that condition (c') holds (in order to apply Lemma 5.5). But we are ensured that condition (c') holds by the preceding point. Thus, for $\alpha = a, b$ or c , Lemma 5.5 ensures that $\varphi_{\alpha} \circ \psi_{\alpha}(M'_T) = M'_T$. Therefore,

$$\varphi_{\alpha w'} \circ \psi_{\alpha w'}(M_T^{\pm\pm}) = \varphi_{w'} \circ \varphi_{\alpha} \circ \psi_{\alpha} \circ \psi_{w'}(M_T^{\pm\pm}) = \varphi_{w'} \circ \varphi_{\alpha} \circ \psi_{\alpha}(M'_T) = \varphi_{w'}(M'_T),$$

and $\varphi_{w'}(M'_T) = M_T^{\pm\pm}$ by the induction hypothesis.

• $\Phi \circ \Psi(M_T) = M_T$.

By definition, the word $w = \Psi(M_T)$ is the longest readable word on $M_T^{-\perp}$. Hence, by Proposition 5.8, $\psi_w(M_T^{-\perp}) = M_\bullet^0$. By the preceding point, $\varphi_w(M_\bullet^0) = \varphi_w \circ \psi_w(M_T^{-\perp}) = M_T^{-\perp}$. By definition, the map $\Phi(w)$ is obtained by closing $\varphi_w(M_\bullet^0) = M_T^{-\perp}$, hence $\Phi(w) = M_T$. Thus, $\Phi \circ \Psi(M_T) = \Phi(w) = M_T$. \square

By Proposition 5.1, the mapping Φ associates a bridgeless 2-near-cubic marked-depth-map with any excursion. Conversely, by Proposition 5.8, the mapping Ψ associates an excursion with any bridgeless 2-near-cubic marked-depth-map. The mappings Φ and Ψ are inverse mappings by Propositions 5.7 and 5.9. Thus, the mapping Φ is a bijection between excursions and bridgeless 2-near-cubic marked-depth-maps. Moreover, if an excursion w has size n (length $3n$), the 2-near-cubic depth-map $\Phi(w)$ has size n ($3n + 1$ edges). This concludes the proof of Theorem 4.7.

6. A bijection between Kreweras walks and cubic depth-maps

In this section, we prove that the mapping Φ establishes a bijection between Kreweras walks ending at the origin and 2-near-cubic depth-maps. This result is stated more precisely in the following theorem.

Theorem 6.1. *Let w be an excursion. The marked edge of the 2-near-cubic depth-map $\Phi(w)$ is the root-edge if and only if the excursion w is a Kreweras walk ending at the origin.*

Thus, the mapping Φ induces a bijection between Kreweras walks of size n (length $3n$) ending at the origin and bridgeless 2-near-cubic depth-maps of size n ($3n + 1$ edges).

Figure 19 illustrates an instance of Theorem 6.1. Before proving this theorem we explore its enumerative consequences. From Theorem 6.1, the number k_n of Kreweras walks of size n is equal to the number d_n of bridgeless 2-near-cubic depth-maps of size n . The number d_n is given by Corollary 4.8. We obtain the following result.

Theorem 6.2. *There are $k_n = \frac{4^n}{(n+1)(2n+1)} \binom{3n}{n}$ Kreweras walks of size n (length $3n$) ending at the origin.*

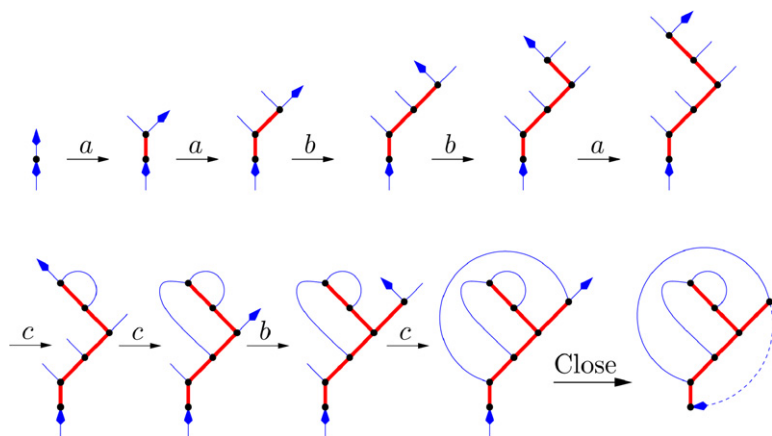


Fig. 19. The image of a Kreweras walk by Φ : the root-edge is marked.

The rest of this section is devoted to the proof of Theorem 6.1.

Consider a growing map M such that the root is a leg. Recall that *making the tour of the head-face* means following its border in counterclockwise direction starting from the head (see Fig. 12). We call *left* (respectively *right*) the legs encountered before (respectively after) the root during the tour of the head-face. For instance, the growing map in Fig. 12 has one left leg and two right legs.

Lemma 6.3. *For any Kreweras walk w ending at the origin, the marked edge of $\Phi(w)$ is the root-edge.*

Proof. Let w' be a suffix of w and let $M'_T = \varphi_{w'}(M_\bullet^0)$ be the corresponding tree-growing map.

- *The root of M'_T is a leg and M'_T has $|w'|_a - |w'|_c$ left legs and $|w'|_b - |w'|_c$ right legs.* (Observe that, these quantities are non-negative by (3) and (4).)

We proceed by induction on the length of w' . The property holds for the empty word. We now write $w' = \alpha w''$ with $\alpha = a, b$ or c and suppose that the property holds for w'' . If $\alpha = a$ or b the property holds for w' since applying φ_a (respectively φ_b) increases by one the number of left (respectively right) legs. We now suppose that $\alpha = c$. We know that $|w''|_a - |w''|_c = |w'|_a - |w'|_c + 1 \geq 1$. Hence, by the induction hypothesis, the tree-growing-map $M''_T = \varphi_{w''}(M_\bullet^0)$ has at least one left leg. Similarly, M''_T has at least one right leg. Therefore, the first (respectively last) leg of M''_T is a left (respectively right) leg. Hence, applying φ_c to M''_T decreases by one the number of left (respectively right) legs. Thus, the property holds for w' .

- For $w' = w$, the preceding point shows that $\varphi_w(M_\bullet^0)$ has only one leg beside the head and that this leg is the root. Thus, the marked edge of $\Phi(w)$ is the root-edge. \square

Lemma 6.4. *For any bridgeless 2-near-cubic depth-map M_T marked on the root-edge, the word $w = \Psi(M_T) = \Phi^{-1}(M_T)$ is a Kreweras walk ending at the origin.*

Proof. Let w be a word readable on $M_T^{\text{+}}$ and let $N_T = \psi_w(M_T^{\text{+}})$. Observe that the root of N_T is a leg (since it is the case in $M_T^{\text{+}}$ and the root never disappears).

- *The tree-growing map N_T has $|w|_c - |w|_a$ left legs and $|w|_c - |w|_b$ right legs.*

We proceed by induction on the length of w . The property holds for the empty word. We now write $w = \alpha w'$ with $\alpha = a, b$ or c and suppose that the property holds for w' . If $\alpha = a$ or b the property holds for w since applying ψ_a (respectively ψ_b) decreases by one the number of left (respectively right) legs. We now suppose that $\alpha = c$. The map $N'_T = \psi_{w'}(M_T^{\text{+}})$ satisfies condition (c). We have already proved (see the first point in the proof of Proposition 5.8) that the endpoint of every leg is an ancestor of the head-vertex. Hence N'_T satisfies condition (c'). Therefore, Lemma 5.6 holds for N'_T . We adopt the notations h, s, t of this lemma which is illustrated in Fig. 18. By Lemma 5.6, the pair of first and last head of $N_T = \psi_c(N'_T)$ is the pair $\{s, t\}$. Hence, in the pair $\{s, t\}$ one is a left leg and the other is a right leg of N_T . Moreover, the other left and right legs of N_T are the same as in N'_T . Thus, applying ψ_c to N'_T increases by one the number of left (respectively right) legs. Hence, the property holds for w .

- *The word $w = \Psi(M_T)$ is a Kreweras walk ending at the origin.*

By definition, w is the longest word readable on $M_T^{\text{+}}$. By Proposition 5.8, $\psi_w(M_T^{\text{+}}) = M_\bullet^0$. By

the preceding point, we get $|w|_c - |w|_a = 0$ and $|w|_c - |w|_b = 0$ (since M_\bullet^0 has no left nor right leg). Moreover, for any suffix w' of w , the preceding point proves that $|w'|_c - |w'|_a \geq 0$ and $|w'|_c - |w'|_b \geq 0$. These properties are equivalent to (3) and (4), hence w is a Kreweras walk ending at the origin. \square

7. Enumerating depth trees and cubic maps

In Section 4, we exhibited a bijection Φ between excursions and bridgeless 2-near-cubic marked-depth-maps. As a corollary we obtained the number of bridgeless 2-near-cubic depth-maps of size n :

$$d_n = \frac{4^n}{(n + 1)(2n + 1)} \binom{3n}{n}.$$

In this section, we prove that any bridgeless 2-near-cubic map of size n has 2^n depth trees (Corollary 7.5). Hence, the number of bridgeless 2-near-cubic maps of size n is

$$c_n = \frac{d_n}{2^n} = \frac{2^n}{(n + 1)(2n + 1)} \binom{3n}{n}.$$

Given the bijection between 2-near-cubic maps and cubic maps (see Fig. 5), we obtain the following theorem.

Theorem 7.1. *There are*

$$c_n = \frac{2^n}{(n + 1)(2n + 1)} \binom{3n}{n}$$

bridgeless cubic maps with $3n$ edges.

By duality, c_n is also the number of loopless triangulations with $3n$ edges. Hence, we recover Eq. (2) announced in the introduction. As mentioned above, an alternative bijective proof of Theorem 7.1 was given in [7].

The rest of this section is devoted to the counting of depth trees on cubic maps and, more generally, on cubic (potentially non-planar) graphs. We first give an alternative characterization of depth trees. This characterization is based on the depth-first search (DFS) algorithm (see Section 23.3 of [2]). We consider the DFS algorithm as an algorithm for constructing a spanning tree of a graph.

Consider a graph G with a distinguished vertex v_0 . If the DFS algorithm starts at v_0 , the subgraph T (see below) constructed by the algorithm remains a tree containing v_0 . We call *visited* the vertices in T and *unvisited* the other vertices. The distinguished vertex v_0 is considered as the root-vertex of the tree. Hence, any vertex in T distinct from v_0 has a *father* in T .

Definition 7.2. Depth-first search (DFS) algorithm.

Initialization: Set the *current vertex* to be v_0 and the tree T to be reduced to v_0 .

Core: While the current vertex v is adjacent to some unvisited vertices or is distinct from v_0 do: If there are some edges linking the current vertex v to an unvisited vertex, then choose one of them. Add the chosen edge e and its unvisited endpoint v' to the tree T . Set the current vertex to be v' .

Else, backtrack, that is, set the current vertex to be the father of v in T .

End: Return the tree T .

It is well known that the DFS algorithm returns a spanning tree. It is also known [2] that the two following properties are equivalent for a spanning tree T of a graph G having a distinguished vertex v_0 :

- (i) Any external edge joins comparable vertices.
- (ii) The tree T can be obtained by a DFS algorithm on the graph G starting at v_0 .

Before stating the main result of this section, we need an easy preliminary lemma.

Lemma 7.3. *Let G be a connected graph with a distinguished vertex v_0 whose deletion does not disconnect the graph. Then, any spanning tree T of G satisfying conditions (i)–(ii) has exactly one edge incident to v_0 .*

Proof. Let e_0 be an edge of T incident to v_0 and let v_1 be the other endpoint of e_0 . We partition the vertex set V of G into $\{v_0\} \cup V_0 \cup V_1$, where V_1 is the set of descendants of v_1 . There is no internal edge joining a vertex in V_0 and a vertex in V_1 . There is no external edge either or it would join two non-comparable vertices. Thus $V_0 = \emptyset$ or the deletion of v_0 would disconnect the graph. \square

Theorem 7.4. *Let G be a loopless connected graph with a distinguished vertex v_0 whose deletion does not disconnect the graph. Let e_0 be an edge incident to v_0 . If G is a k -near-cubic graph (v_0 has degree k and the other vertices have degree 3) of size n ($3n + 2k - 3$ edges), then there are 2^n trees containing e_0 and satisfying conditions (i)–(ii).*

Given that the depth trees are the spanning trees satisfying conditions (i)–(ii) and not containing the root, the following corollary is immediate.

Corollary 7.5. *Any bridgeless 2-near-cubic map of size n ($3n + 1$ edges) has 2^n depth trees.*

Remark. Theorem 7.4 implies that any k -near-cubic loopless graph of size n has $k2^n$ trees satisfying the conditions (i)–(ii).

The rest of this section is devoted to the proof of Theorem 7.4. The proof relies on the intuition that exactly n real binary choices have to be made during the execution of a DFS algorithm on a k -near-cubic map of size n .

Given a graph G and a subset of vertices U , we say that two vertices u and v are U -connected if there is a path between u and v containing only vertices in $U \cup \{u, v\}$.

Lemma 7.6. *Let v be the current vertex and let U be the set of unvisited vertices at a given time of the DFS algorithm. The vertices that will be visited before the last visit to v are the vertices in U that are U -connected to v .*

Proof. Let S be the set of vertices in U that are U -connected to v . We make an induction on the cardinality of S . If the set S is empty, there is no edge linking v to an unvisited vertex. Hence, the next step in the algorithm is to backtrack and the vertex v will never be visited again. In other words, it is the last visit to v , hence the property holds. Suppose now that S is non-empty. In this case, there are some edges linking the current vertex v to an unvisited vertex. Let e be the

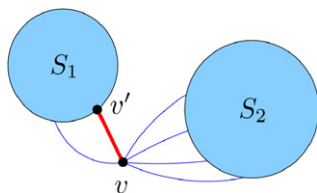


Fig. 20. Partition of the vertices in S .

edge chosen by the DFS algorithm and let $v' \in U$ be the corresponding endpoint. Let S_1 be the set of vertices in U that are U -connected to v' and let $S_2 = S \setminus S_1$. Observe that no edge joins a vertex in S_1 and a vertex in S_2 . This situation is represented in Fig. 20. The set of vertices in $U' = U \setminus \{v'\}$ that are U' -connected to v is $S'_1 = S_1 \setminus \{v'\}$ (since a vertex is U -connected to v' if and only if it is U' -connected to v'). By the induction hypothesis, S'_1 is the set of vertices visited between the first and last visit to v' . Hence S_1 is the set of vertices visited before the algorithm returns to v . Since no edge joins a vertex in S_1 and a vertex in S_2 , the vertices in S_2 are the vertices in $U \setminus S_1$ that are $(U \setminus S_1)$ -connected to v . By the induction hypothesis, S_2 is the set of vertices visited before the last visit to v . Thus, the property holds. \square

Proof of Theorem 7.4. Clearly, the spanning trees containing e_0 and satisfying the conditions (i)–(ii) are the spanning trees obtained by a DFS algorithm for which the first *core step* is to choose e_0 . We want to prove that there are 2^n such spanning trees.

We consider an execution of the DFS algorithm for which the first *core step* is to choose e_0 and denote by \mathcal{T} the spanning tree returned by the DFS algorithm (in order to distinguish it from the evolving tree T). After the first *core step*, the tree T is reduced to e_0 and its two endpoints v_0 and v'_0 . Let V be the vertex set of G and let $V' = V \setminus \{v_0, v'_0\}$. Since the deletion of v_0 does not disconnect the graph, every vertex in V' is V' -connected to v'_0 . Hence, by Lemma 7.6, every vertex will be visited before the algorithm returns to v_0 . Thus, from this stage on, the current vertex v is incident to 3 edges e, e_1, e_2 , where $e \in \mathcal{T}$ links v to its father.

- We denote by v_1 and v_2 the endpoints of e_1 and e_2 , respectively (these endpoints are not necessarily distinct), and we denote by U the set of unvisited vertices. We distinguish three cases:

- (α) at least one of the vertices v_1, v_2 is not in U ,
- (β) the two vertices v_1, v_2 are in U and are U -connected with each other,
- (γ) the two vertices v_1, v_2 are in U and are not U -connected with each other.

The three cases are illustrated by Fig. 21. We prove successively the following properties:

- In case (α), no choice has to be done by the algorithm.

Indeed, there is at most one edge (e_1 or e_2) linking the current vertex v to an unvisited vertex.

- In case (β), the algorithm has to choose between e_1 and e_2 . This choice necessarily leads to two different spanning trees \mathcal{T} . Indeed the edge e_1 (respectively e_2) is in \mathcal{T} if and only if the choice of e_1 (respectively e_2) is made.

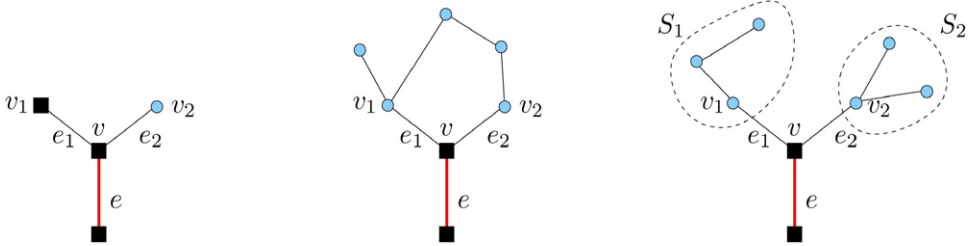


Fig. 21. Case (α) (left), case (β) (middle) and case (γ) (right). The visited vertices are indicated by a square while unvisited ones are indicated by a circle.

Suppose (without loss of generality), that the choice of e_1 is made. The vertex v_2 is $(U \cup \{v_1\})$ -connected to v_1 (a vertex is $(U \cup \{v_1\})$ -connected to v_1 if and only if it is U -connected to v_1). Hence, by Lemma 7.6, the vertex v_2 will be visited before the last visit to v_1 , that is, before the algorithm returns to the vertex v . Therefore, the edge e_2 will not be in the spanning tree \mathcal{T} .

– In case (γ) , the algorithm has to choose between e_1 and e_2 . Moreover, any tree \mathcal{T} obtained by choosing e_1 can be also obtained by choosing e_2 .

Let S_1 and S_2 be the set of vertices in U that are U -connected to v_1 and v_2 , respectively. Observe that the sets S_1 and S_2 are disjoint and no edge links a vertex in S_1 and a vertex in S_2 (otherwise the vertices v_1 and v_2 would be U -connected). Suppose that the choice of e_1 is made. The set of vertices in $U \setminus \{v_1\}$ that are $(U \setminus \{v_1\})$ -connected to v_1 is $S_1 \setminus \{v_1\}$. Hence, by Lemma 7.6, the set of vertices visited before the last visit to v_1 , that is, before the algorithm returns to v is S_1 . Since v_2 is not in S_1 the next step of the algorithm is to choose e_2 . Let $U_2 = U \setminus S_1$ be the set of unvisited vertices at this stage. Since no vertex in S_1 is adjacent to a vertex in S_2 , the set of vertices in $U_2 \setminus \{v_2\}$ that are $(U_2 \setminus \{v_2\})$ -connected to v_2 is $S_2 \setminus \{v_2\}$. Hence, by Lemma 7.6, the set of vertices visited before the last visit to v_2 , that is, before the algorithm returns to v is S_2 . Let T_1 (respectively T_2) be the subtree constructed by the algorithm between the first and last visit to v_1 (respectively v_2). Since no vertex in S_1 is adjacent to a vertex in S_2 , the subtree T_1 could have been constructed exactly the same way if the algorithm had chosen e_2 (instead of e_1) at the beginning. Similarly, the subtree T_2 could have been constructed exactly in the same way if the algorithm had chosen e_2 at the beginning. Therefore, the tree \mathcal{T} returned by the algorithm could have been constructed if the algorithm had chosen e_2 (instead of e_1) at the beginning.

- During any execution of the DFS algorithm we are exactly n times in case (β) .

The k -near-cubic graph G has $3n + 2k - 3$ edges and $2n + 2k - 1$ vertices. Hence, the spanning tree \mathcal{T} has $2n + 2k - 2$ edges. Thus, there are $n + k - 1$ external edges among which $k - 1$ are incident to v_0 . Let E_β be the set of the n external edges not incident to v_0 . Since G is loopless and the spanning tree \mathcal{T} satisfies (i)–(ii), the edges in E_β have distinct and comparable endpoints. For any edge e in E_β , we denote by v_e the endpoint of e which is the ancestor of the other endpoint. The vertex v_e is incident to e , to the edge of \mathcal{T} linking v to its father and to another edge in \mathcal{T} linking v_e to its son (otherwise v_e has no descendant). In particular, if e and e' are distinct edges in E_β , then the vertices v_e and $v_{e'}$ are distinct. Thus, the set of vertices $V_\beta = \{v_e/e \in E_\beta\}$ has size n .

We want to prove that the case (β) occurs when the algorithm visit a vertex in V_β for the first time (and not otherwise). Let v be a vertex in V_β . The vertex v is incident to an edge e_1 in E_β ,

an edge e in \mathcal{T} linking v to its father and another edge e_2 in \mathcal{T} linking v to its son. Let T be the tree constructed by the algorithm at the time of the first visit to v and let U be the set of unvisited vertices. Any descendant of v is in U . In particular, the endpoints v_1 and v_2 of e_1 and e_2 are in U and are U -connected with each other (take the \mathcal{T} -path between v_1 and v_2). Thus, we are in case (β) . Conversely, if we are in case (β) during the algorithm, the current vertex v is visited for the first time (or one of the vertices v_1, v_2 would already be in U). Moreover, by the preceding point, one of the edges (e_1 or e_2) incident to v is not in \mathcal{T} and joins v to one of its descendants. Hence, the current vertex v is in V_β .

- During the DFS algorithm we have to make n binary choices that will affect the outcome of the algorithm (case (β)). The other choices (case (γ)) do not affect the outcome of the algorithm. Therefore, there are 2^n possible outcomes. \square

8. Applications, extensions and open problems

8.1. Random generation of triangulations

The random generation of excursions of length $3n$ (with uniform distribution) reduces to the random generation of 1-dimensional walks of length $3n$ with steps $+2, -1$ starting and ending at 0 and remaining non-negative. The random generation of these walks is known to be feasible in linear time. (One just needs to generate a word of length $3n + 1$ containing n letters c and $2n + 1$ letters α and to apply the cycle lemma.) Given an excursion w , the construction of the 2-near-cubic marked-depth-map $\Phi(w)$ can be performed in linear time. Therefore, we have a linear time algorithm for the random generation (with uniform distribution) of bridgeless 2-near-cubic marked-depth-maps. For any bridgeless 2-near-cubic map there are 2^n depth trees and $(n + 1)$ possible marking. Therefore, if we drop the marking and the depth tree at the end of the process, we obtain a uniform distribution on bridgeless 2-near-cubic maps. This allows us to generate uniformly bridgeless cubic maps or, dually, loopless triangulations, in linear time.

8.2. Kreweras walks ending at $(i, 0)$ and $(i + 2)$ -near-cubic maps

The Kreweras walks ending at $(i, 0)$ are the words w on the alphabet $\{a, b, c\}$ with $|w|_a + i = |w|_b = |w|_c$ such that any suffix w' of w satisfies $|w'|_a + i \geq |w'|_c$ and $|w'|_b \geq |w'|_c$. There is a very nice formula [4] giving the number of Kreweras walks of size n (length $3n + 2i$) ending at $(i, 0)$:

$$k_{n,i} = \frac{4^n(2i + 1)}{(n + i + 1)(2n + 2i + 1)} \binom{2i}{i} \binom{3n + 2i}{n}. \tag{8}$$

There is also a similar formula [5] for non-separable $(i + 2)$ -near-cubic maps of size n ($3n + 2i + 1$ edges):

$$c_{n,i} = \frac{2^n(2i + 1)}{(n + i + 1)(2n + 2i + 1)} \binom{2i}{i} \binom{3n + 2i}{n}. \tag{9}$$

In this subsection, we show that the bijection Φ (Definition 4.2) can be extended to Kreweras walks ending at $(i, 0)$. This gives a bijective correspondence explaining why $k_{n,i} = 2^n c_{n,i}$.

Consider the tree-growing map M_\bullet^i reduced to a vertex, a root, a head and i left legs (Fig. 22). We define the image of a Kreweras walk w ending at $(i, 0)$ as the map obtained by closing $\varphi_w(M_\bullet^i)$. We get the following extension of Theorem 6.1.

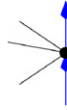


Fig. 22. The tree-growing map M_i^* when $i = 3$.

Theorem 8.1. *The mapping Φ is a bijection between Kreweras walks of size n (length $3n + 2i$) ending at $(i, 0)$ and non-separable $(i + 2)$ -near-cubic maps of size n ($3n + 2i + 1$ edges) marked on the root-edge with a depth tree that contains the edge following the root in counterclockwise order around the root-vertex.*

By Theorem 7.4, there are 2^n such depth trees. Consequently, we obtain the following corollary:

Corollary 8.2. *The number $k_{n,i}$ of Kreweras walks of size n ending at $(i, 0)$ and the number $c_{n,i}$ of non-separable $(i + 2)$ -near-cubic maps of size n are related by the equation $k_{n,i} = 2^n c_{n,i}$.*

One can define the counterpart of excursions for Kreweras walks ending at $(i, 0)$. These are the walks obtained when one chooses an external edge in a non-separable $(i + 2)$ -near-cubic depth-map such that the edge following the root is in the tree and applies the mapping $\Psi = \Phi^{-1}$. Alas, we have found no simple characterization of this set of walks nor any bijective proof explaining why this set has cardinality

$$\frac{4^n (2i + 1)}{(2n + 2i + 1)} \binom{2i}{i} \binom{3n + 2i}{n}.$$

Acknowledgment

I thank Mireille Bousquet-Mélou for steady encouragements and support.

References

- [1] M. Bousquet-Mélou, Walks in the quarter plane: Kreweras' algebraic model, *Ann. Appl. Probab.* 15 (2) (2005) 1451–1491.
- [2] T.H. Cormen, C.E. Leiserson, R.L. Rivest, *Introduction to Algorithms*, first ed., MIT Press, 1990.
- [3] I.M. Gessel, A probabilistic method for lattice path enumeration, *J. Statist. Plann. Inference* 14 (1986) 49–58.
- [4] G. Kreweras, Sur une classe de problèmes liés au treillis des partitions d'entiers, *Cahier du B.U.R.O.* 6 (1965) 5–105.
- [5] R.C. Mullin, On counting rooted triangular maps, *Canad. J. Math.* 17 (1965) 373–382.
- [6] H. Niederhausen, The ballot problem with three candidates, *European J. Combin.* 4 (2) (1983) 175–188.
- [7] D. Poulalhon, G. Schaeffer, A bijection for triangulations of a polygon with interior points and multiple edges, *Theoret. Comput. Sci.* 307 (2) (2003) 385–401.
- [8] G. Schaeffer, *Conjugaison d'arbres et cartes combinatoires aléatoires*, PhD thesis, Université Bordeaux I, 1998.
- [9] R.P. Stanley, *Enumerative Combinatorics*, vol. 2, Wadsworth & Brooks/Cole, 1999.
- [10] R.P. Stanley, *Clay Research Academy problems*, 2005.