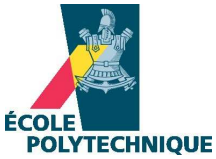
Combinatorics of planar maps and algorithmic applications.

Fusy Eric



LIX, École Polytechnique
et
Projet ALGO, INRIA Rocquencourt

Thèse de doctorat

Spécialité : Informatique

présentée par

Éric Fusy

pour obtenir le grade de Docteur de l'École Polytechnique

Sujet :

Combinatoire des cartes planaires et applications algorithmiques

Soutenue le lundi 11 juin 2007 devant le jury composé de:

Mireille BOUSQUET-MÉLOU	Rapporteur
Stefan FELSNER.....	Rapporteur
Philippe FLAJOLET	Directeur de thèse
Hubert de FRAYSSEIX	Examinateur
Marc NOY.....	Examinateur
Gilles SCHAEFFER.....	Directeur de thèse

Contents

Remerciements.

Je tiens à exprimer toute ma gratitude à mes deux encadrants de thèse Gilles Schaeffer et Philippe Flajolet. Dès le stage de Masters, Gilles m'a guidé et encouragé à travailler sur des sujets variés et enthousiasmants. Sa grande sérénité en recherche et en général rend le travail à ses côtés très agréable, presque comme un jeu. Je le remercie beaucoup également pour sa gentillesse et sa bienveillance, et pour m'avoir mis en relation avec Philippe alors que je cherchais un endroit où faire ma thèse. J'ai ainsi pu préparer ma thèse au projet ALGO à l'INRIA dans d'excellentes conditions, sous la co-direction de Gilles et Philippe. J'ai beaucoup apprécié l'encadrement chaleureux de Philippe, sa disponibilité, et la liberté qu'il m'a donnée. Il m'a aussi énormément appris lors de nos multiples discussions, me permettant en particulier de progresser dans l'approche des problèmes et la présentation des résultats. De manière plus groupée, je suis très reconnaissant à Gilles et Philippe pour l'attitude très positive qu'ils communiquent: ils m'ont présenté de nombreux chercheurs et étudiants, et m'ont toujours encouragé à aller vers les autres, à voyager, à collaborer, à trouver du plaisir dans les échanges d'idées.

Je remercie vivement Mireille Bousquet-Mélou et Stefan Felsner pour avoir accepté d'être rapporteurs et pour leurs remarques et conseils précieux. Je suis aussi très heureux que Marc Noy et Hubert de Fraysseix aient accepté d'être membres du jury.

J'ai eu la chance de passer mes trois années de thèse au projet ALGO où j'ai beaucoup apprécié la grande liberté offerte pour la recherche, dans une ambiance stimulante et chaleureuse. Je remercie pour cela tous les membres du projet, en particulier Bruno Salvy pour m'y avoir accueilli et Viriginie Collette pour sa disponibilité et sa gentillesse. J'ai aussi bénéficié au LIX à Polytechnique de très bonnes conditions de travail et d'un cadre sympathique, je pense en particulier à mon ami Luca Castelli Aleardi, et je remercie Nicole Dubois pour sa gentillesse et son aide à préparer la soutenance.

Un des grands plaisirs de ma thèse a été de pouvoir beaucoup voyager. Je tiens à remercier les chercheurs à qui j'ai rendu visite pour leur accueil très chaleureux: Stefan Felsner, Florian Zickfeld, Mihyun Kang et Manuel Bodirsky à Berlin, Omer Giménez et Marc Noy à Barcelone, Andreas Weissl et Stefanie Gerke à Zurich, et Bilyana Shoilekova à Oxford.

La thèse a été l'occasion pour moi de sympathiser et échanger des idées avec de très nombreux étudiants et chercheurs, notamment dans la communauté ALEA et la communauté Graph Drawing. Je pense en particulier à Frédéric Giroire, Frédéric Meunier, Olivier Gandouet, Carine Pivoteau, Olivier Bodini, Nicolas Pouyanne, Dominique Poulalhon, Manuel Bodirsky, Mihyun Kang, et Stefan Vigerske, avec qui j'ai eu la chance de collaborer, ainsi qu'à Olivier Bernardi, Nicolas Bonichon et Guillaume Chapuy avec qui j'ai eu beaucoup de plaisir à discuter sur les bijections et les orientations.

Enfin, j'ai une grande pensée pour mes parents et mon frère Jean-Yves, je leur témoigne ici toute mon affection.

Résumé. Cette thèse traite de l'algorithmique des cartes planaires (graphes dessinés dans le plan sans intersection d'arêtes) et propose des procédures efficaces pour le codage, la génération aléatoire, et le dessin de plusieurs familles importantes: 3-connexes, triangulations, quadrangulations... En particulier, on décrit le premier algorithme optimal de codage des incidences faces-arêtes-sommets des maillages polygonaux de topologie sphérique, qui atteint la borne inférieure de 2bits par arête. En partant d'un générateur de cartes 3-connexes, on développe un nouveau générateur aléatoire uniforme de graphes planaires dont la complexité est la meilleure connue actuellement: quadratique (en espérance) en taille exacte et linéaire (en espérance) en taille approchée. Enfin, on donne plusieurs algorithmes de dessin en lignes droites (arêtes représentées par des segments) de cartes planaires sur la grille. Les procédures de dessin sont à la fois très simples à décrire et donnent les meilleures performance (en probabilité) pour le dessin de deux familles de cartes: les triangulations du carré sans 3-cycle rempli —dites irréductibles— et les quadrangulations. Pour développer les algorithmes présentés dans la thèse, on exploite plusieurs structures combinatoires sur les cartes (orientations spécifiques, partitions en arbres couvrants...) ainsi que de nouvelles constructions bijectives.

Abstract. This thesis describes algorithms on planar maps (graphs embedded in the plane without edge-crossings) based on their combinatorial properties. For several important families of planar maps (3-connected, triangulations, quadrangulations), efficient procedures of random generation, encoding, and straight-line drawing are described. In particular, the first optimal encoder for the combinatorial incidences of polygonal meshes with spherical topology is developed. Starting from a generator for 3-connected maps, a new random generator for planar graphs is introduced. The complexity of generation is the best currently known: quadratic (in expectation) for exact-size sampling and linear (in expectation) for approximate-size sampling. Finally, several straight-line drawing algorithms for planar maps are introduced. The procedures are both simple to describe and very efficient, yielding the best known grid size for two families of maps: triangulations of the 4-gon with no filled 3-cycle —called irreducible— and quadrangulations. The algorithms presented in the thesis take advantage of several combinatorial structures on planar maps (orientations, partitions into spanning trees) as well as new bijective constructions.

Introduction

Cette thèse a pour vocation d'explorer les interactions fructueuses entre la combinatoire et l'algorithmique. Pour dire les choses simplement "comprendre la structure des objets permet de les manipuler efficacement". Notre étude porte principalement sur les cartes planaires, objets fondamentaux au carrefour entre les mathématiques, l'algorithmique, et la physique statistique (une carte planaire est un graphe planaire dessiné sans croisement d'arête sur la sphère ou le plan).

L'ubiquité des cartes planaires apparaît dès leur définition, les cartes pouvant être vues selon les contextes comme polyèdres convexes, surfaces discrètes, ou graphes plongés. Ainsi, la question naturelle d'énumérer les polyèdres convexes, qui remonte à Euler, est en fait un problème de comptage de cartes (précisément, de cartes dites 3-connexes).

En tant que surface "discrète", une carte représente une surface fermée spécifiable en utilisant une quantité finie d'informations. Typiquement une carte est obtenue par un jeu de construction consistant à assembler une surface fermée en prenant une collection de polygones et en collant leurs bords deux à deux de sorte qu'il ne reste plus de bord libre. Par exemple, le cube s'obtient par recollement de six carrés et le tétraèdre par recollement de quatre triangles assemblés convenablement. Seule l'information sur le recollement intervient ici dans la définition de l'objet, de sorte qu'on parle de surface *topologique*, par opposition aux surfaces *géométriques* spécifiées par les coordonnées de leurs points.

Dans ce contexte, les cartes sont utilisées en physique comme modèles de surfaces aléatoires discrètes, voir la figure 1(b). Le concept de surface aléatoire est central en physique théorique moderne dans une branche appelée gravitation quantique bidimensionnelle; les surfaces aléatoires discrètes sont alors un bon moyen d'appréhender les phénomènes, par exemple pour l'étude des transitions de phase [**19**]. En informatique appliquée, on utilise aussi beaucoup les cartes dans le cadre de la géométrie algorithmique. En effet, pour pouvoir être stockées et transmises efficacement, les surfaces manipulées sont discrétisées sous forme de maillages, voir la figure 1(a). On distingue alors l'information géométrique —les coordonnées des sommets— de l'information combinatoire —les incidences entre faces, arêtes, et sommets— qui est en fait exactement une carte. Il est donc crucial de bien comprendre la structure des cartes pour compresser efficacement l'information combinatoire sur le maillage, qui est la plus coûteuse en pratique.

Par l'observation qu'une carte est décrite par des relations d'incidence entre sommets, faces, et arêtes, un autre point de vue naturel consiste à considérer une carte comme un graphe dessiné sur une surface sans croisement d'arêtes; on parle plus précisément de "plongement" topologique sur une surface. Notons qu'un même graphe peut être plongé sur des surfaces différentes. Par exemple, la figure 2 illustre le plongement du graphe complet à quatre sommets dans le plan, sur la sphère (topologique), et sur le tore. Comme le plan est équivalent à la sphère privée d'un

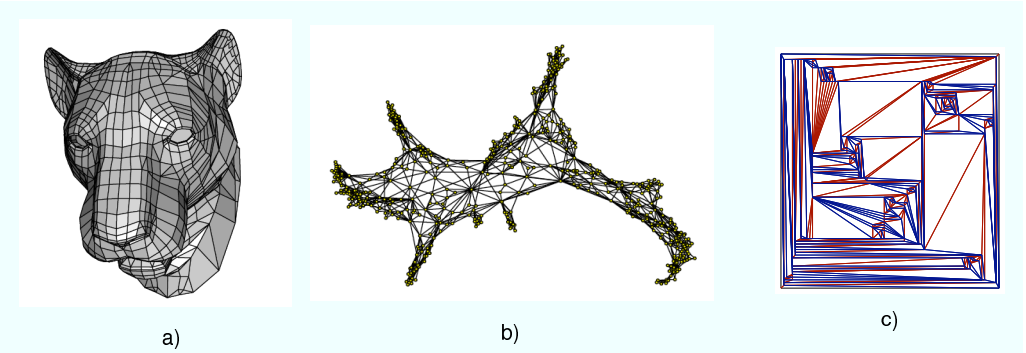


FIGURE 1. Un maillage (Fig. a), une surface aléatoire discrète (Fig.b, figure obtenue par Gilles Schaeffer par génération aléatoire), et le dessin d'une triangulation aléatoire (Fig.c, l'algorithme utilisé est présenté au chapitre 5.)

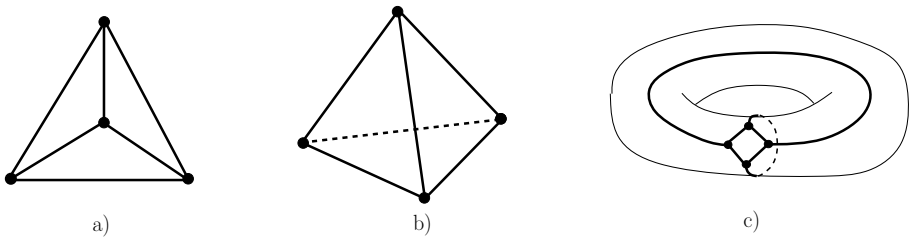


FIGURE 2. Le graphe complet à 4 sommets plongé dans le plan (Fig. a), sur la sphère topologique (Fig. b), et sur le tore (Fig. c).

point, considérer les cartes dans le plan ou sur la sphère revient au même (au choix d'une face extérieure près). Ainsi, la figure 2(a) s'obtient à partir de la figure 2(b) par simple projection stéréographique.

Les cartes apportent un éclairage précieux pour l'étude des graphes. En effet, l'enrichissement par une structure de plongement donne une meilleure intuition géométrique et permet d'utiliser des outils spécifiques efficaces: citons par exemple la formule d'Euler reliant les nombres de sommets, faces, et arêtes, ou encore le théorème de Jordan selon lequel tout cycle partitionne la sphère topologique en deux régions disjointes. En tant que graphes plongés, les cartes s'inscrivent dans le cadre plus général des graphes stables par mineur, développé par Robertson et Seymour dans leur fameuse série d'articles "Graph minors". Cette théorie très puissante implique par exemple que l'obstruction au plongement d'un graphe sur une surface donnée ne peut être due qu'à un nombre fini de configurations (appelées mineurs) interdites. Dans le cas planaire, le théorème de Kuratowski [**81**] assure que le nombre de configurations interdites n'est que de deux. Dans le cas de surfaces de genre plus élevé (c'est-à-dire des surfaces ayant des "trous"), par exemple le tore, le nombre de configurations interdites augmente très rapidement. Plus généralement, les problèmes de comptage et d'algorithmique sont beaucoup plus difficiles à résoudre en genre supérieur, et la plupart des très jolies propriétés combinatoires vérifiées dans le cas planaire ne semblent pas s'y adapter.

Dans cette thèse, nous nous concentrons uniquement sur les cartes planaires et adoptons le point de vue des cartes comme graphes plongés; ce cadre est pratique et usuel pour décrire les applications algorithmiques. Un exemple d'application algorithmique abordée dans cette thèse est le dessin automatique de graphes planaires, une discipline qui a connu un très grand essor ces dernières années et répond à de nombreux besoins industriels comme la visualisation de diagrammes, la disposition de composants sur un circuit électrique, ou encore le déploiement de réseaux. Dans cette thèse, nous nous penchons sur l'un des problèmes les plus classiques dans ce domaine, à savoir le dessin de carte planaire sur une grille (de sorte que les sommets ont des coordonnées entières) et tel que les arêtes sont représentées par des segments. La figure 1(c) illustre un tel plongement pour une carte triangulée aléatoire de 200 sommets (le générateur alátoire est présenté au chapitre 4 et l'algorithme de dessin au chapitre 5).

Petit historique des cartes planaires. Bien qu'étant des objets très naturels, les cartes n'ont été étudiées de manière systématique qu'à partir des années 1960 avec les travaux fondateurs de Tutte. Originellement, la motivation était de prouver le théorème des 4 couleurs. Ce but n'a pas été atteint, mais les efforts entrepris ont ouvert la voie à la combinatoire énumérative des cartes. En particulier, Tutte a développé dans sa série d'articles "A census of..." [**110, 111, 112, 113**] une méthode systématique très astucieuse de comptage de cartes par séries génératrices, récemment formalisée dans un cadre général [**21**]. De manière surprenante, ces résultats ont été retrouvés par des physiciens [**26**], suite à l'observation que les cartes planaires sont les diagrammes de Feynman dominants apparaissant dans le calcul d'intégrales de matrices sous distribution gaussienne.

Les formules d'énumération obtenues sont étonnamment simples, étant donnée la complexité des méthodes de calculs mises en œuvre, que ce soit par séries génératrices ou par intégrales de matrices. Pour de nombreuses familles naturelles de cartes planaires (triangulations, quadrangulations, cartes sans boucles), les formules de comptage s'expriment au moyen de coefficients binomiaux, suggérant de fortes propriétés combinatoires sous-jacentes. Par exemple, le nombre c_n de cartes planaires "enracinées" (l'enracinement est une opération naturelle permettant d'éliminer les symétries) à n arêtes est

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

Une percée importante concernant la combinatoire structurelle des cartes et graphes planaires est survenue en 1989, lorsque Schnyder a introduit la notion de *réaliseur* (aussi appelé forêt de Schnyder) et a montré que toute triangulation (graphe planaire maximal plongé) est naturellement partitionnée au niveau de ses arêtes en trois arbres couvrants [**101**]. Ce résultat fondamental a donné un nouvel éclairage sur les graphes planaires, permettant d'obtenir des critères originaux de planarité en terme d'ordres partiels [**46, 101, 119**] et des algorithmes efficaces pour dessiner [**16, 102**] et coder [**17, 18**] les graphes planaires. De manière plus générale, le concept d'orientation s'est avéré très fructueux pour décrire les propriétés combinatoires des cartes, avec notamment les travaux de Propp [**96**], De Fraysseix *et al* [**37**], la thèse d'Ossona de Mendez sur les orientations bipolaires [**92**] et le récent article de Felsner [**48**].

Concernant la combinatoire bijective des cartes planaires, les premiers travaux remontent à Cori et Vauquelin [**33**] qui établissent de manière récursive des bijections entre cartes et arbres dits "bien étiquetés". Les premières constructions bijectives directes ont été données par G. Schaeffer dans sa thèse [**99**], où il décrit une reformulation non récursive de [**33**] et introduit de nouvelles bijections reposant sur le concept d'arbres bourgeonnants, qui ont été récemment généralisées par Bouttier, Di Francesco et Guitter [**23**]. Ces constructions ont notamment permis de résoudre combinatoirement des modèles de physique statistique sur carte aléatoire, tels que le modèle d'Ising [**22**] ou le modèle à particules dures [**24**]. Elles rendent également possible de définir de manière probabiliste une notion de carte limite continue [**85**] et d'étudier les propriétés métriques [**30, 84**] et topologiques [**66**] de l'object limite. De nouvelles constructions bijectives dues à O. Bernardi sont venues récemment enrichir le domaine, permettant de compter respectivement les cartes boisées [**7**] et les cartes triangulées [**9**] via comptage de chemins dans un demi-plan.

Notre approche. Nous nous concentrons dans cette thèse sur les cartes planaires vues comme des graphes plongés dans le plan. Comme mentionné plus haut, les cartes planaires ont des propriétés structurelles fortes: existence d'orientations, de partitions en arbres couvrants... Notre approche consiste à étudier en détail ces propriétés combinatoires et de les faire fructifier sous forme d'algorithmique efficace sur les cartes et graphes planaires, pour la génération aléatoire, le codage, et le dessin. Nous nous situons donc à l'interaction entre la théorie et la pratique.

L'avantage qu'il y a à tirer partie de propriétés combinatoires structurelles a priori plutôt théoriques est double: tout d'abord les algorithmes dérivés des structures étudiées sont particulièrement efficaces et simples à décrire. Par exemple, les algorithmes de dessin donnés dans cette thèse sont très faciles à effectuer à la main, par opposition aux premiers algorithmes historiquement développés, tel le "spring embedding" de Tutte [**114**], qui nécessitent des calculs itérés sur les nombres rationnels. Le second point est que la bonne compréhension des propriétés structurelles sous-jacentes permet d'analyser les algorithmes et de valider les performances observées expérimentalement. Ainsi nous avons pu prouver rigoureusement, en utilisant notamment des outils d'analyse de singularité [**52**], que le nouvel algorithme de dessin de triangulations introduit dans cette thèse donne un gain d'un facteur 27/22 pour la taille de la grille par rapport aux meilleurs algorithmes précédemment connus.

À l'origine de notre travail se trouve une bijection de G. Schaeffer et D. Poulalhon [**95**] pour compter les triangulations planaires et obtenir des procédures efficaces de génération aléatoire et de codage pour cette famille de cartes (qui correspondent aux graphes planaires maximaux). La bijection fait apparaître de manière frappante la structure de réaliseur (forêt de Schnyder) sur les triangulations, révélant un lien entre propriétés structurelles et bijectives des cartes planaires. C'est précisément ce lien que nous explorons dans le chapitre central de cette thèse (chapitre 3), en nous appuyant sur un article récent de Felsner [**48**] étudiant les propriétés d'orientations des cartes planaires avec degré sortant fixé pour chaque sommet. Notre approche combinatoire des cartes est donc au croisement entre l'approche structurelle par les orientations et l'approche bijective développée par G. Schaeffer. En cela, nous sommes proches des récents travaux de O. Bernardi [**7**], qui s'appuie également sur la notion d'orientation pour trouver une méthode bijective générale de comptage de cartes boisées (cartes munies d'un arbre couvrant).

Contributions. Notre travail se situe dans le domaine de l'algorithmique sur les cartes et graphes planaires et trouve notamment des applications en géométrie algorithmique et en tracé automatique de graphes. Les principales contributions de cette thèse sont les suivantes:

- Le premier algorithme de codage optimal des incidences entre faces, arêtes, et sommets de maillages polygonaux de la sphère (qui sont des cartes 3-connexes), atteignant la borne inférieure de 2 bits par arête (théorème 4.1 page 127). Un travail récent de Castelli-Aleardi, Devillers et Schaeffer [**28**] assure que notre codage peut être enrichi tout en conservant l'optimalité (asymptotique), de sorte à rendre possible de naviguer rapidement sur le maillage et de répondre à des requêtes locales en temps constant.
- Un nouvel algorithme de génération aléatoire uniforme de graphes planaires, tel que la complexité moyenne de chaque tirage est quadratique en taille exacte et linéaire en taille approchée (théorèmes 4.3 et 4.4 page 132). Par contraste, le meilleur algorithme précédemment connu a une complexité un peu au dessus de n^7 [**13**]. Notre générateur devrait notamment trouver une grande utilité pour estimer la complexité moyenne des innombrables algorithmes opérant sur les graphes planaires (bien souvent, seule une borne supérieure sur la complexité dans le cas le pire peut être facilement prouvée).
- Un nouvel algorithme de dessin en lignes droites sur la grille (théorème 5.2 page 160), qui s'applique aux familles de triangulations du carré sans triangle séparateur —dites irréductibles— et aux quadrangulations. En terme de taille de la grille, le gain par rapport aux meilleurs algorithmes précédemment connus est de 27/22 pour les triangulations irréductibles du carré (théorème 5.7 page 175) et de 27/26 pour les quadrangulations (théorème 5.9 page 180); les preuves font appel à des techniques de combinatoire analytique.

Pour développer ces algorithmes, nous nous appuyons principalement sur deux ingrédients:

(1) Une étude théorique des propriétés structurelles des cartes planaires, où nous obtenons également des résultats originaux, notamment des algorithmes d'orientations (chapitre 2). Un de nos algorithmes d'orientations (théorème 2.1 page 66) est un ingrédient clé pour obtenir le meilleur algorithme connu de dessin de graphes planaires en lignes droites, tel qu'introduit par Bonichon *et al* [**16**].

(2) De nouvelles constructions bijectives de cartes planaires (chapitre 3), intimement liées aux propriétés structurelles, en particulier à la notion d'orientations à degré sortant fixé pour chaque sommet, qui sont appelées α-orientations. Ces constructions permettent de compter, coder, et générer aléatoirement de nombreuses familles bien connues de cartes planaires (triangulations, quadrangulations...).

Tous les algorithmes présentés dans cette thèse ont été implantés dans le langage Java et sont ou seront bientôt disponibles sur la page web de l'auteur.

Plan. Cette thèse se compose de 5 chapitres d'une trentaine de pages chacun. Les dépendances de lecture sont les suivantes:

chapitre 1 $\rightarrow$ {chapitre 2, chapitre 3, chapitre 5},

chapitre 3 $\rightarrow$ {chapitre 4, chapitre 5 (deuxième partie)}.

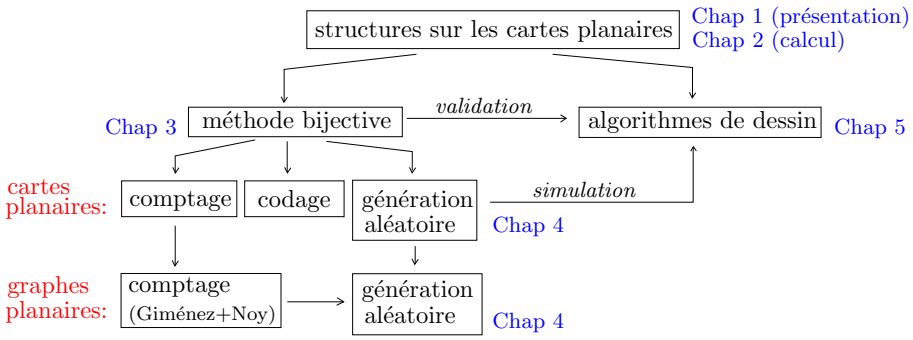


FIGURE 3. Le schéma général de la thèse.

Le schéma général de la thèse est illustré par la figure 3.

Le chapitre 1 "*Combinatorial structures on planar maps*" considère plusieurs familles naturelles de cartes planaires et décrit pour chaque famille une structure combinatoire qui caractérise l'appartenance à la famille. Par exemple, une carte avec deux sommets adjacents marqués est 2-connexe (c'est-à-dire sans sommet déconnectant) si et seulement si elle admet une orientation acyclique avec les deux sommets marqués comme uniques éléments extrémaux. Une telle orientation est dite bipolaire. Dans le cas des cartes 3-connexes (cartes sans paire de sommets séparante), la structure combinatoire est un ensemble de trois arbres couvrants —appelée forêt de Schnyder— avec des relations d'incidence spécifiques. Toutes les structures combinatoires considérées peuvent être formulées en termes d'orientations à degré sortant fixé pour chaque sommet, appelées α-orientations dans la littérature [**48**]. En particulier, une propriété que nous rappelons est que l'ensemble des orientations d'une carte qui réalisent une affectation fixée des degrés sortants est un treillis distributif. Notre contribution dans ce chapitre est de faire une étude détaillée de la structure combinatoire qui caractérise la famille des triangulations du 4-gone qui sont sans 3-cycle séparateur, dites irréductibles. La structure en question, introduite par Xin He dans [**73**] et appelée ici structure transverse, est une partition des arêtes internes en deux orientations bipolaires transverses. Notre résultat principal (théorème 1.4 page 50) est de donner une formulation des structures transverses en termes d'α-orientation et de décrire l'opération locale de navigation (appelée "flip") sur le treillis distributif associé.

Le chapitre 2 "*Efficient computation*" se concentre sur le calcul efficace des sructures combinatoires présentées dans le premier chapitre. Plus précisément, pour chaque structure combinatoire et chaque carte pouvant être munie d'une telle structure, nous décrivons un algorithme qui calcule en temps linéaire la structure dite minimale: celle réalisant le minimum du treillis distributif associé. Nous rappelons tout d'abord l'algorithme pour les orientations bipolaires donné par Ossona de Mendez [**92**], dont le principe nous inspire un nouvel algorithme de calcul de structures transversales (après ceux donnés par Kant et He dans [**78**]). Notre algorithme a l'avantage de pouvoir être spécifié pour calculer la structure transversale minimale en temps linéaire (théorème 2.7 page 70). Nous introduisons également un algorithme permettant de calculer la forêt de Schnyder minimale d'une carte 3-connexe en temps linéaire (théorème 2.1 page 66). Ces algorithmes de calcul de

structures minimales sont utiles à la fois pour coder les cartes planaires (chapitre 3) et les dessiner sur la grille (chapitre 5).

Le chapitre 3 "*Bijective counting of maps*" est dédié à des constructions bijectives de cartes à partir d'arbres (cartes planaires à une face). Tout d'abord, nous décrivons en section 1 une méthode bijective générale reposant sur la notion d'α-orientation minimale et s'appliquant à des familles de cartes enracinées. Nous reformulons ainsi plusieurs bijections dans un cadre unifié. Ensuite, nous introduisons en section 2 deux nouvelles constructions bijectives s'appliquant à des cartes non-enracinées. La première construit une dissection quadrangulée d'un hexagone sans 4-cycle séparateur —dite irréductible— à partir d'un arbre binaire. La seconde construit une dissection triangulée d'un carré sans 3-cycle séparateur —dite irréductible— à partir d'un arbre ternaire. Nous en déduisons la première preuve bijective (proposition 3.5 page 100) de la formule d'énumération des triangulations planes non enracinées, trouvée par Brown [**27**].

Le chapitre 4 "*Algorithmic applications*" exploite les bijections présentées au chapitre 3 pour obtenir des algorithmes efficaces pour la génération aléatoire et le codage de cartes planaires. On utilise la bijection entre arbres binaires et dissections irréductibles de l'hexagone pour compter les cartes 3-connexes enracinées (proposition 4.1 page 124), effectuer leur génération aléatoire en temps linéaire, et les coder avec seulement 2 bits par arête (théorème 4.1 page 127), atteignant la borne inférieure donnée par l'entropie. De manière similaire, la bijection entre arbres ternaires et triangulations irreductibles du carré permet de réaliser l'énumération (proposition 4.2 page 125), la génération aléatoire en temps linéaire (proposition 4.5 page 130), et le codage optimal (théorème 4.2 page 128) des triangulations enracinées dites 4-connexes, qui ont un degré de connectivité de plus que les triangulations classiques. La contribution principale du chapitre est un générateur uniforme de graphes planaires (la différence avec les cartes est qu'il n'y a pas de plongement attaché) extrêmement efficace, qui réalise des appels répétés au générateur bijectif de cartes 3-connexes et assemble les cartes générées en un graphe planaire. Les probabilités de branchement sous-jacentes au processus d'assemblage reposent sur le cadre des générateurs de Boltzmann [**43**]. Le générateur de graphes planaires obtenu a une complexité moyenne quadratique en taille exacte et linéaire en taille approchée (théoremes 4.3 et 4.4 page 132).

Le chapitre 5 "*Straight-line drawing*" introduit un nouvel algorithme de dessin en lignes droites sur une grille, basé sur les structures transversales (théorème 5.2 page 160). Il se situe dans la famille des algorithmes par comptage de faces, comme l'algorithme de dessin de cartes 3-connexes récemment développé par Bonichon *et al* [**16**]. Notre algorithme s'applique au dessin des triangulations irréductibles du carré (théorème 5.4 page 165) et des quadrangulations (théorème 5.5 page 168). Pour les triangulations irréductibles du carré (resp. les quadrangulations), la grille obtenue est avec grande probabilité de taille $11n/27 \times 11n/27$ (resp. $13n/27 \times 13n/27$) à petites fluctuations près, ce qui représente en probabilité un gain par un facteur $27/22$ (resp. $27/26$) par rapport aux meilleurs algorithmes connus précédemment [**89, 10**], lesquels ne peuvent garantir qu'une grille $n/2 \times n/2$. Nous analysons également en probabilité la taille de la grille pour l'algorithme de dessin de Bonichon *et al*, et montrons qu'une carte 3-connexe aléatoire uniforme à n sommets et $\lfloor \beta n \rfloor$ faces est dessinée avec grande probabilité sur une grille $(1-(4\beta)^{-1})n \times (1-(4\beta)^{-1})n$ à petites fluctuations près (théorème 5.6 page 170).

Notre méthode d'analyse probabiliste de la taille de la grille pour les algorihmes par comptage de faces repose sur plusieurs ingrédients: 1) la taille de la grille est reformulée en termes de paramètres explicites de la carte munie de sa structure combinatoire minimale, 2) les paramètres en question s'avèrent correspondre à des paramètres explicites de l'arbre associé bijectivement, 3) l'analyse probabiliste des paramètres sur l'arbre est effectuée en utilisant un outil bien connu de la combinatoire analytique, le théorème des quasi-puissances [**52**].

Publications sur les travaux de thèse. La bijection entre arbres binaires et dissections irréductibles de l'hexagone (chapitre 3) et ses applications au traitement algorithmique de cartes 3-connexes (maillages sphériques polygonaux) a donné lieu à une communication à la conférence ACM SODA'05 à Vancouver [**64**]. L'article long est à paraître dans une édition spéciale du journal *Transactions on Algorithms*, sous le titre:

> "Dissections, orientations, and trees, with applications to optimal mesh encoding and random sampling", avec Dominique Poulalhon et Gilles Schaeffer.

Les travaux concernant les triangulations irréductibles du carré (structures transversales, algorithme de dessin, comptage, codage, et génération aléatoire de triangulations 4-connexes) ont donné lieu à une communication à la conférence Graph Drawing'05 à Limerick [**58**]. L'article long est à paraître dans une édition spéciale de *Discrete Math.*, sous le titre

> "Transversal structures on triangulations: combinatorial study and straight-line drawing algorithm".

L'algorithme de dessin de quadrangulations a été présenté sous forme de communication courte (6 pages) à la conférence Graph Drawing'06 à Karlsruhe, sous le titre

> "Straight-line drawing of quadrangulations" [**60**].

L'algorithme de génération aléatoire de graphes planaires a été présenté à la conférence Analysis of Algorithms AofA'05 à Barcelone, sous le titre

> "Quadratic exact-size and linear approximate-size random generation of planar graphs" [**57**].

Une version journal est soumise [**62**].

Autres travaux. Outre l'algorithmique sur les cartes planaires, je me suis intéressé à plusieurs thématiques durant la période de thèse, qui sont principalement le comptage à symétrie près, la génération aléatoire, et le comptage probabiliste.

Comptage à symétrie près. Tout d'abord j'ai développé une méthode de comptage de cartes dites "non enracinées", un travail engagé pendant mon stage de DEA sous les précieux conseils de Gilles Schaeffer. Comme il s'agit d'énumération non enracinée, les symétries potentielles doivent être prises en compte. Notre méthode consiste à visualiser les symétries des cartes au niveau d'une décomposition arborescente, voir la figure 4(a). On en déduit l'énumération (par séries génératrices) de nombreuses familles de cartes non enracinées. En particulier, notre approche permet d'obtenir les formules les plus simples connues actuellement pour énumérer les cartes 3-connexes, et donne le meilleur algorithme de comptage. Cette famille est particulièrement intéressante. En effet, par un théorème de Steinitz, les cartes 3-connexes correspondent exactement aux polyèdres convexes à isotopie près, dont le problème de comptage avait déjà été attaqué par Euler [**55**]. Ces résultats ont été

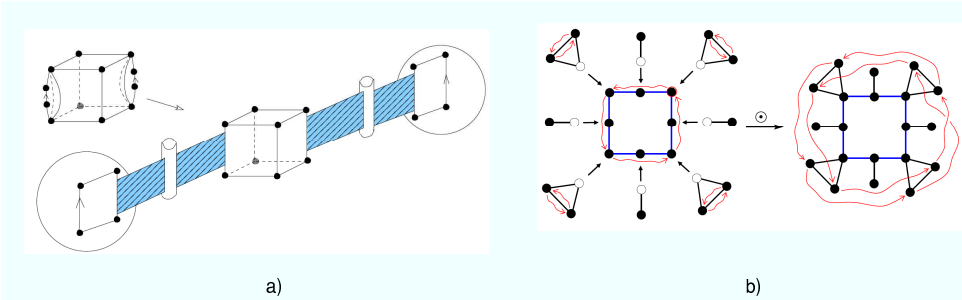


FIGURE 4. Décomposition arborescente d'une quadrangulation symétrique (Fig.a), et opération de substitution préservant une symétrie sous-jacente (Fig.b).

présentés à la conférence FPSAC'05. L'article long a été publié dans le Séminaire Lotharingien de Combinatoire sous le titre

"Counting unrooted maps using tree-decomposition" [**61**]

Durant un séjour à Berlin, j'ai eu la chance de collaborer avec Manuel Bodirsky, Mihyun Kang et Stefan Vigerske (Université Humboldt de Berlin). En partant des idées principales de mon travail sur le comptage de cartes non enracinées, nous avons conçu une méthode très générale d'énumération de structures non étiquetées, qui combine un opérateur de pointage "non biaisé" —pointage d'un cycle de symétrie— et de nouvelles constructions adaptées à cet opérateur. La figure 4(b) illustre la construction par substitution, qui généralise la construction par substitution sur les cartes non enracinées (figure 4(a)). Ce travail a été présenté à la conférence ACM SODA'07 à la Nouvelle Orléans sous le titre

"An unbiased pointing operator for unlabeled structures, with applications to counting and sampling" [**12**], avec Manuel Bodirsky, Mihyun Kang, and Stefan Vigerske.

Dans le domaine du comptage à symétries près, j'ai travaillé également à Berlin sur un problème concernant les polytopes qui m'a été donné par Günter Ziegler. Précisément, il s'agit de compter le nombre de polytopes en dimension d avec $(d+3)$ sommets et à isotopie/réflexion près. Par l'usage de "diagrammes de Gale" [**71**, Sect. 6.3], ce problème se ramène à compter certains diagrammes du plan à rotation/réflexion près. Les symétries sont ici prises en compte par des outils classiques de théorie des groupes tels que le lemme de Burnside, et l'énumération se fait par automates et séries génératrices. Par l'usage d'automates, ma présentation diffère d'une première résolution du problème par Lloyd [**83**], dans laquelle une erreur s'était glissée. Ce travail est paru dans le Electronic Journal of Combinatorics sous le titre

"Counting d-polytopes with $(d+3)$ vertices" [**59**].

Génération aléatoire. Trouver des méthodes efficaces de génération aléatoire uniforme trouve de nombreuses applications, par exemple pour tester des conjectures, estimer le complexité moyenne des algorithmes, et observer des propriétés structurelles de grands objets aléatoires. Comme il est nécessaire de tirer de très grands objets pour observer des phénomènes asymptotiques, les générateurs de complexité

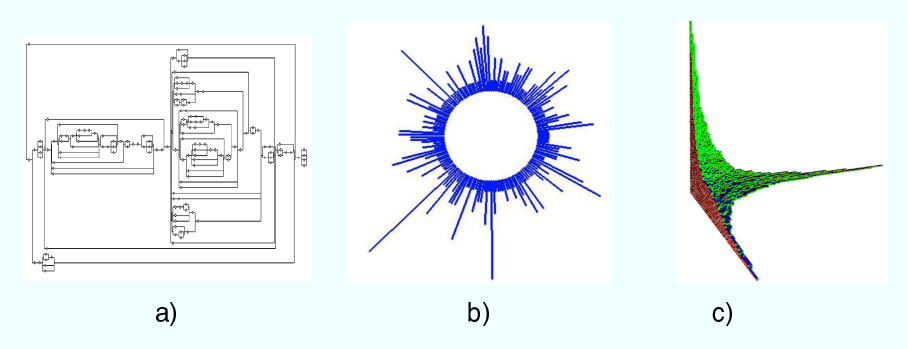


FIGURE 5. Un graphe série-parallèle (Fig.a), une composition circulaire (Fig.b), et une partition plane (Fig.c) engendrés par génération de Boltzmann (figures obtenues par Carine Pivoteau).

linéaire sont nettement préférables. Dans ce but, une méthode très puissante a été récemment introduite, appelée génération de Boltzmann [**43**]. La complexité de ces générateurs est linéaire, au prix d'une légère perte de contrôle sur la taille de la sortie.

Avec Carine Pivoteau (durant son stage de Master) et Philippe Flajolet, nous avons étendu le premier cadre de génération décrit dans [**43**] pour prendre en compte les constructions impliquant des symétries (Multiensemble, Ensemble, Cycle). La variété des classes combinatoires traitables par génération de Boltzmann est ainsi considérablement accrue. Les figures 5(a) et (b) ont été obtenues par Carine Pivoteau et montrent respectivement un circuit série-parallèle et une composition circulaire engendrés par nos générateurs. Ce travail a été présenté au Workshop ANALCO'07 à la Nouvelle Orléans, sous le titre

> "Boltzmann sampling of unlabelled structures" [**50**], avec Philippe Flajolet et Carine Pivoteau.

En collaboration avec Carine Pivoteau et Olivier Bodini, nous avons également développé un générateur de Boltzmann pour la classes des partitions planes, qui peuvent être vues comme des empilements de cubes "tassés" dans l'espace $\mathbb{R}^3_+$. L'idée est de tirer partie d'une bijection de Pak entre partitions planes et une classe décomposable. Les règles de génération de [**50**] combinées avec la bijection permettent d'obtenir un générateur très efficace, de complexité quasi-linéaire en taille approchée. Ce travail a fait l'objet d'une communication à la conférence GASCOM'06, sous le titre

> "Random sampling of plane partitions" [**11**], avec Olivier Bodini et Carine Pivoteau.

La figure 5(c) montre une partition plane aléatoire engendrée par notre générateur. On observe par simulation une surface limite, dont la description a été donnée par Cerf et Kenyon [**29**].

Comptage probabiliste. La fouille de données est un domaine de recherche très actif, dont le but est d'extraire des informations dans des flux de données massifs. Ces flux peuvent souvent être vus comme des multi-ensembles, par exemple le trafic internet transitant par un routeur est une suite de paquets, chaque paquet étant identifé

par une paire (adresse source, adresse destination), appelée connexion. Dans ce contexte, les éléments du multi-ensemble sont les paquets et les éléments distincts sont les connexions. Le nombre d'éléments distincts du multi-ensemble, appelé cardinalité, est une information importante sur le flux, qui permet notamment de détecter des attaques sur le réseau internet. En pratique, on se contente de calculer une estimée probabiliste de la cardinalité, ce qui permet de concevoir des procédures très rapides et ne nécessitant qu'une mémoire auxiliaire d'ordre constant. De nombreux algorithmes obéissant à ce principe ont été développés (voir [**49**] pour un survol du domaine). Dans un travail récent avec Philippe Flajolet, Olivier Gandouet, et Frédéric Meunier, nous avons introduit une nouvelle estimée basée sur la moyenne harmonique, qui réalise le meilleur compromis actuellement connu entre précision et mémoire auxiliaire. Ce travail va être présenté à la conférence AofA'07 à Antibes, sous le titre

> "HyperLogLog, the analysis of a near-optimal cardinality estimation algorithm", avec Philippe Flajolet, Olivier Gandouet et Frédéric Meunier.

Dans un autre travail en collaboration avec Frédéric Giroire, nous avons adapté l'algorithme MinCount [**69**] que Frédéric a développé dans sa thèse au modèle de la fenêtre coulissante [**34**], de manière à répondre à des requêtes sur la cardinalité en temps réel et pour une fenêtre de temps fixée. Ce travail a été présenté au Workshop ANALCO'07 à la Nouvelle Orléans, sous le titre

> "Estimating the number of active flows in a data stream over a sliding window " [**63**], avec Frédéric Giroire.

CHAPTER 1

Combinatorial structures on planar maps

Introduction. The aim of this chapter is first to provide a short and accessible presentation of planar maps. For a more detailed introduction, see the introductory chapter in the theses of Bernardi [**8**] and Schaeffer [**99**]. Then, the key point we develop is that, for several well known families of planar maps, there exists a particular combinatorial structure that characterises the maps of the family. The combinatorial structure is often an orientation with specific properties, which can be combined with a coloration of the edges such that the induced subgraph in each color has a simple form (e.g. a tree). As we will see in Chapter 3, 4, and 5, such a structure is a precious tool to study the corresponding family of maps: in each case, it makes it possible to count bijectively the family according to a size parameter and gives rise to efficient algorithms for random sampling, encoding, and straight-line drawing.

After presenting planar maps in Section 1, we will focus on four examples of combinatorial structures: 1) Eulerian orientations for the family of Eulerian maps, 2) bipolar orientations for the family of 2-connected maps, 3) Schnyder woods for families related to 3-connected maps, 4) transversal structures for families related to 4-connected maps. As we will see, all these structures can be formulated in terms of orientations with prescribed outdegree for each vertex. These are called α-orientations and obey nice structural properties —in particular lattice properties— that are recalled in Section 2.

Results obtained in this chapter. Most of the results presented in this chapter are not new. The theory of α-orientations is developed in not less than 3 sources: an unpublished article by Propp [**96**], the thesis of Ossona de Mendez [**92**], and a more recent article by Felsner [**48**]. Our presentation of Eulerian orientations and Schnyder woods closely follows [**48**]. In the literature, the lattice property of bipolar orientations is studied in [**92**] and the correspondence with 2-orientations of quadrangulations is described in [**38**]; our presentation in Section 4 sketches how the lattice property of bipolar orientations is easily described via the lattice structure of 2-orientations of a quadrangulation; essentially we combine the arguments presented in [**92**] and in [**38**] so as to provide a shorter presentation of the results.

Our main contribution in this chapter is a detailed combinatorial study of transversal structures, which were introduced by Xin He —under the name of regular edge-labeling— for triangulations of the 4-gon with no filled 3-cycle [**73**], called hereafter irreducible triangulations. We extend the definition to any planar map, show a characterisation as a transversal pair of bipolar orientations, and describe conditions of existence. Then we study the set of transversal structures of a fixed irreducible triangulation T. Our main result is Theorem 1.4 page 50, where we show that the set of transversal structures of T is a distributive lattice, with an

explicit simple flip operation. Similarly as for bipolar orientations, the description is obtained via α-orientations of an associated quadrangulated map.

Motivations. The results on orientations recalled and obtained in this chapter are extensively used to develop a general bijective framework for counting planar maps, and to analyse the straight-line drawing algorithms presented in Chapter 5. Our study of transversal structures has also a specific application to a result of cartography [**104**], where an exhaustive generation of transversal structures is useful to obtain the best cartographic representation (the discussion is given in Section 6).

1. Planar maps

1.1. Definitions. Graphs. A graph G is given by a set $V = \{v_1, \ldots, v_n\}$ of *vertices* and a set E of unordered pairs of vertice that are called *edges* . An edge is classically written $e = \{v, v'\}$. The vertices v and v' are called the *extremities* of e; if $v = v'$ e is called a *loop*, otherwise e is said to *connect* v and v'. A *multiple edge* is a pair of distinct vertices connected by at least 2 edges. The *degree* of a vertex v is the number of edges incident to v, with multiplicity 2 for loops incident to v. A *path* of G is a sequence $v_0, e_0, v_1, \ldots, e_k, v_{k+1}$ of vertices and edges such that, for $0 \leq i \leq k$ the edge e_i connects v_i and v_{i+1}. The vertices v_0 and v_{k+1} are said to be *connected* by the path. The path is called *simple* if the vertices $v_0, \ldots, v_{k+1}$ are different. A graph is said to be *connected* iff any pair of vertices are connected by a path. A *cycle* of G is a cyclic sequence $v_0, e_0, v_1, \ldots, v_k, e_k$ such that for each $0 \leq i \leq k$, the edge e_i connects v_i to $v_{(i+1) \mod k}$. The cycle is *simple* if the vertices $v_0, \ldots, v_k$ are different. A graph is *oriented* if all its edges are directed. An edge e from a vertex v to a vertex v' is denoted by $e = (v, v')$, the vertex v is called the *origin* of e and v' is called the *end-vertex* of e. For each vertex v, the number of edges whose origin is v is called the outdegree of v and is denoted by $\mathrm{Outdeg}(v)$. Similarly, the number of edges whose end-vertex is v is called the indegree of v and is denoted by $\mathrm{Indeg}(v)$. An *oriented path* is a path $v_0, e_0, v_1, \ldots, e_k, v_{k+1}$ such that, for $0 \leq i \leq k$, e_i goes from v_i to v_{i+1}; and a *circuit* is a cyclic sequence $v_0, e_0, v_1, \ldots, v_k, e_k$ such that for each $0 \leq i \leq k$, the edge e_i goes from v_i to $v_{(i+1) \mod k}$. The notion of simple oriented path and simple circuit are defined similarly as for unoriented graphs.

Embeddings. An embedding $\mathcal{D}$ (or drawing) of a graph G in the plane $\mathbb{R}^2$ is given by an injective mapping $\Phi_V : v \in V \to (x_v, y_v)$ from the vertices of G to plane points, and by a mapping Φ_E from the edges of G to open smooth arcs in the plane such that the extremities of any edge e are mapped by Φ_V to the extremities of the arc $\Phi_E(e)$. An embedding is said to be *planar* iff the (closure of the) arcs $(\Phi_E(e))_{e \in E}$ do not meet except at common extremities, see Figure 1(a). A graph G that admits a planar embedding is called a *planar graph* . Notice that a planar graph G admits infinitely many planar embeddings. However it admits only finitely many if the embeddings are considered up to isotopy, i.e., two planar embeddings $\mathcal{D}_0$ and $\mathcal{D}_1$ are identified if there exists a continuous family $\{\mathcal{D}(t), t \in [0,1]\}$ of planar embeddings of G such that $\mathcal{D}(0) = \mathcal{D}_0$ and $\mathcal{D}(1) = \mathcal{D}_1$. In the plane, this is equivalent to the existence of an orientation-preserving homeomorphism mapping $\mathcal{D}_0$ to $\mathcal{D}_1$. The isotopy relation allows us to abstract the geometry and to consider only the embeddings at the topological level.

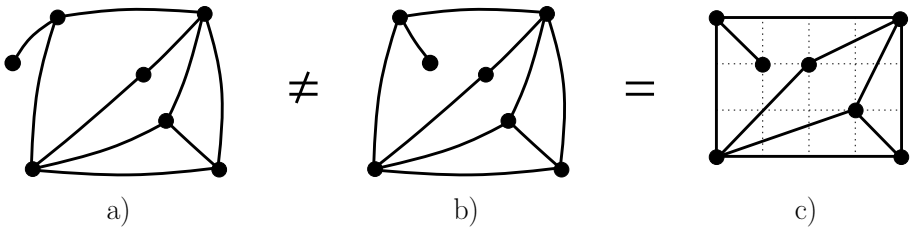


FIGURE 1. Two different planar maps with the same underlying graphs (Fig.a and Fig.b), and a straight-line drawing of the second map (Fig.c).

Maps. A *planar map* (hereafter shortly called a map) is an isotopy class of planar embeddings of a connected planar graph. Notice that the graphs embedded are *unlabelled*. To state it simply, a planar map is a connected unlabelled graph drawn in the plane without edge-crossings and up to continuous deformation. Planar maps are often called *plane graphs* in the literature. As illustrated in Figure 1(a)-(b), a planar graph can have non-isotopic planar embeddings, so that it gives rise to several maps. Due to the topological embedding, a map has more structure than a graph. In particular, a map has *faces* , each face corresponding to a connected component of the plane split by the embedding. The unique unbounded face is called the outer (or infinite) face. Vertices, edges, and faces are called the 0-cells, 1-cells, and 2-cells of the map, respectively. The numbers $|V|$, $|E|$, and $|F|$ of vertices, edges, and faces (including the outer face) of a map are related by the well known Euler's relation:

$$|V| - |E| + |F| = 2. \tag{2}$$

A pair of cells are *incident* to one another if one of the two cells belongs to the closed boundary of the other one, e.g. a vertex v is incident to a face f (and vice versa) if v belongs to the boundary of f. It will be convenient to see an edge e as a pair of half-edges starting at each of the two extremities of e and meeting in the middle of e, i.e., we imagine that each edge is cut at its middle into two half-edges. The face incident to a half-edge h is defined to be the face on the right of h (looking from the origin of h). The *degree* of a vertex v is the number of half-edges incident to v, i.e., having v as origin. If there is no loop nor multiple edges, the degree of v is its number of neighbours. The degree of a face f is the number of edges traversed during a tour of the boundary of f. Finally, let us make precise the terminology and properties concerning cycles and circuits. First, cycles and circuits will always be assumed to be simple when dealing with maps. A theorem of Jordan ensures that a cycle C partitions the plane into two regions, a bounded one called the *interior* of C, and an unbounded one called the *exterior* of C. A vertex or edge will be said to be inside C (outside of C) if it is in the interior of C (exterior of C, respectively). Concerning circuits (i.e., oriented cycles), a circuit $\mathcal{C}$ is said to be *clockwise* (shortly written cw) if the interior of $\mathcal{C}$ is on the right of $\mathcal{C}$ and is said to be *counter-clockwise* (shortly written ccw) otherwise .

Combinatorial encoding of maps. Even if the definition is geometric, planar maps are combinatorial objects. Indeed, a map M is encoded by the so-called *half-edge structure*. Given a half-edge h of M, define the *opposite half-edge* of h —denoted by $\mathrm{opp}(h)$— as the half-edge complementing h into an edge, and define

the *next half-edge* of h —denoted by $\text{next}(h)$— as the half-edge following h in counterclockwise order around the origin of h. If the half-edges of M are given distinct labels in $\{1, \dots, 2n\}$, the map M is completely encoded (up to specifying the outer face) by the two permutations σ_{opp}, which maps the label of each half-edge to the label of the opposite half-edge (hence σ_{opp} has one cycle of length 2 for each edge), and σ_{next}, which maps the label of each half-edge to the label of the next half-edge (hence σ_{next} has a cycle for each vertex). Notice that the permutation $\sigma_{\text{face}} := \sigma_{\text{next}} \circ \sigma_{\text{opp}}$ maps each half-edge h to the following half-edge in clockwise order around the face incident to h. The half-edge encoding is also very convenient to handle maps as data structures; this is the data structure we have used to implement all algorithms presented in this thesis. In practice, each half-edge has a pointer to its opposite half-edge and a pointer to its next half-edge. The half-edge data structure makes it possible to navigate around a vertex, a face, to traverse a map according to various orders...

Rooted maps. Even though the encoding of maps is done via labeling of the half-edges, maps are unlabeled objects. Hence, they are subject to symmetries, which makes a combinatorial treatment more complicated. Precisely, given two maps M_1 and M_2, an isomorphism from M_1 to M_2 is a bijection ϕ mapping the half-edges of M_1 to the half-edges of M_2, preserving the opposite and next half-edge, and preserving the outer face (this last condition is dropped if maps are considered on the sphere rather than in the plane, see [**8**, Chap 0]). In other words, $\text{opp}(\phi(h)) = \phi(\text{opp}(h))$ and $\text{next}(\phi(h)) = \phi(\text{next}(h))$, and ϕ maps the half-edges of the outer face of M_1 to the half-edges of the outer face of M_2. An automorphism of a map M is an isomorphism from M to itself. To avoid problems due to automorphisms, it is more convenient to consider rooted maps instead of maps. A map is *rooted* by marking and orienting an edge so that this edge has the outer face on its left. The marked oriented edge is called the *root* and its origin is called the *root-vertex*. If no outer face is specified (map on the sphere rather then on the plane), the operation of rooting consists in marking a half-edge and choosing the face on the left as outer face, so as to get a rooted planar map. It can be proved that an automorphism $\Phi \neq \text{Id}$ of a map M on the sphere has no fixed half-edge, so that a rooted map has no symmetry. Notice that this property is not true for unlabeled planar graphs, i.e., a planar graph with a marked oriented edge can have nontrivial automorphisms.

Straight-line drawing. Like planar graphs, a planar map can be represented by infinitely many planar embeddings. We will focus in Chapter 5 on a very natural representation of a planar map, called *straight-line drawing*. A planar embedding of a map M is called a straight-line drawing if the vertices are mapped to points of a regular integer grid $[0..W] \times [0..H]$ and edges are mapped to segments, see Figure 1(c). The integers W and H are called the width and the height of the grid. It is well known that a planar map M with no loop nor multiple edge admits a straight-line drawing. Indeed, M can be triangulated by adding edges, and there exist straight-line drawing algorithms for triangulations, as we will see in Chapter 5.

1.2. Families of planar maps. Several families of planar maps will be considered in this thesis. Our aim is to have a good understanding of the combinatorial properties of these families, in order to develop an efficient algorithmics. Classical families of maps are obtained by imposing a regularity on the degrees of vertices or faces, or by imposing a higher connectivity. Much attention will be devoted to triangulated maps (all inner faces are triangles) and quadrangulated maps (all

inner faces are quadrangles). These families are historically important, as triangulations are maximal planar graphs, and quadrangulations are maximal bipartite planar graphs. Hence, these two families often capture the difficulty of conjectures or algorithmic tasks on planar graphs and bipartite planar graphs.

Trees. A *plane tree* is a map with only one face, the outer face. Families of trees with refined conditions will be considered in Chapter 3.

Eulerian maps. A map is called *Eulerian* iff all its vertices have even degree. Loops and multiple edges are allowed.

Triangulated maps. A *triangulated map* is a loopless map such that all faces have degree 3. A *triangulation* is a map with no loop nor multiple edge and such that all faces have degree 3. Triangulations are also called maximal plane graphs because any edge addition breaks planarity. A *triangulation of the k-gon* is a map with no loop nor multiple edge and with an outer face of degree k and all inner faces of degree 3. A triangulation T of the k-gon is said to be *irreducible* if the interior of any 3-cycle is a face, i.e, there is no filled 3-cycle. Notice that $k > 3$ unless T is reduced to the triangle. A lot of attention in this thesis will be devoted to irreducible triangulations of the 4-gon, which we shortly call irreducible triangulations.

Bicolored maps. *Bipartite maps* are maps whose vertices can be partitioned into black and white vertices, with the property that any pair of adjacent vertices have different colors. A bipartite map endowed with such a bicoloration is called a *bicolored map*. It is easily shown that bipartite maps are exactly maps with all faces of even degree. Moreover, the bicoloration of vertices is unique up to the color of the first vertex. Rooted bipartite maps will always be endowed with their unique bicoloration such that the origin of the root is black.

Quadrangulated maps. *Quadrangulated maps* are maps with all faces of degree 4. Notice that such maps are bipartite, so that they have no loops. *Quadrangulations* are quadrangulated maps with no multiple edges. Quadrangulations are also called maximal bipartite plane graphs because any edge addition either breaks bipartiteness or planarity. A *separating 4-cycle* of a quadrangulation is a 4-cycle C such that neither the interior nor the exterior of C is a face. A quadrangulation is called *irreducible* if it has no separating 4-cycle. For $k \geq 3$, a quadrangulated dissection of the $2k$-gon —shortly called dissection— is a map with outer face of degree $2k$ and all inner faces of degree 4. A dissection is said to be *irreducible* if the interior of any 4-cycle is a face. Notice that the definition is simpler than for quadrangulations, since the outer cycle is not a 4-cycle (hence, the exterior of a 4-cycle can not be a face). A lot of attention will be devoted to irreducible dissections of the hexagon, which are of great interest for the algorithmic treatment of 3-connected maps.

k-connected maps. A map is called k-connected if at least k vertices (and their incident edges) have to be removed to disconnect the map; in addition the map must have at least k vertices, loops are forbidden if $k > 1$, and multiple edges are forbidden if $k > 2$. It is easily shown that a quadrangulation is 2-connected, a triangulation is 3-connected, and a triangulation is 4-connected iff the interior of any 3-cycle—except the outer one—is a face. Notice that the characterisation of 4-connected triangulations is very close to the condition of irreducibility, except that

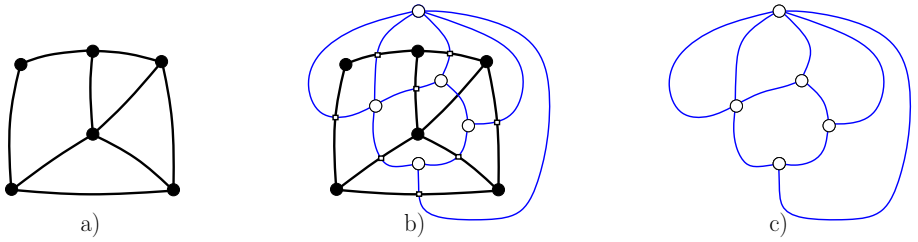


FIGURE 2. A planar map (Fig.a), its derived map (Fig.b), and its dual map (Fig.c).

the outer 3-cycle can be filled in a 4-connected triangulation. This link will be used in Chapter 3 to study 4-connected triangulations from irreducible triangulations.

1.3. Correspondences between families of maps. There exist several simple correspondences between families of maps. We recall here the duality mapping and the angular mapping, which will be intensively used.

1.3.1. *Duality.* The dual of a map M is the map M^* having one vertex v_f in each face f of M, two vertices v_f and $v_{f'}$ being adjacent in M^* iff the faces f and f' are adjacent in M, i.e., there exists an edge having f and f' on each side; see Figure 2 for an example. Each edge $e \in M$ gives rise to exactly one edge $e^* \in M^*$, which crosses e at a vertex v_e. The map obtained by superimposing M and M^* is called the *derived map* of M. The vertices of M' corresponding to intersections of an edge with its dual edge are called *edge-vertices*. Clearly each edge-vertex has degree 4 and each edge of the derived map either corresponds to a half-edge of M —these are called *primal edges* of M'— or corresponds to a half-edge of M^* —these are called *dual edges* of M'. A vertex of M' is said to be *primal* (*dual*) if it belongs to M (to M^*, respectively). The duality exchanges the number of vertices and faces, i.e., the dual of a map with i vertices, n edges, and j faces, is a map with j vertices, n edges, and i faces. It is also easily checked that the duality is involutive, i.e., $(M^*)^* = M$. If maps are rooted, the root of M^* is the dual edge of the root edge e of M, oriented from the left of e to the right of e. In its rooted version, the duality is involutive up to the orientation of the root. A nice property of duality is to preserve the connectivity degree up to order 3. The properties are summarized in the following theorem.

THEOREM 1.1 (Duality). *The families of connected, 2-connected, and 3-connected maps are stable under duality. The duality mapping is involutive and the parameters of a map M and of its dual map M^* satisfy the following correspondences:*

$$\begin{array}{rcl} \textit{vertex of } M & \leftrightarrow & \textit{face of } M^*, \\ \textit{face of } M & \leftrightarrow & \textit{vertex of } M^*, \\ \textit{edge of } M & \leftrightarrow & \textit{edge of } M^*. \end{array}$$

1.3.2. *Angular mapping.* There is a well known correspondence, due to Tutte, between quadrangular maps and unconstrained maps, which we call the *angular mapping*. Given a bicolored quadrangulated map Q, the *primal map* of Q is the map M whose vertices are the black vertices of Q, two black vertices being adjacent in M iff they are incident to the same face of Q, see Figure 3. Notice that the same construction with white vertices instead of black vertices would give the dual map of M. Clearly, there is an edge of M for each face of Q. Conversely, Q is called

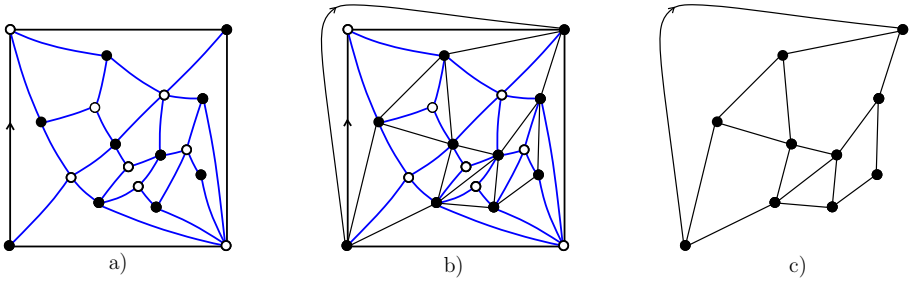


FIGURE 3. A rooted quadrangulation Q (Fig.a), addition of the black diagonals (Fig.b), and the primal map of Q (Fig.c).

the *angular map* of M, because each edge of Q corresponds to an angle of M. If Q is rooted, then the primal map also receives a root, which is chosen to be the edge following the root of Q in counterclockwise order around the root vertex of Q, see Figure 3. The angular mapping has a nice restriction to 2-connected and 3-connected maps. Indeed, the presence of a multiple edge in Q is equivalent to the presence of a separating vertex in M, and the presence of a separating 4-cycle in Q is equivalent to the presence of a separating pair of vertices in M [**90**]. The properties of the angular mapping are summarized in the following theorem.

THEOREM 1.2 (Angular mapping). *The angular mapping is a bijection between the following families of maps:*

$$\begin{aligned} \textit{bicolored quadrangulated maps} &\leftrightarrow \textit{maps,}\\ \textit{bicolored quadrangulations} &\leftrightarrow \textit{2-connected maps,}\\ \textit{bicolored irreducible quadrangulations} &\leftrightarrow \textit{3-connected maps.} \end{aligned}$$

In its rooted version, the angular mapping is a bijection between the following families of rooted maps:

$$\begin{aligned} \textit{rooted quadranulated maps} &\leftrightarrow \textit{rooted maps,}\\ \textit{rooted quadrangulations} &\leftrightarrow \textit{rooted 2-connected maps,}\\ \textit{rooted irreducible quadrangulations} &\leftrightarrow \textit{rooted 3-connected maps.} \end{aligned}$$

The parameters of a quadrangulated map Q and of its primal map M satisfy the following correspondences:

$$\begin{aligned} \textit{edge of } Q &\leftrightarrow \textit{angle of } M,\\ \textit{black vertex of } Q &\leftrightarrow \textit{vertex of } M,\\ \textit{white vertex of } Q &\leftrightarrow \textit{face of } M,\\ \textit{face of } Q &\leftrightarrow \textit{edge of } M. \end{aligned}$$

2. The theory of α-orientations

Orientations with prescribed outdegree, the so-called *α-orientations*, are at the heart of all combinatorial structures to be presented later. A very elegant theory of these orientations has been developed by Propp [**96**], Ossona de Mendez [**92**], and Felsner [**48**]. The ideas can also be traced back to an earlier paper by Kuhler et al on the lattice properties of circulations in planar graphs [**80**]. This section recalls the main properties to be used later, in particular the lattice rules. First, let us recall the definition of distributive lattices.

Distributive lattices. A *lattice* is a partially ordered set $(E, \leq)$ such that, for each pair (x, y) of elements of E, there exists a unique element $x \wedge y$ and a unique element $x \vee y$ satisfying the conditions:

- $x \wedge y \leq x$, $x \wedge y \leq y$, and $\forall z \in E$, $z \leq x$ and $z \leq y$ implies $z \leq x \wedge y$,
- $x \vee y \geq x$, $x \vee y \geq y$, and $\forall z \in E$, $z \geq x$ and $z \geq y$ implies $z \geq x \vee y$.

In other words, each pair admits a unique common lower element dominating all other common lower elements, and the same holds with common upper elements. The lattice is said to be *distributive* if the operators $\wedge$ and $\vee$ are distributive with respect to each other, i.e., $\forall(x, y, z)$, $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ and $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$. We define a flip (flop) as the operation of moving from an element of the lattice to a lower (upper, respectively) covering element. The nice feature of distributive lattices is that, in most cases, moving from an element of the lattice to a covering lower (upper) element has a simple geometric interpretation, which we informally call a flip (flop, respectively). As we recall next, in the case of orientations of a plane graph with prescribed vertex outdegrees, the flip (flop) operation consists in reversing a clockwise circuit (counter-clockwise circuit, respectively).

Definition of α-orientations. Given a planar map $M = (V, E)$ and a function $\alpha : V \to \mathbb{N}$, an α-orientation is an orientation of the edges of M such that each vertex v has outdegree $\alpha(v)$. A function $\alpha : V \to \mathbb{N}$ such that M has at least one α-orientation is called *feasible*.

Conditions of existence. Given a function $\alpha : V \to \mathbb{N}$, a necessary condition of existence of an α-orientation is clearly $\sum_{v \in V} \alpha(v) = |E|$. There is a refinement of this condition to all subsets of V. For $X \subset V$, write $\delta(X, \overline{X})$ for the set of edges connecting a vertex in X and a vertex in $\overline{X} := V \backslash X$, and write $E[X]$ for the set of edges connecting two vertices in X. Then, in any α-orientation of M, the number of edges going from X to $\overline{X}$ has to be equal to $\mathrm{Out}_X := \sum_{v \in X} \alpha(v) - |E[X]|$. The *cut condition* states that, for any subset $X \subset V$, $0 \leq \mathrm{Out}_X \leq |\delta(X, \overline{X})|$. Clearly, the cut condition is a necessary condition of existence of an α-orientation. It is also a sufficient condition, as proved by induction on the number of edges [**48**].

Invariants. Given a planar map $M = (V, E)$ and a fixed feasible function $\alpha : V \to \mathbb{N}$ for M, there are several properties shared by all α-orientations of M. We have seen that the number of edges going out of a subset $X \subset V$ does not depend on the α-orientation, and satisfies $0 \leq \mathrm{Out}_X \leq |\delta(X, \overline{X})|$.

In the two extremal cases $\mathrm{Out}_X = 0$ and $\mathrm{Out}_X = |\delta(X, \overline{X})|$, the edge cut (the set of edges connecting X and $\overline{X}$) is called *rigid*. Clearly an edge belonging to a rigid edge-cut, called a *rigid edge*, has the same orientation in all α-orientations of M. Based on the cut condition, it is easily shown that the accessibility from a vertex v to a vertex v' does not depend on the α-orientation. Hence the decomposition of M into strongly connected components does not depend on the α-orientation: the rigid edges are the edges connecting different strongly connected components, and the non-rigid edges are the edges inside a strongly connected component. In particular, any non-rigid edge e belongs to an oriented circuit $\mathcal{C}$. Reversing $\mathcal{C}$ yields another α-orientation where e has the reverse direction. Circuit reversion is the fundamental operation to navigate in the set of α-orientations of a planar map.

Essential circuits. Consider a planar map M endowed with an α-orientation. A circuit $\mathcal{C}$ is called *chordal* if there exists an oriented path $\mathcal{P}$ of edges inside $\mathcal{C}$

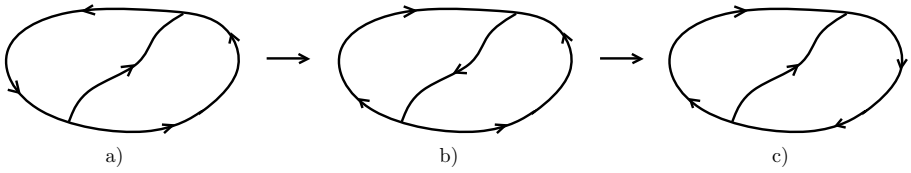


FIGURE 4. Reversing an oriented chordal circuit in two steps.

such that the two extremities of $\mathcal{P}$ are on $\mathcal{C}$. As illustrated in Figure 4, reversing a chordal circuit can be done by reversing successively two smaller oriented cicuits, in the sense that the regions enclosed by these circuits are strictly included in the region enclosed by $\mathcal{C}$. The circuits that are not chordal are called *essential circuits*. Hence, a circuit reversion can always be performed as a sequence of reversions of essential circuits. Notice that the status (essential or chordal) of a circuit $\mathcal{C}$ does not depend on the α-orientation. Indeed, $\mathcal{C}$ is essential iff it is chordless and no vertex inside $\mathcal{C}$ is in the same strongly connected component as $\mathcal{C}$. A cycle C appearing as an essential circuit in at least one α-orientation of M is called an *essential cycle*.

Definition. A flip (flop) is the operation of reversing a clockwise (counterclockwise, respectively) circuit.

THEOREM 1.3 (Lattice property [**48, 92, 96**]). *Given a planar map M and a feasible function $\alpha : V \to \mathbb{N}$, the set of α-orientations of M endowed with the flip-flop operations is a distributive lattice.*

Proof (Sketch). The idea is to associate a potential-vector to each α-orientation, so that the corresponding set of vectors is stable under minimum and maximum, hence is a distributive lattice for the min-max partial order. Define $\mathcal{E}$ as the set of essential cycles of M for the function $\alpha : V \to \mathbb{N}$. Starting from an α-orientation X, it can be shown that the number of times an essential C cycle is flipped in a maximal flip-sequence is a finite number $N_X(C)$ not depending on the maximal flip-sequence [**48**]. As the number of essential cycles is finite, any flip-sequence terminates at an α-orientation X_0, which has no clockwise circuit. In particular, this proves the existence of an α-orientation with no clockwise circuit. The uniqueness of such an orientation is also easily proved [**48**, Lem.1]. Then, the lattice structure relies on the following property; for two α-orientations X and X', there exists an α-orientation $X \wedge X'$ and an α-orientation $X \vee X'$ such that $N_{X\wedge X'}(C) = \min(N_X(C), N_{X'}(C))$ and $N_{X\vee X'}(C) = \max(N_X(C), N_{X'}(C))$ $\forall C \in \mathcal{E}$. In other words, α-orientations of M are in bijection with a set of vectors of $\mathbb{N}^{|\mathcal{E}|}$ stable under minimum-maximum. The fact that such a vector-set is a distributive lattice is a folklore result. Finally it easily follows from the definition that the cover relation in the set of potential vectors corresponds to reversing the orientation of an essential circuit. □

Applications. As discussed by Felsner [**48**], distributive lattices are well understood structures. Let us cite the fundamental theorem [**105**]: "a distributive lattice can be represented as the set of downward ideals of a partially ordered set". This theorem is the formal general statement explaining why it is mostly easy to navigate in a distributive lattice using simple local operations. Accordingly, efficient algorithms have been developed for uniform random generation [**97**] (using coupling

from the past techniques) and exhaustive generation. An application of exhaustive generation to optimize a result of cartography will be given in Section 6.

The next sections are devoted to the presentation of combinatorial structures that characterise several families of planar maps, and have a natural formulation in terms of α-orientations. Let us mention a first example (not developed in this thesis); the set of spanning trees of a fixed rooted planar map M is in bijection with the set of α-orientations of the derived map M', for the function α equal to 1 for non-root primal and dual vertices, equal to 0 for the root vertex and v_∞ (the dual vertex corresponding to the infinite face), and equal to 3 for edge-vertices. The bijection makes use of the fact that the edges dual to edges not in the spanning tree form a spanning tree of the dual map. Propp [**96**] and Felsner [**48**] characterise the flip operation directly on spanning trees.

3. Eulerian orientations

The first example of structure we investigate are Eulerian orientations. Let us first recall the definition of *Eulerian circuits*, which represent one of the first historical developments in graph theory. Given a graph $G = (V, E)$, an Eulerian circuit $\mathcal{C}$ of G is a circuit traversing each edge of G exactly once. Eulerian circuits (and the similar notion of Eulerian path) have been first considered by Euler to give a negative answer to the classical Königsberg problem (whether all the bridges of the city of Königsberg can be traversed exactly once). Notice that the orientation of G induced by an Eulerian circuit is such that each vertex has same indegree as outdegree, in particular each vertex has even degree. A graph G with all vertices of even degree is called an *Eulerian graph*, and an orientation of the edges of G such that $\mathrm{Indeg}(v) = \mathrm{Outdeg}(v)$ is called an *Eulerian orientation*.

Proposition 1.1 (Characterization (Folklore)). *A graph admits an Eulerian orientation iff it is Eulerian.*

Proof. Clearly, a graph endowed with an Eulerian orientation is Eulerian. Conversely, if a graph G is Eulerian, there exist classical algorithms to compute an Eulerian circuit for each connected component of G [**54**], inducing an Eulerian orientation. In fact, it is even simpler to compute directly an Eulerian orientation in a greedy way [**103**, Vol A, Sect6.4]. □

It can be shown that the problem of counting the number of Eulerian orientations of a fixed Eulerian graph is difficult, namely is #P-complete [**87**]. Nevertheless, as observed in [**96, 48**], the set $\mathcal{O}$ of Eulerian orientations of a fixed *Eulerian map* has an explicit combinatorial structure.

Proposition 1.2 (Lattice structure [**48, 96**]). *Given an Eulerian map M, the set of Eulerian orientations of M is a distributive lattice. The flip operation consists in reversing the orientation of a clockwise circuit delimiting a face.*

Proof. The set of Eulerian orientations of M is exactly the set of α-orientations of M, for the function $\alpha(v) = \deg(v)/2$. Clearly each edge-cut $\delta(X, \overline{X})$ has even cardinality and has the same number of edges (non zero by connectivity of M) from X to $\overline{X}$ as from $\overline{X}$ to X. Hence, there are no rigid edges, so that an essential circuit is always the contour of a face. □

As there are no rigid edges, Eulerian orientations can be considered as the simplest example of α-orientations. An example of distributive lattice formed by

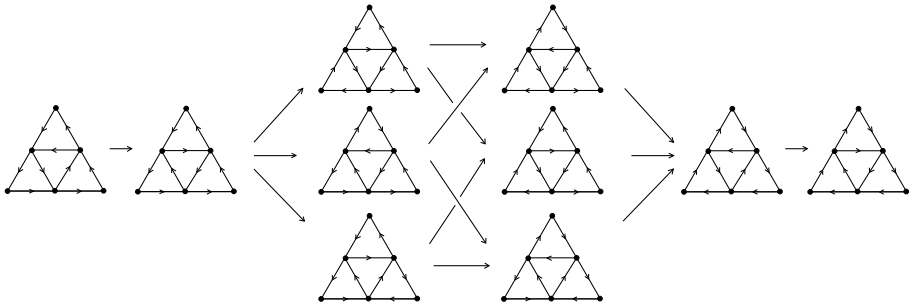


FIGURE 5. The distributive lattice of Eulerian orientations of an Eulerian map.

the Eulerian orientations of an Eulerian map is shown in Figure 5: each arrow of the (Hasse) diagram corresponds to reversing a ccw face boundary (flop operation).
Definition. Given an eulerian map, the eulerian orientation at the bottom of the distributive lattice is called the *minimal eulerian orientation*.

4. Bipolar orientations

We consider in this section bipolar orientations, a powerful combinatorial structure that proves insightful to solve many algorithmic problems such as planar graph embedding [**31, 82**] and geometric representations of graphs in various flavours (visibility [**107**], floor planning [**78**], straight-line drawing [**108**]). We first recall basic properties of bipolar orientations, and then focus on the planar case. Using a bijection between bipolar orientations and some orientations of the angular map, we give a precise description of the lattice of bipolar orientations of a fixed map. Our study closely follows the thesis of Ossona de Mendez [**92**].

Definition. Given a graph $G = (V, E)$, a *bipolar orientation* of G is an acyclic orientation of G such that the induced partial order on V has a unique minimum s, called the *source*, and a unique maximum t, called the *sink*.

4.1. Properties and characterization. A rich source of results on bipolar orientations is an article of De Fraysseix *et al* [**38**]. Among the classical properties, it is shown that for any vertex v, there exists a simple oriented path from s to t passing by v. To state the condition of existence of a bipolar orientation, it is convenient to consider *rooted graphs*: a graph is rooted by marking and directing one of its edges, called the *root*. In this framework, a *bipolar orientation* is always such that the origin of the root is the source and the end-vertex of the root is the sink.

PROPOSITION 1.3 (Characterization [**38**]). *A rooted graph $G = (V, E)$ admits a rooted bipolar orientation iff G is 2-connected.*

Proof. Let G be a rooted graph admitting a bipolar orientation. Assume there exists a separating vertex v. If $v = s$, there is clearly at least one sink in each component of $G\backslash\{s\}$, contradicting the uniqueness of the sink; similarly v can not be equal to t, i.e., $v \in G\backslash\{s, t\}$. As the source s and the sink t are adjacent, they are in the same connected component of $G\backslash\{v\}$. Hence, there exists a connected component G_1 of $G\backslash\{v\}$ that contains neither s nor t. Let w be a vertex in G_1. As

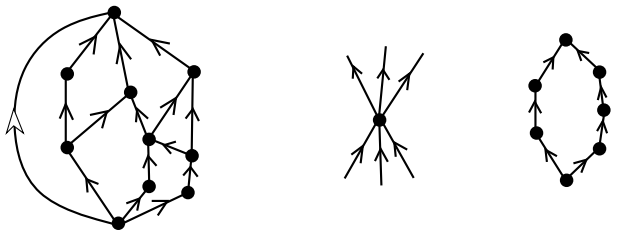


FIGURE 6. A bipolar orientation of a rooted 2-connected map (Fig.a), the configurations of a vertex and of a face (Fig.b).

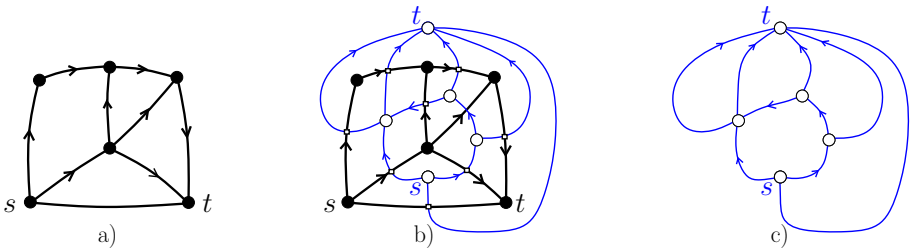


FIGURE 7. The duality relation on bipolar orientations

mentioned above, there exists a simple path from s to t passing by w. Clearly this path has to pass twice by v, once from s to w and once from w to t. This contradicts the fact that the path is simple. Conversely, given a rooted 2-connected graph G, there exist iterative algorithms to compute a bipolar orientation of G [**44**]. In the planar case, there exists a simplified algorithm due to De Fraysseix *et al* [**36**], which is recalled in Chapter 2 (Section 3). □

4.2. Bipolar orientations in the plane. We focus here on the properties of bipolar orientations embedded in the plane. When considering rooted maps, bipolar orientations will always be assumed to have the origin of the root as source and the end-vertex of the root as sink, see Figure 8(a). Such bipolar orientations are called *plane bipolar orientations*. According to Proposition 1.3, the family of rooted maps that admit a bipolar orientation is the family of rooted 2-connected maps. In other words, the combinatorial structure "bipolar orientation" characterizes the family of rooted 2-connected maps.

Duality relation. A nice property of bipolar orientations is the duality relation. Given a rooted map M endowed with a rooted bipolar orientation, orient the edges of M^* from right to left (each edge e^* dual to an edge $e \in M$ is oriented from the right of e to the left of e), except for the root edge and its dual whose directions are not considered, see Figure 7. Clearly, the duality relation is anti-involutive: applying the operation twice yields the reverse bipolar orientation.

Counting plane bipolar orientations. Given a pair (i, j) of integers, let θ_{ij} be the total number of rooted plane bipolar orientations on rooted maps with $i + 1$ vertices and $j + 1$ faces. A first remark is that $\theta_{ij} = \theta_{ji}$, a consequence of the

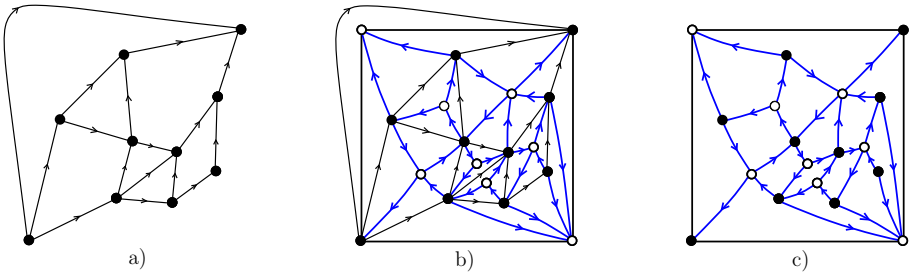


FIGURE 8. A rooted 2-connected map endowed with a bipolar orientation (a), the angular map of M endowed with the associated angular orientation (c).

duality relation. Precisely, θ_{ij} satisfies the following exact formula:

$$\theta_{ij} = 2\frac{(i+j-2)!(i+j-1)!(i+j)!}{(i-1)!i!(i+1)!(j-1)!j!(j+1)!}. \tag{3}$$

Formula (3) has first been discovered by Baxter [**4**, Eq 5.3] using technical guess-and-check calculations on the Tutte polynomial. In a subsequent article, a direct computation has been given by Bousquet-Mélou [**20**], based on the so-called obstinate Kernel method. In collaboration with Gilles Schaeffer and Dominique Poulalhon, I have obtained a direct bijective proof (not presented in this thesis) of this formula [**65**], using a construction that draws much of its inspiration from a nice bijective proof of Nicolas Bonichon for the total number of Schnyder woods (defined in Section 5) on rooted triangulations with n vertices [**15**]. Our bijection also makes use of a formulation of plane bipolar orientations as α-orientations on the associated angular map, which is described next.

Formulation as α-orientations on the angular map. Given a bipolar orientation X of M with source s and sink t, an angle (e, e') formed by two edges of M around a vertex v —with e' following e in cw order around v— is called an *extremal angle* if (e, e') are (ingoing, ingoing) or (outgoing, outgoing), and is called a *lateral angle* if (e, e') are (ingoing, outgoing) or (outgoing, ingoing). Accordingly, the *inner edges* of the angular map Q are partitioned into so-called *extremal edges* and *lateral edges*, depending on the type of their associated angle. We orient the lateral edges outward of their incident black vertex, and the extremal edges outward of their incident white vertex. This orientation is called the *angular orientation* of Q associated with X, see Figure 8(b).

Plane bipolar orientations satisfy the following important property, illustrated in Figure 6(b). The edges incident to an inner vertex are partitioned into a non-empty interval of ingoing edges and a non-empty interval of outgoing edges; and, dually, each face f of M has two particular vertices s_f and t_f such that the contour of f consists of two non-empty oriented paths both going from s_f to t_f, called *left lateral path* and *right lateral path* of f, respectively. Hence, each inner black vertex v of Q has two outgoing edges, corresponding to the two lateral angles incident to v. Dually, each white vertex of Q, corresponding to a face f of M, has two outgoing edges, corresponding to the two extremal angles incident to the face f. Notice also that all edges of Q incident to an outer vertex v are oriented toward v.

Definition. Given a quadrangulation Q, an orientation of the inner edges of Q such that all inner vertices have outdegree 2 and the four outer vertices have outdegree 0 is called a *2-orientation* (the four outer edges are not oriented).

PROPOSITION 1.4 ([**38**]). *Let M be a rooted 2-connected map and let Q be the angular map of M. The set of bipolar orientations of M is in bijection with the set of 2-orientations of Q.*

Proof. As discussed above, a bipolar orientation X of M gives rise to a 2-orientation $\Phi(X)$ of Q, obtained by orienting the inner edges of Q corresponding to extremal angles (lateral angles) toward (outward, respectively) the corresponding black vertex of Q. Conversely, using Euler's relation on 2-orientations, it can be shown that a 2-orientation Y gives rise to a unique bipolar orientation $X := \Psi(Y)$ of M, in such a way that the mappings Ψ and Φ are mutually inverse [**38**]. □

As any vertex of a bipolar orientation is on a path connecting the two poles, 2-orientations satisfy the following *accessibility property*.

Accessibility property: "For any inner vertex v of a quadrangulation endowed with a 2-orientation, there is at least one outer vertex accessible from v".

4.3. Lattice structure of plane bipolar orientations. According to Theorem 1.3, the set $\mathcal{O}$ of 2-orientations of a fixed quadrangulation Q is a distributive lattice. Hence, the set $\mathcal{B}$ of bipolar orientations of a fixed rooted 2-connected map M is a distributive lattice, via the bijection with the 2-orientations of the angular map Q of M (Proposition 1.4). By studying the essential circuits of a 2-orientation, we are able to obtain a direct simple formulation of the flip operation on bipolar orientations, see Proposition 1.5. This makes the lattice structure of $\mathcal{B}$ more explicit than a formulation via circuit reversion in the 2-orientations of the angular map Q. Our presentation is very sketchy, as the detailed study is due to Ossona de Mendez [**92**]. Nevertheless, we find interesting to give the ideas, as they will appear again to describe the lattice property of transversal structures in Section 6.

Essential circuits of 2-orientations. Given a quadrangulation Q endowed with a 2-orientation O, it is easily shown that the essential circuits of O are exactly the circuits of length 4. The proof goes in two steps. 1) No edge e inside an essential circuit $\mathcal{C}$ has its origin on $\mathcal{C}$, otherwise e could be extended to an internal path of $\mathcal{C}$ (using the accessibility property), contradicting the fact that $\mathcal{C}$ is essential. 2) Let n be the number of vertices inside $\mathcal{C}$. As no edge inside $\mathcal{C}$ has its origin on $\mathcal{C}$, the number of edges inside $\mathcal{C}$ is equal to $2n$. Moreover, if $2k$ denotes the length of $\mathcal{C}$, Euler's relation ensures that the number of edges inside $\mathcal{C}$ is equal to $2n + k - 2$. Hence $k = 2$.

2-pieces in 2-connected maps. Let $M = (V, E)$ be a rooted 2-connected map. Given a subset $F \subset E$ of edges of M, we denote by $M(F)$ the embedded graph (possibly made of several connected components) formed by the edges in F and their incident vertices, and by $M(E\backslash F)$ the embedded graph formed by the edges in $E\backslash F$ and their incident vertices. A non-empty subset $F \subset E$ not containing the root edge is called a *2-piece* if $M(F)$ is contained in one face of $M(E\backslash F)$, and $M(F)$ and $M(E\backslash F)$ intersect at exactly two vertices $\{v, w\}$, which are called the *poles* of the 2-piece. The connectivity of M implies that $M(F)$ and $M(E\backslash F)$ are connected, i.e., are planar maps. It is easily shown that the 2-pieces of M are in

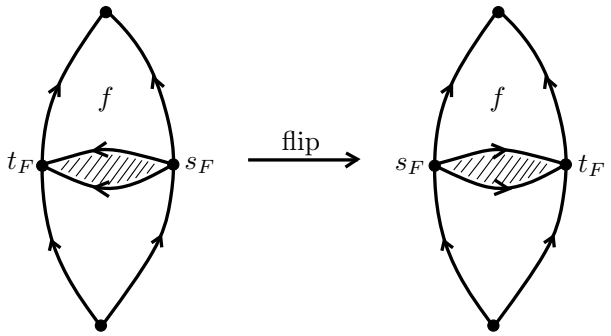


FIGURE 9. The flip operation on bipolar orientations.

bijection with the 4-cycles of the angular map Q (for any 4-cycle C of Q, associate the set of edges of M that are inside C).

Transversal 2-pieces. We explain now how the bijection between 2-pieces and 4-cycles is lifted so as to take the orientations into account. Consider a bipolar orientation X of M, and let Y be the corresponding 2-orientation on the angular map Q. Given a 2-piece F of M, the induced orientation $X(E\backslash F)$ on $E\backslash F$ is a bipolar orientation with same source and sink as M; and the induced orientation on F is a bipolar orientation $X(F)$ whose poles $\{s_F, t_F\}$ are the two poles of the 2-piece. A 2-piece is called *transversal* if s_F and t_F are not comparable in the bipolar orientation $X(E\backslash F)$. Hence, there exists a face f of $X(E\backslash F)$ such that one pole is in the (open) left lateral path of f and the other pole is in the (open) right lateral path of f, see Figure 9. The transversal 2-piece is said to be *left-oriented* (*right-oriented*) if t_F is in the left lateral path (right lateral path, respectively) of f.

Bijection between transversal 2-pieces and oriented 4-circuits. The bijection between 2-pieces of M and 4-cycles of Q specializes as follows. Let M be a rooted 2-connected map endowed with a bipolar orientation X and let Y be the corresponding 2-orientation of the angular map Q of M. Then the clockwise (counterclockwise) essential circuits of Q are in bijection with the left-oriented (right-oriented, respectively) transversal 2-pieces of M. Recall that a flip operation on the 2-orientations of Q consists in reversing the orientation of an essential clockwise circuit, i.e., a clockwise circuit of length 4. Thanks to the bijection between transversal 2-pieces and 4-circuits, we can formulate the flip operation directly on bipolar orientations, see Figure 9:

PROPOSITION 1.5 (Lattice structure [**92**]). *The set $\mathcal{B}$ of (rooted) bipolar orientations of a rooted 2-connected map M is a distributive lattice. The flip operation consists in reversing the orientations of edges in a left-oriented transversal 2-piece, thus making the 2-piece right-oriented.*

Definition. Given a rooted 2-connected map M, the bipolar orientation that has no left-oriented transversal 2-piece is called the *minimal* bipolar orientation of M. This is the bipolar orientation associated with the minimal 2-orientation of the angular map (the one with no clockwise circuit).

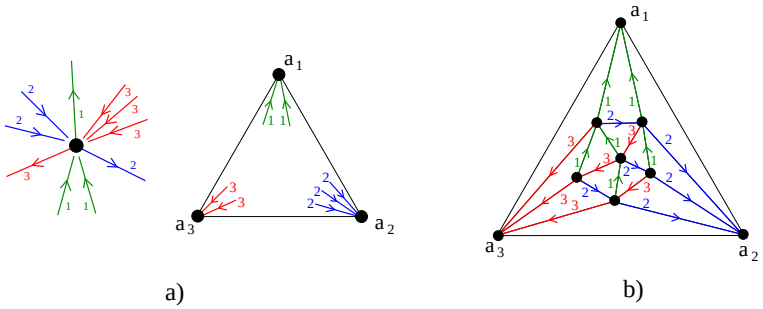


FIGURE 10. Local conditions of Schnyder woods (Fig.a) and an example of Schnyder wood of a triangulation (Fig.b).

5. Schnyder woods

W. Schnyder has introduced in [**101**] a beautiful combinatorial structure for triangulations, which he called *realizer*, and is now usually called *Schnyder wood*. Several equivalent definitions have been given in terms of angle labelings, orientations, or colorings. The concept has also a natural extension to 3-connected maps [**3, 46**]. Essentially a Schnyder wood is an orientation and partition of the inner edges of a triangulation into three trees that are directed to each of the three outer vertices. The definition for 3-connected maps is similar, expect that an edge is allowed to be traversed by either one tree or by two trees traversing the edge in opposite directions. Originally, Schnyder woods have been introduced to obtain a nice characterization of planarity in terms of a "dimension" parameter associated with the incidences vertices-edges [**101**]. These properties have beautiful geometric interpretations in terms of orthogonal surfaces in the 3D space, see [**47, 88, 119**]. Schnyder woods also lead to an elegant straight-line drawing algorithm based on barycentric representation of vertices, also due to Schnyder [**102**] and subsequently improved by Zhang and He [**118**] and by Bonichon *et al* [**16**] (see Chapter 5 for a presentation and analysis of the latter algorithm). In this section we define Schnyder woods and state their main properties, in particular the decomposition into three spanning trees. We also describe the lattice structure, following the study by Brehm [**25**] for triangulations and Felsner [**48**] for 3-connected maps.

5.1. Schnyder woods on triangulations. Let M be a planar map with a triangular outer face, the outer vertices in clockwise order being denoted by a_1, a_2, a_3. A *Schnyder wood* of M is an orientation and labeling, with labels in $\{1, 2, 3\}$, of the inner edges of M such that the following conditions are satisfied (see Figure 10):

Schnyder woods (definition with edges):

- **C1:** Each inner vertex v of T has outdegree 1 in each color. The outgoing edges e_1, e_2, e_3 in each label $\{1, 2, 3\}$ occur in clockwise order around v. For $i \in \{1, 2, 3\}$, all edges entering v with label i are in the clockwise sector between e_{i+1} and e_{i-1}.
- **C2:** For $i \in \{1, 2, 3\}$, all inner edges incident to a_i are ingoing and have label i.

PROPOSITION 1.6 (Characterisation). *A planar map M with triangular outer face admits a Schnyder wood iff M is a triangulation.*

Proof. Let M be a map with triangular outer face admitting a Schnyder wood. Let n be the number of inner vertices of M. The conditions C1 and C2 imply that M has $3n$ inner edges. Hence, M has $n+3$ vertices and $3n+3$ edges. Euler's relation thus ensures that M has $2n+2$ faces, so that the average degree of a face is 3. It is easily checked (also based on Euler's relation) that M has no loop nor multiple edge. Hence all faces of M have degree at least 3. As the average degree of the faces is 3, all faces must have degree 3, hence M is a triangulation.

Conversely, if M is a triangulation, there exist algorithms to compute a Schnyder wood of M. Schnyder has given an algorithm based on edge-contraction [**102**], which has been reformulated by Brehm as a simple iterative procedure [**25**]. We recall the algorithm of Brehm in Chapter 2 (Section 4.1). □

Acyclicity property. Given a triangulation T endowed with a Schnyder wood, let T_1, T_2, T_3 denote the graphs formed by the edges with labels 1, 2, 3, respectively. Then Euler's relation and the Schnyder conditions imply the following acyclicity property [**48**].

Acyclicity property: "for $i = \{1, 2, 3\}$, the graph obtained from T by reversing the edges with label i is acyclic".

The acyclicity property yields two key results on Schnyder woods:

(1) **Partition into 3 spanning trees.** For $i \in \{1, 2, 3\}$, the graph T_i is a spanning tree of $M\backslash\{a_{i-1}, a_{i+1}\}$ rooted at a_i and oriented toward a_i.
(2) **Path property.** For each inner vertex v of T and $i \in \{1, 2, 3\}$, let $P_i(v)$ be the path of T_i from v to its root a_i. Then the paths $P_1(v)$, $P_2(v)$, and $P_3(v)$ do not intersect, except at their origin v.

The second result makes it possible to associate with each inner vertex v three coordinates $f_1(v), f_2(v), f_3(v)$. This leads to embedding algorithms for triangulations and 3-connected maps both in the context of orthogonal surfaces [**88, 47, 119**] and straight-line drawing [**102, 46, 3, 16**].

Definition. We call a *3-orientation* of a triangulation T an orientation of the inner edges of T such that each inner vertex has outdegree 3 and the outer vertices have outdegree 0.

Clearly, the orientation given by a Schnyder wood (forgetting the labels) is a 3-orientation. Conversely, for a 3-orientation, there is exactly one way to give labels so as to satisfy the Schnyder conditions. Uniqueness is clear, as there is no choice in propagating the labels from a_i. Existence is established by associating to each inner edge e an acyclic path starting at e. This path ends at one outer vertex a_i, whose index is chosen to be the label given to e, see [**25, 48**] for details. Thus, by existence and uniqueness of the labeling, Schnyder woods of T are in bijection with 3-orientations of T, so that Theorem 1.3 ensures that the set of Schnyder woods of a fixed triangulation is a distributive lattice. The following proposition gives a precise description of the lattice property, see Figure 11.

PROPOSITION 1.7 (Lattice structure [**25, 92**]). *For any fixed triangulation T, the set of Schnyder woods of T is a distributive lattice. The flip operation consists in reversing the orientation of a clockwise circuit Δ of length 3, and applying $i \to i+1$ to the labels of the edges on Δ and $i \to i-1$ to the labels of the edges inside Δ.*

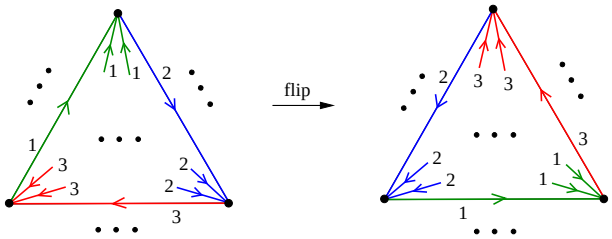


FIGURE 11. The flip operation on Schnyder woods of triangulations

Proof. First we show that essential circuits of a Schnyder wood are exactly circuits of length 3. Consider an essential clockwise circuit Δ of a Schnyder wood of a triangulation T. We claim that all edges inside Δ have their origin inside Δ. (Proof: assume a contrario that an edge e inside Δ has its origin on Δ, and let i be the label of e. Then the path with label i starting at e and going to a_i; this path thus reaches Δ, contradicting the fact that $\mathcal{C}$ has no chordal path.) Hence, if n denotes the number of vertices inside Δ, the number of edges inside Δ is $3n$. Moreover, writing k for the length of Δ, Euler's relation implies that the number of edges inside Δ is $3n+k-3$, so that $k=3$. Conversely, similar arguments ensure that an oriented circuit of length 3 is essential. Hence, essential circuits of Schnyder woods are exactly circuits of length 3, and no edge goes from a vertex of Δ toward the interior of Δ. It easily follows from Condition C1 that the effect of the circuit reversion on the labels is $i \to i+1$ on the circuit and $i \to i-1$ inside the circuit. □

Definition. Given a triangulation T, the Schnyder wood associated to the minimal 3-orientation is called the *minimal* Schnyder wood of T. This corresponds to the unique 3-orientation of T with no clockwise circuit.

5.2. Schnyder woods on 3-connected maps. We describe here an extended definition of Schnyder woods to 3-connected maps, worked out independently by Di Battista *et al* [**3**] and Felsner [**46**]. It is convenient for us to formulate the definition in terms of half-edges (originally, the extended definition has been given in terms of edges, allowing bi-oriented bicolored edges). We will see that a nice property appearing in the extended definition is a duality relation. A map M with no loop nor multiple edge is *suspended* by distinguishing 3 vertices a_1, a_2, a_3 incident to the outer face in clockwise order, and adding to $\{a_1, a_2, a_3\}$ a half-edge reaching to the outer face. Given a suspended map M, a Schnyder wood of M is an orientation and labeling —with labels $\{1,2,3\}$— of the *half-edges* of M such that the following conditions are satisfied, (see Figure 13(a) for an example):

Schnyder woods (definition with half-edges):

- **C1:** Every edge e satisfies two possible configurations. 1) The two half-edges of e are both outgoing and have different labels; then e is called *bi-oriented*. 2) One half-edge is outgoing and the other is ingoing, both half-edges having the same label; then e is called *simply oriented*.
- **C2:** Each vertex v of M has outdegree 1 in each color, and the half-edges h_1, h_2, h_3 leaving v in color $\{1,2,3\}$ occur in clockwise order. Each half-edge entering v in color i is in the clockwise sector between h_{i+1} and h_{i-1}.
- **C3:** There is no interior face whose boundary is a directed circuit in one color.

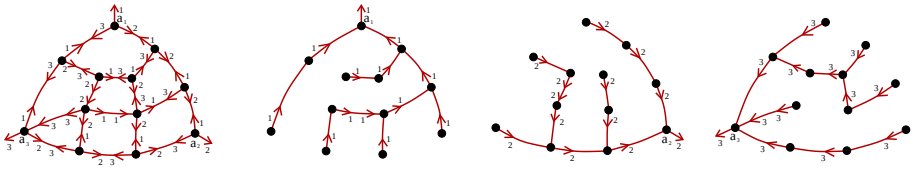


FIGURE 12. A suspended map endowed with a Schnyder wood, and the three spanning trees, one in each color.

Notice that the definition of Schnyder woods for triangulations —with edges— is a particular case of the definition with half-edges, namely when all edges are simply oriented and the labels of the outer edges are forgotten (then condition C3 is not useful, it follows from the acyclicity property).

Acyclicity property. The same acyclicity property as for triangulations holds for 3-connected maps. Given a suspended map M endowed with a Schnyder wood, let T_1, T_2, T_3 denote the maps formed by the edges of M having at least one of their half-edges with label 1, 2, 3, respectively. Each edge e of T_i is oriented like the half-edge of e with label i. Then Euler's relation and the Schnyder conditions (this is where C3 is useful) imply the following acyclicity property [**48**] similar to the one for triangulations:
Acyclicity property: "for $i = \{1, 2, 3\}$, the graph obtained from M by reversing the orientation of the half-edges with label i is acyclic".

As the acyclicity property extends from triangulations to 3-connected maps, the tree property and path property also extend to 3-connected maps, i.e.,

- for $i = \{1, 2, 3\}$, the graph T_i is a spanning tree of M rooted at a_i and endowed with the orientation toward a_i, see Figure 12.
- For each inner vertex v of T and $i \in \{1, 2, 3\}$, let $P_i(v)$ be the path in T_i from v to the root a_i. Then the paths $P_1(v)$, $P_2(v)$, and $P_3(v)$ do not intersect, except at their origin v.

Definition. Given a suspended map M, the *augmented map* of M is the map $\widehat{M}$ obtained by adding a vertex in the outer face of M connected to the three suspension vertices of M. The map M is called *internally 3-connected* if $\widehat{M}$ is 3-connected.

PROPOSITION 1.8 (Characterisation [**88, 46**]). *A suspended map admits a Schnyder wood (definition with half-edges) iff it is internally 3-connected.*

Proof. Assume that a suspended map M admits a Schnyder wood. Then, for any pair of vertices v and v' of $\widehat{M}$, it is easily shown that there exist three internally vertex-disjoint paths connecting v and v'. (The three paths are constructed as suitable concatenations of parts of the paths $P_i(v)$ and $P_j(v')$.) Menger's theorem [**14**] ensures that $\widehat{M}$ is 3-connected, i.e., M is internally 3-connected. Conversely, given an internally 3-connected map M, the existence of a Schnyder wood of M can be proved either inductively [**46**] or algorithmically [**3**]; in Section 4 we present an iterative algorithm computing a Schnyder wood of an internally 3-connected map. □

Duality relation and derived map. An advantage of the extended definition to 3-connected maps is to satisfy a simple duality relation. For suspended maps, we

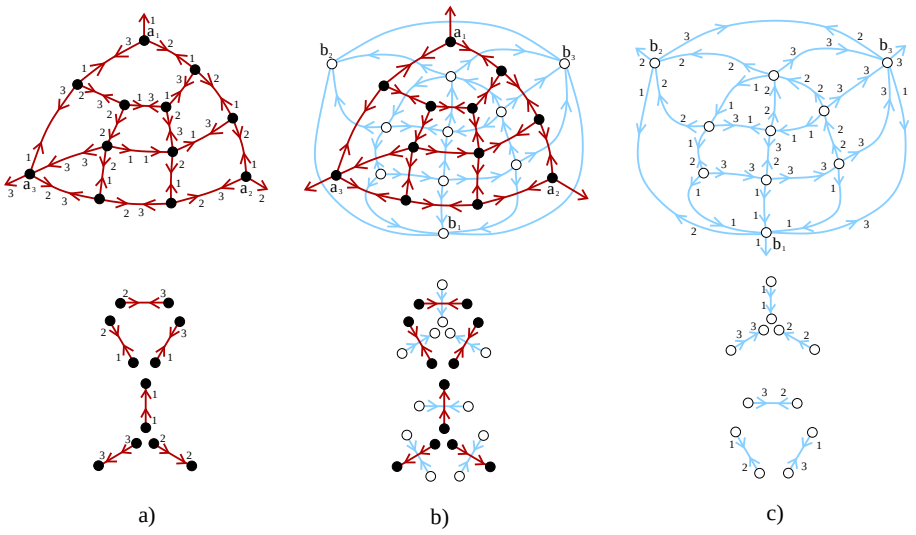


FIGURE 13. The duality rules (down) and an example (top): each configuration of an edge of M (Fig.a) corresponds to a configuration of the dual edge in M^* (Fig.c). The corresponding orientation of the derived map is an α_3-orientation (Fig.b).

adapt the definitions of dual maps and derived maps so as to take the suspension vertices into account. Given an internally 3-connected suspended map M, the *dual of* M is defined as the dual map M^* (as defined in Section 1.3) of the augmented map $\widehat{M}$. The *derived map* M' of M is defined as the superimposition of M and M^*, see Figure 13(b). Recall that each edge of M' corresponds either to a half-edge of M or to a half-edge of M^*. The vertices of M' are partitioned into the vertices of M —called *primal vertices*—, the vertices of M^* —called *dual vertices*—, and the vertices at a crossing between an edge of M and its dual edge —called *edge-vertices*—. As $\widehat{M}$ is 3-connected, Theorem 1.9 ensures that M^* is a 3-connected map with triangular outer face, called an *outer-triangular 3-connected map*. In particular, the map M^*, when suspended at its 3 outer vertices, admits a Schnyder wood according to Proposition 1.8. There is in fact a bijection between the Schnyder woods of M and the Schnyder woods of M^*, the proof relying on the above mentioned acyclicity properties of Schnyder woods.

Definition. Let M' be the derived map of a suspended internally 3-connected map. An α_3-orientation of M' is an orientation of M' such that all primal and dual vertices have outdegree 3 and all edge-vertices have outdegree 1.

PROPOSITION 1.9 (Duality for Schnyder woods [**48**]). *Let M be an internally 3-connected suspended map, and let M^* be the dual of M. Then the following sets are in bijection:*

- *the set of Schnyder woods of M,*
- *the set of Schnyder woods of M^*,*
- *the set of α_3-orientations of M'.*

The bijective rules are illustrated in Figure 13.

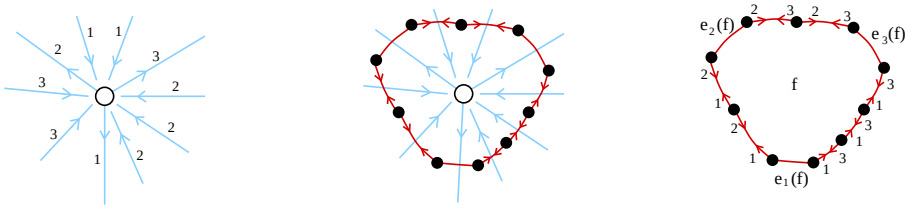


FIGURE 14. The Schnyder configuration of a (white) vertex in the dual map yields a generic configuration for the contour of a face.

The lattice structure of Schnyder woods then follows from the bijection with α_3-orientations and from the fact that the set of α_3-orientations of the derived map is a distributive lattice (Theorem 1.3). Felsner has characterised the essential circuits for an α_3-orientation, as stated in the following theorem.

PROPOSITION 1.10 (Lattice structure [**48**]). *The set of Schnyder woods of a fixed internally 3-connected map M is a distributive lattice, via the bijection with α_3-orientations of the derived map M'. The flip operation consists in reversing an essential clockwise circuit of the derived map, which is either the contour of an inner (quadrangular) face of M', or satisfies one of the configurations illustrated in Figure 18 page 80.*

Definition. For each internally 3-connected suspended map M, the Schnyder wood associated with the minimal α_3-orientation of the derived map M' is called the *minimal Schnyder wood* of M.

Remark. Given an internally 3-connected suspended map M endowed with a Schnyder wood, the duality relation ensures that the contour of an inner face f contains three particular edges $e_1(f)$, $e_2(f)$ and $e_3(f)$ such that (see Figure 14):

(1) for $i \in \{1, 2, 3\}$, $e_i(f)$ is either simply oriented in clockwise direction with label $i-1$ or in counterclockwise direction with label $i+1$, or bi-oriented with label $i-1$ in cw direction and label $i+1$ in ccw direction,
(2) the other edges of the contour are partitioned into three (possibly empty) intervals, the edges of the interval between $e_i(f)$ and $e_{i+1}(f)$ being bi-oriented with label i in cw direction and label $i+1$ in ccw direction.

6. Transversal structures

6.1. Introduction. In this section we define and investigate the so-called *transversal structures*, which play a similar role for irreducible triangulations as Schnyder woods for triangulations. Like Schnyder woods, the definition is local, but there is also a global characterisation, as a partition of the edges into two bipolar orientations (Proposition 1.11).

Transversal structures were originally introduced by Xin He [**73**], under the name of regular edge-labeling, for the family of irreducible triangulations. This gives rise to a nice algorithm of rectangular-dual drawing, which was recently applied to the theory of cartograms [**104, 35**]. However, the authors of [**78**] did not provide a combinatorial study. Our contribution in this section is to extend the definition to any planar map, give conditions of existence, and provide a complete description of the lattice property of transversal structures. Similarly as for bipolar orientations in

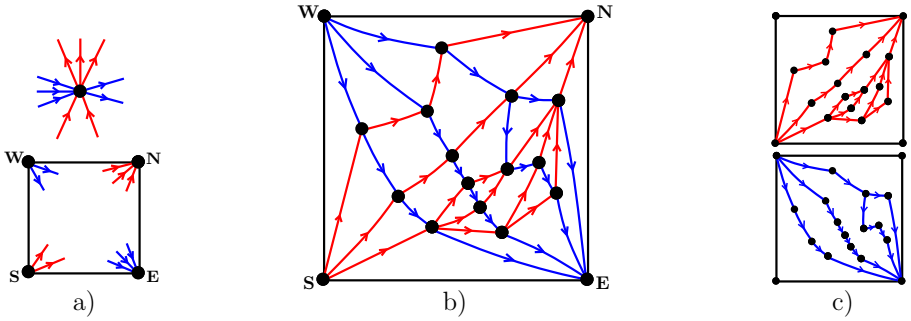


FIGURE 15. The local conditions C1 and C2 for transversal structures (Fig.a), a planar map M endowed with a transversal structure (Fig.b), the bipolar orientations on red and blue edges (Fig.c).

Section 4, the description is derived from a bijection between transversal structures of an irreducible triangulation T and α-orientations of the angular map of T. A first application of our result is to optimize cartogram algorithms [**104, 35**]. Indeed, our combinatorial study makes it possible to choose the best cartogram representation via exhaustive generation of the transversal structures, as discussed in Section 6.6.

The plan is the following. We define transversal structures, show a characterisation as a transversal pair of bipolar orientations, and prove that a triangulation admits a transversal structure iff it is irreducible (i.e., has no filled 3-cycle). Then we focus on the family $\mathcal{T}$ of irreducible triangulations. Using an alternative definition without orientation —called transversal edge-partition— we show that the set of transversal edge-partitions of a fixed irreducible triangulation is a distributive lattice (Theorem 14 page 39). In addition, we give a simple geometric formulation of the flip operation. The application to cartogram representation is then discussed.

6.2. Definition of transversal structures. Let M be a planar map with quadrangular outer face. Call N, E, S, and W the four outer vertices of M in clockwise order around the outer face. A *transversal structure* is an orientation and a partition of the inner edges of T into red and blue edges such that the following two conditions are satisfied (see Figure 15(b) for an example):

C1 (Inner vertices): In clockwise order around each inner vertex of T, its incident edges form: a non empty interval of outgoing red edges, a non empty interval of outgoing blue edges, a non empty interval of ingoing red edges, and a non empty interval of ingoing blue edges, see Figure 15(a).

C2 (Border vertices): All inner edges incident to N, E, S and W are ingoing red, ingoing blue, outgoing red, and outgoing blue, respectively.

Remark. This structure is also defined in [**78**] under the name of regular edge labeling, and I call it *transversal pair of bipolar orientations* in the article to be published [**58**]. In the thesis, I prefer to use the terminology of *transversal structures* and then give the characterisation as a transversal pair of bipolar orientations.

6.3. Characterisation. Recall that Schnyder woods have a nice formulation as a triple of trees with specific incidence relations. A parallel formulation exists for transversal structures, this time in terms of a transversal pair of bipolar orientations.

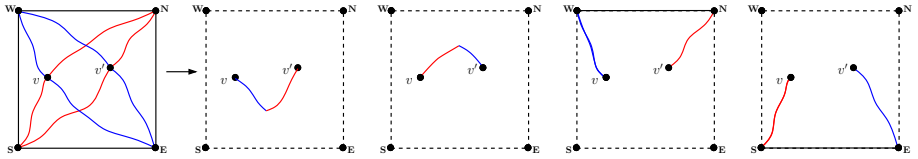


FIGURE 16. Given a planar map endowed with a transversal structure, there are 4 internally vertex-disjoint paths connecting any pair of inner vertices of M.

PROPOSITION 1.11. *Let M be planar map with quadrangular outer face and endowed with a transversal structure. Then the orientation induced by the red edges is a bipolar orientation on $M\backslash\{W, E\}$ with source S and sink N, and the orientation induced by the blue edges is a bipolar orientation on $M\backslash\{N, S\}$ with source W and sink E. Moreover, the two bipolar orientations are* transversal*: for any pair (P_r, P_b), where P_r is a red oriented path from S to N and P_b is a blue oriented path from W to E, the paths P_r and P_b meet at exactly one vertex, where they cross.*

Proof. If M has a monochrome circuit, consider a minimal monochrome circuit $\mathcal{C}$, i.e., no other monochrome circuit has its interior strictly contained in the interior of $\mathcal{C}$. Assume that $\mathcal{C}$ is clockwise red (the three other cases are treated similarly). Condition C2 ensures that all vertices of $\mathcal{C}$ are inner vertices of M. Hence, any vertex v on $\mathcal{C}$ satisfies Condition C1, ensuring that there is a blue path P starting from v toward the interior of $\mathcal{C}$. This path can not run into a circuit, by minimality of $\mathcal{C}$. Hence P reaches $\mathcal{C}$ at another vertex v'. As $\mathcal{C}$ is clockwise red, this contradicts Condition C1 at the vertex v'. Hence there is no monochrome circuit, i.e., the orientation of red edges and the orientation of the blue edges are acyclic. According to Conditions C1 and C2, the unique minimum of the red acyclic orientation is S and the unique maximum is N. Similarly, the unique minimum of the blue orientation is W and the unique maximum is E. The proof that the two bipolar orientations are transversal (as stated) follows similar arguments. A red path P_r from S to N and a blue path P_b from W to E clearly meet at least once at a vertex v, where they cross; and they can not meet again after crossing at v without breaking Condition C1. □

Unfortunately, we have not been able to characterise the family of planar maps that admit a transversal structure. We only provide here necessary conditions of existence in terms of the degree of faces and the degree of connectivity. We provide a precise condition of existence only for triangulations of the 4-gon, the condition of existence being the absence of separating 4-cycle (irreducibility).

FACT 1.1 (Configuration of faces). *Let M be a planar map with quadrangular outer face and endowed with a transversal structure. Then all inner faces of M have degree 3 or 4 and obey one of the configurations illustrated in Figure 18.*

Proof. Define the blue-map M_b of M as the map obtained from M by deleting red edges, coloring blue the 4 outer edges, and directing the outer edges from W to E. Consider an inner face f_b of the blue map. As the orientation of the blue edges is bipolar (Proposition 1.11), there exists a source-vertex s and a sink-vertex t incident to f_b such that the contour of f_b is partitioned into two paths P_l (P_r)

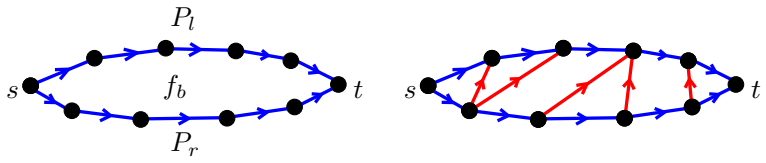


FIGURE 17. A face of the blue-map split by the transversal red edges.

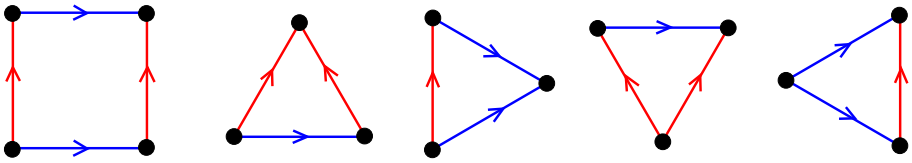


FIGURE 18. Possible configurations for an inner face of a map endowed with a transversal structure.

that go from s to t with f on the right (on the left, respectively). Given Condition C1 of transversal structures, each red edge inside f_b goes from a vertex of $P_r\backslash\{s,t\}$ to a vertex of $P_l\backslash\{s,t\}$. In addition, Condition C1 ensures that each vertex of $P_l\backslash\{s,t\}$ has at least one incident ingoing red edge inside f_b and each vertex of $P_r\backslash\{s,t\}$ has at least one incident outgoing red edge inside f_b. Hence, as illustrated in Figure 17, the red edges split f_b into faces that obey one of the configurations shown in Figure 18. □

Definition. Let M be a planar map with no loop nor multiple edge and with quadrangular outer face. The augmented map $\widehat{M}$ of M is the map obtained by adding a vertex in the outer face of M connected to the four outer vertices of M. The map M is called *internally 4-connected* if $\widehat{M}$ is 4-connected.

FACT 1.2 (Connectivity condition). *Let M be a planar map with quadrangular outer face and endowed with a transversal structure. Then M is internally 4-connected.*

Proof. Given two vertices v and v' of M, Proposition 1.11 ensures that there exist red paths $P_r(v)$ ($P_r(v')$) going from S to N and passing by v (by v', respectively). Using suitable concatenations of parts of these paths, it is easily shown that there exist 4 internally vertex-disjoint paths connecting v and v', see Figure 16 for an example. Hence, Menger's theorem ensures that $\widehat{M}$ is 4-connected, i.e., M is internally 4-connected. □

PROPOSITION 1.12 (Characterisation). *A triangulation T of the 4-gon admits a transversal structure iff T is irreducible.*

Proof. Let T be a triangulation of the 4-gon and let $\widehat{T}$ be the augmented map of T. It is easily checked that the presence of a filled 3-cycle $\Delta = \{x, y, z\}$ implies that the triple $\{x, y, z\}$ is separating in T because the deletion of $\{x, y, z\}$ separates the interior of Δ from the exterior of Δ. Hence, Fact 1.2 ensures that a triangulation has to be irreducible to admit a transversal structure. Conversely, Kant and He [**78**] have proved constructively that each irreducible triangulation (of the 4-gon) admits a transversal structure. (We will provide in Section 5 an alternative algorithm.) □

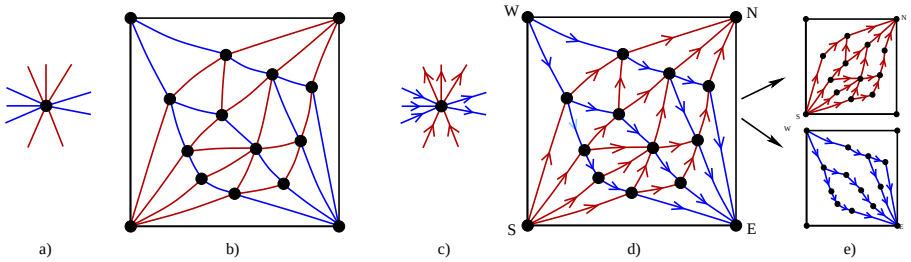


FIGURE 19. The local conditions of transversal edge-partitions (a) and a complete example (b). In parallel, the local conditions of transversal structures (c) and a complete example (d).

The following proposition is of minor interest to our study, but presents itself as parallel to the acyclicity property of Schnyder woods.

PROPOSITION 1.13 (Acyclicity property). *Given a map M endowed with a transversal structure, the orientation of M (forgetting the colors) is acyclic. The orientations obtained by reversing all red or all blue edges are also acyclic.*

Proof. Consider an inclusion-minimal oriented circuit $\mathcal{C}$. Clearly, $\mathcal{C}$ is not the contour of a face, according to Figure 18. Hence, there is an edge e inside $\mathcal{C}$. This edge is extended to a path using the fact that each vertex of M has indegree and outdegree at least 1. The path is simple inside $\mathcal{C}$, otherwise it would run into a circuit strictly inside $\mathcal{C}$. Hence the extended path, considered inside $\mathcal{C}$, is a chordal oriented path of $\mathcal{C}$, yielding an oriented circuit whose interior is strictly contained in the interior of $\mathcal{C}$, a contradiction. □

6.4. Transversal edge-partitions. We now focus on the family of irreducible triangulations. According to Proposition 1.12, these are exactly the triangulations of the 4-gon admitting a transversal structure. We give here a simplified definition with no orientation, which proves more convenient to describe the lattice property (Theorem 1.4 page 50).

Let T be an irreducible triangulation, and call W, N, E, and S its outer vertices traversed in clockwise order. A *transversal edge-partition* of T is a partition of the inner edges of T into blue and red edges, such that the following conditions are satisfied, see Figure 19.

- **C1' (Inner vertices):** In clockwise order around each inner vertex, its incident edges form: a non empty interval of red edges, a non empty interval of blue edges, a non empty interval of red edges, and a non empty interval of blue edges, see Figure 19(a).
- **C2' (Border vertices):** All inner edges incident to W and to E are blue and all inner edges incident to S and to N are red.

Observe that the conditions for the colors of edges in a transversal structure and in a transversal edge-partition are the same. Thus, transversal structures seem to contain more informations, as the edges have in addition to be oriented in a specific way. The following proposition ensures that the additional information given by the orientation of edges is in fact redundant, which means that there is a unique way to orient the edges of a transversal edge partition so as to yield a transversal structure.

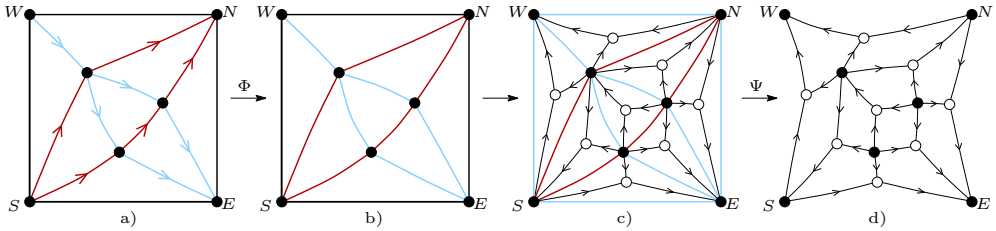


FIGURE 20. Given an irreducible triangulation T endowed with a transversal structure Z (Fig.a) and the induced transversal edge-partition $E = \Phi(Z)$ (Fig.b), construction of the angular map $Q(T)$ and of the α_4-orientation of $Q(T)$ image of E by the mapping Ψ (Fig.d).

PROPOSITION 1.14. *Let T be an irreducible triangulation. To each transversal structure of T corresponds a transversal edge-partition of T, obtained by removing the direction of the edges. This correspondence is a bijection.*

Proof. This bijection will be established in Proposition 1.15 as an intermediate mapping of another bijection. □

Proposition 1.14 allows us to manipulate equivalently transversal structures and transversal edge-partitions. While the first point of view is useful to describe the straight-line drawing algorithm given in Chapter 5, the definition without orientations is more convenient to describe the lattice structure and a bijection with ternary trees given in Chapter 3.

6.5. Lattice property of transversal edge-partitions. We investigate on the set $\mathcal{E}(T)$ of transversal edge-partitions of a fixed irreducible triangulation T. Kant and He [**78**] have shown that $\mathcal{E}(T)$ is not empty and that an element of $\mathcal{E}(T)$ can be computed in linear time. In this section, we prove that $\mathcal{E}(T)$ is a distributive lattice. This property is to be compared with the lattice property of euleration orientations, bipolar orientations, and Schnyder woods. Our proof follows similar lines as the proof of the lattice structure of bipolar orientations, as we prove that transversal structures of an irreducible triangulation are in bijection with specific α-orientations of the angular map.

6.5.1. *Bijection with orientations of the angular map.* Let T be an irreducible triangulation and let $Q(T)$ be the angular map of T, where for convenience we discard the white vertex corresponding to the outer face of T as well as its incident edges, see Figure 20.

We consider the function $\alpha_4 : V_Q \to \mathbf{N}$ specified as follows,

- each black vertex v of $Q(T)$ corresponding to an inner vertex of T satisfies $\alpha_4(v) = 4$,
- each white vertex v of $Q(T)$ satisfies $\alpha_4(v) = 1$,
- the outer black vertices satisfy $\{\alpha_4(N) = \alpha_4(S) = 2,\ \alpha_4(W) = \alpha_4(E) = 0\}$.

PROPOSITION 1.15. *Given T an irreducible triangulation and $Q(T)$ the angular map of T, the following sets are in bijection:*

- *transversal pairs of bipolar orientations of T,*
- *transversal edge-partitions of T,*

- *α_4-orientations of $Q(T)$.*

In particular, Proposition 1.15 ensures that the definitions of transversal edge-partition and of transversal structures are equivalent, i.e., the orientation of edges is a redundant information once the colors are fixed.

Given T an irreducible triangulation and $Q(T)$ the angular map of T, we denote by $\mathcal{B}$ the set of transversal pairs of bipolar orientations of T, $\mathcal{E}$ the set of transversal edge-partitions of T, and $\mathcal{O}$ the set of α_4-orientations of $Q(T)$. To prove Proposition 1.15, we introduce two mappings Φ and Ψ respectively from $\mathcal{B}$ to $\mathcal{E}$ and from $\mathcal{E}$ to $\mathcal{O}$.

Given $Z \in \mathcal{B}$, $\Phi(Z)$ is simply the edge-bicoloration induced by Z, as illustrated in Figure 20(a)-(b). It is straightforward that $\Phi(Z) \in \mathcal{E}$. In addition, Φ is clearly injective. Indeed, starting from a transversal edge-partition, the directions of edges of $Q(T)$ are assigned greedily so as to satisfy the local rules of a transversal structure. The fact that the propagation of edge directions is done without conflict is not straightforward; it will follow from the surjectivity of the mapping Φ, to be proved later.

Given $X \in \mathcal{E}$, we define $\Psi(X)$ as the following orientation of $Q(T)$. First, color blue the four outer edges of T. Then, for each angle (v, f) of T, orient the corresponding edge of $Q(T)$ out of v if (v, f) is bicolored, and toward v if (v, f) is unicolored. (An angle (v, f) of T is called *bicolored* if it is delimited by two edges of T having different colors, and is called unicolored otherwise.) Condition C1 implies that all inner black vertices of $Q(T)$ have outdegree 4. In addition, Condition C2 and the fact that the four outer edges of T have been colored blue imply that E and W have outdegree 0 and that N and S have outdegree 2.

The following lemma ensures that all white vertices have outdegree 1 in $\Psi(X)$, so that $\Psi(X)$ is an α_4-orientation.

LEMMA 1.1. *Let T be a planar map with quadrangular outer face, triangular inner faces, and endowed with a transversal edge-partition, the four outer edges being additionally colored blue. Then there is no mono-colored inner face, i.e., each inner face of T has two sides of one color and one side of the other color.*

Proof. Let Λ be the number of bicolored angles of T and let n be the number of inner vertices of T. Condition C1 implies that there are $4n$ bicolored angle incident to an inner vertex of T. Condition C2 and the fact that all outer edges are colored blue imply that two angles incident to N and two angles incident to S are bicolored. Hence, $\Lambda = 4n + 4$.

Moreover, as T has quadrangular outer face and triangular inner faces, Euler's relation ensures that T has $2n + 2$ inner faces. For each inner face, two cases can arise: either the three sides have the same color, or two sides are of one color and one side is of the other color. In the first (second) case, the face has 0 (2, respectively) bicolored angles. As there are $2n + 2$ inner faces and $\Lambda = 4n + 4$, the pigeonhole principle implies that all inner faces have a contribution of 2 to the number of bicolored angles, which concludes the proof. □

Lemma 1.1 ensures that Ψ is a mapping from $\mathcal{E}$ to $\mathcal{O}$. In addition, it is clear that Ψ is injective. To prove Proposition 1.15, it remains to prove that Φ and Ψ are surjective. As Φ is injective, it is sufficient to show that $\Psi \circ \Phi$ is surjective. Thus, given $O \in \mathcal{O}$, we have to find $Z \in \mathcal{B}$ such that $\Psi \circ \Phi(Z) = O$.

Computing the preimage of an α_4-orientation. We now describe a method to compute a transversal structure Z consistent with a given α_4-orientation O, i.e., such that $\Psi \circ \Phi(Z) = O$. The algorithm makes use of a sweeping process to orient and color the inner edges of T; a simple (i.e., not self-intersecting) path $\mathcal{P}$ of inner edges of T going from W to E is maintained, the path moving progressively toward the vertex S (at the end, the path is $\mathcal{P} = W \to S \to E$). We require that the following invariants are satisfied throughout the sweeping process.

(1) Each vertex of $\mathcal{P}\backslash\{W, E\}$ has two outgoing edges on each side of $\mathcal{P}$ for the α_4-orientation O.
(2) The inner edges of T already oriented and colored are those on the left of $\mathcal{P}$.
(3) Condition C1 holds around each inner vertex of T on the left of $\mathcal{P}$.
(4) A partial version of C1 holds around each vertex v of $\mathcal{P}\backslash\{W, E\}$. The edges incident to v on the left of $\mathcal{P}$ form in clockwise order: a possibly empty interval of ingoing blue edges, a non-empty interval of outgoing red edges, and a possibly empty interval of outgoing blue edges.
(5) All edges already oriented and colored and incident to N, E, S, W are ingoing red, ingoing blue, outgoing red, and outgoing blue, respectively.
(6) The edges of T already colored (and oriented) are consistent with the α_4-orientation O, i.e., for each angle (v, f) delimited by two edges of T already colored, the corresponding edge of $Q(T)$ is going out of v iff the angle is bicolored.

At first we need a technical result ensuring that Invariant (1) is sufficient for a path to be simple.

LEMMA 1.2. *Let T be an irreducible triangulation and $Q(T)$ be the angular map of T, endowed with an α_4-orientation O. Let $\mathcal{P}$ be a path of inner edges of T from W to E such that each vertex of $\mathcal{P}\backslash\{W, E\}$ has outdegree 2 on each side of $\mathcal{P}$ for the α_4-orientation. Then the path $\mathcal{P}$ is simple.*

Proof. Assume that the path $\mathcal{P}$ loops into a circuit; and consider an inclusion-minimal such circuit $\mathcal{C} = (v_0, v_1, \ldots, v_k = v_0)$. We define $n_\bullet$, $n_\circ$ and e as the numbers of black vertices (i.e., vertices of T), white vertices and edges of $Q(T)$ inside $\mathcal{C}$. As T is triangulated and $\mathcal{C}$ has length k, Euler's relation ensures that there are $2n_\bullet + k - 2$ faces inside $\mathcal{C}$, i.e., $n_\circ = 2n_\bullet + k - 2$. Counting edges of $Q(T)$ inside $\mathcal{C}$ according to their incident white vertex gives (i) : $e = 3n_\circ = 6n_\bullet + 3k - 6$. Edges of $Q(T)$ inside $\mathcal{C}$ can also be counted according to their origin for the α_4-orientation. Each vertex of $\mathcal{C}$ —except possibly the self-intersection vertex v_0— has outdegree 2 in the interior of $\mathcal{C}$ for the α_4-orientation. Hence, (ii) : $e = 4n_\bullet + n_\circ + 2k - 2 + \delta = 6n_\bullet + 3k - 4 + \delta$, where $\delta \geq 0$ is the outdegree of v_0 inside $\mathcal{C}$. Taking $(ii) - (i)$ yields $\delta = -2$, a contradiction. □

The path $\mathcal{P}$ is initialized with all neighbours of N, from W to E. In addition, all inner edges incident to N are initially colored red and directed toward N, see Figure 23(b). The invariants (1)-to-(6) are clearly true at the initial step.

Let us introduce some terminology in order to describe the sweeping process. Thoughout the process, the vertices of $\mathcal{P}$ are ordered from left to right, with W as leftmost and E as rightmost vertex. Given v, v' a pair of vertices on $\mathcal{P}$ —with v on the left of v'— the part of $\mathcal{P}$ going from v to v' is denoted by $[v, v']$. For each vertex w on $\mathcal{P}$, let $f_1, \ldots, f_k$ be the sequence of faces of T incident to w on the right of $\mathcal{P}$, taken in counterclockwise order. The edge of $Q(T)$ associated to the angle (w, f_1)

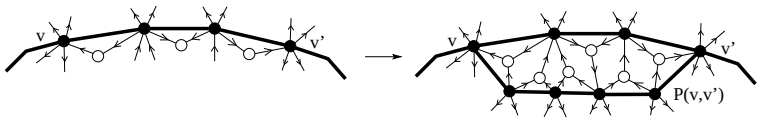


FIGURE 21. An admissible pair v, v' of vertices, and the matching path $P(v, v')$.

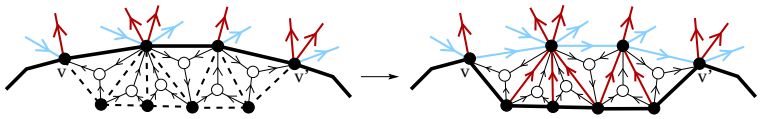


FIGURE 22. The update step of the iterative algorithm finding the preimage of an α_4-orientation.

(angle (w, f_k)) is denoted by $\epsilon_{\text{left}}(w)$ (by $\epsilon_{\text{right}}(w)$, respectively). A pair of vertices v, v' on $\mathcal{P}$ —with v on the left of v'— is called *admissible* if $\epsilon_{\text{right}}(v)$ is ingoing at v, $\epsilon_{\text{left}}(v')$ is ingoing at v', and for each vertex $w \in [v, v']\backslash\{v, v'\}$, the edges $\epsilon_{\text{left}}(w)$ and $\epsilon_{\text{right}}(w)$ are going out of w. Clearly an admissible pair always exists: take a pair v, v' of vertices on $\mathcal{P}$ that satisfy $\{\epsilon_{\text{right}}(v)$ ingoing at v, $\epsilon_{\text{left}}(v')$ ingoing at $v'\}$ and are closest possible. Notice that two vertices v, v' forming an admissible pair are not neighbours on $\mathcal{P}$ (otherwise the white vertex associated to the face on the right of $[v, v']$ would have outdegree > 1). Let $w_0 = v, w_1, \ldots, w_k, w_{k+1} = v'$ $(k \geq 1)$ be the sequence of vertices of $[v, v']$. The *matching path* of v, v' is the path of edges of T that starts at v, then visits the neighbours of $w_1, w_2, \ldots, w_k$ on the right of $\mathcal{P}$, and finishes at v'. The matching path of v, v' is denoted by $P(v, v')$. Let $\mathcal{P}'$ be the path obtained from $\mathcal{P}$ when substituting $[v, v']$ by $P(v, v')$. As shown in Figure 21, the path $\mathcal{P}'$ goes from W to E and each vertex of $\mathcal{P}'\backslash\{W, E\}$ has outdegree 2 on each side of $\mathcal{P}'$ for the α_4-orientation O. Hence the path $\mathcal{P}'$ is simple according to Lemma 1.2. Moreover, by definition of $P(v, v')$, all edges of T in the region enclosed by $[v, v']$ and $P(v, v')$ connect a vertex of $[v, v']\backslash\{v, v'\}$ to a vertex of $P(v, v')\backslash\{v, v'\}$, see Figure 22.

We can now describe the operations performed at each step of the sweeping process, as shown in Figure 22.

- Choose an admissible pair v, v' of vertices on $\mathcal{P}$.
- Color blue and orient from left to right all edges of $[v, v']$.
- Color red all edges inside the area enclosed by $[v, v']$ and $P(v, v')$, and orient these edges from $P(v, v')$ to $[v, v']$.
- Update the path $\mathcal{P}$, the part $[v, v']$ being replaced by $P(v, v')$.

According to the discussion above, the path $\mathcal{P}$ is still simple after these operations, and satisfies Invariant (1). All other invariants (2)-to-(6) are easily shown to remain satisfied, as illustrated in Figure 22. At the end, the path $\mathcal{P}$ is equal to $W \to S \to E$. Invariants (3) and (5) ensure that the directions and colors of the inner edges of T form a transversal structure Z; and Invariant (6) ensures that Z is consistent with the α_4-orientation O, i.e., $\Psi \circ \Phi(Z) = O$, see also Figure 23 for a complete execution of the algorithm. This concludes the proof of Proposition 1.15.

6.5.2. *Essential circuits of an α_4-orientation.* Proposition 1.15 ensures that the set $\mathcal{E}$ of transversal edge-partitions of an irreducible triangulation is a distributive

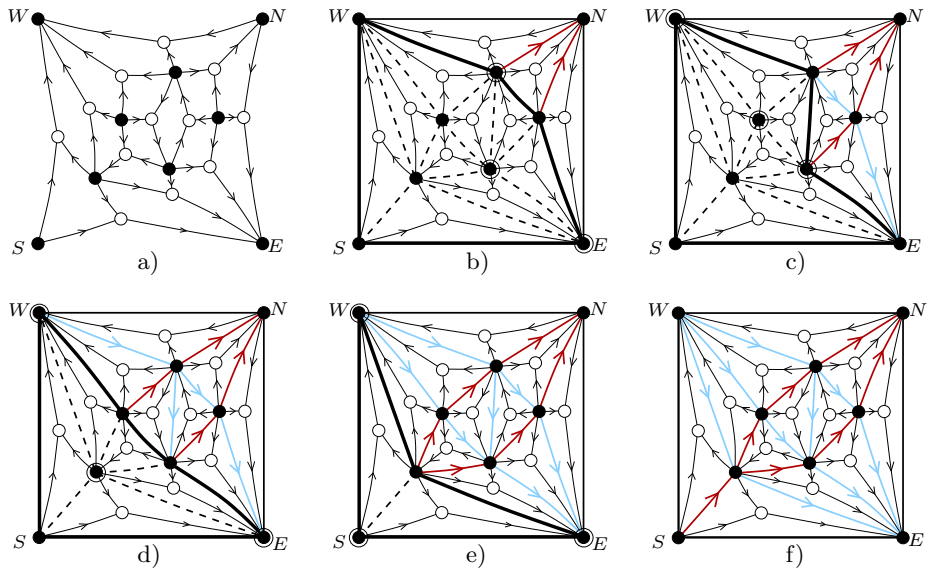


FIGURE 23. The complete execution of the algorithm calculating the preimage of an α_4-orientation O. At each step, the vertices of the matching path $P(v, v')$ are surrounded.

lattice, as $\mathcal{E}$ is in bijection with the distributive lattice formed by the α_4-orientations of the angular map. By definition, the flip operation on $\mathcal{E}$ is the effect of a flip operation on $\mathcal{O}$ via the bijection. Recall that a flip operation on an α-orientation consists in reversing a clockwise essential circuit (circuit with no chordal path). Hence, to describe the flip operation on $\mathcal{E}$, we have to characterise the essential circuits of an α_4-orientation. For this purpose, we introduce the concept of *straight path*.

Consider an irreducible triangulation T endowed with a transversal structure. Color blue the four outer edges of T and orient them from W to E. The conditions C1 and C2 ensure that there are four possible types for a bicolored angle (e, e') of T, with e' following e in cw order: (outgoing red, outgoing blue) or (outgoing blue, ingoing red) or (ingoing red, ingoing blue) or (ingoing blue, outgoing red). The type of an edge of $Q(T)$ corresponding to a bicolored angle of T (i.e., an edge going out of a black vertex) is defined as the type of the bicolored angle. For such an edge e, the *straight path* of e is the oriented path $\mathcal{P}$ of edges of $Q(T)$ that starts at e and such that each edge of $\mathcal{P}$ going out of a black vertex has the same type as e. Such a path is unique, as there is a unique choice for the outgoing edge at a white vertex (each white vertex has outdegree 1).

LEMMA 1.3. *The straight path $\mathcal{P}$ of an edge $e \in Q(T)$ going out of a black vertex is simple and ends at an outer black vertex of $Q(T)$.*

Proof. Notice that the conditions of a transversal structure remain satisfied if the directions of edges of one color are reversed and then the colors of all inner edges are switched; hence the edge e can be assumed to have type (outgoing red, outgoing blue) without loss of generality. Let $v_0, v_1, v_2, \ldots, v_i, \ldots$ be the sequence of vertices of the straight path $\mathcal{P}$ of e, so that the even indices correspond to black vertices of $Q(T)$ and the odd indices correspond to white vertices of $Q(T)$. Observe

that, for $k \geq 0$, the black vertices v_{2k} and v_{2k+2} are adjacent in T and the edge (v_{2k}, v_{2k+2}) is either outgoing red or outgoing blue. Hence $(v_0, v_2, v_4, \ldots, v_{2k}, \ldots)$ is an oriented path of T, so that it is simple, according to Proposition 1.11. Hence, $\mathcal{P}$ does not pass twice by the same black vertex, i.e., $v_{2k} \neq v_{2k+2}$ for $k \neq k'$; and $\mathcal{P}$ neither passes twice by the same white vertex (otherwise $v_{2k+1} = v_{2k+3}$ for $k \neq k'$ would imply $v_{2k+2} = v_{2k+4}$ by unicity of the outgoing edge at each white vertex, a contradiction). Thus $\mathcal{P}$ is a simple path, so that it ends at a black vertex of $Q(T)$ having no outgoing edge of type (outgoing red, outgoing blue), i.e., $\mathcal{P}$ ends at an outer black vertex of $Q(T)$. □

PROPOSITION 1.16. *Given T an irreducible triangulation and $Q(T)$ its angular map endowed with an α_4-orientation X, an essential clockwise circuit $\mathcal{C}$ of X satisfies either of the two following configurations,*

- *The circuit $\mathcal{C}$ is the boundary of a (quadrangular) inner face of $Q(T)$, see Figure 24(a).*
- *The circuit $\mathcal{C}$ has length 8. The four black vertices of $\mathcal{C}$ have no outgoing edge inside $\mathcal{C}$. The four white vertices of $\mathcal{C}$ have their unique incident edge not on $\mathcal{C}$ inside $\mathcal{C}$, see Figure 24(b) for an example.*

Proof. First we claim that no edge of $Q(T)$ inside $\mathcal{C}$ has its origin on $\mathcal{C}$; indeed the straight path construction ensures that such an edge could be extended to a chordal path of $\mathcal{C}$, which is impossible. We define $n_\bullet$, $n_\circ$ and e as the number of black vertices, white vertices and edges inside $\mathcal{C}$. We denote by $2k$ the number of vertices on $\mathcal{C}$, so that there are k black and k white vertices on $\mathcal{C}$. Euler's relation and the fact that all inner faces of $Q(T)$ are quadrangular ensure that $(i):\ e = 2(n_\bullet + n_\circ) + k - 2$. As each white vertex of $Q(T)$ has degree 3, a white vertex on $\mathcal{C}$ has a unique incident edge not on $\mathcal{C}$. Let l be the number of white vertices such that this incident edge is inside $\mathcal{C}$ (notice that $l \leq k$). Counting edges inside $\mathcal{C}$ according to their incident white vertex gives $(ii):\ e = 3n_\circ + l$. Edges inside to $\mathcal{C}$ can also be counted according to their origin for the α_4-orientation. As no edge inside $\mathcal{C}$ has its origin on $\mathcal{C}$, we have $(iii):\ e = 4n_\bullet + n_\circ$. Taking $2(i) - (ii) - (iii)$ yields $l = 2k - 4$. As k is a positive integer and l is a nonnegative integer satisfying $l \leq k$, the only possible values for l and k are $\{k = 4, l = 4\}$, $\{k = 3, l = 2\}$, and $\{k = 2, l = 0\}$. It is easily seen that the case $\{k = 3, l = 2\}$ would correspond to a separating 3-cycle. Hence, the only possible cases are $\{k = 2, l = 0\}$ and $\{k = 4, l = 4\}$, shown in Figure 24(a) and 24(b), respectively. The first case corresponds to a circuit of length 4, which has to be the boundary of a face, as the angular map of an irreducible triangulation (more generally, of a 3-connected map) is well known to have no filled 4-cycle. □

6.5.3. *Flip operation on transversal structures.* As we prove now, essential circuits of α_4-orientations correspond to specific patterns on transversal structures, making it possible to have a simple geometric interpretation of the flip operation, formulated directly on the transversal structure.

Given T an irreducible triangulation endowed with a transversal edge-partition, we define an *alternating 4-cycle* as a cycle $\mathcal{C} = (e_1, e_2, e_3, e_4)$ of length 4 of edges of T that are color-alternating (i.e., two adjacent edges of $\mathcal{C}$ have different colors). Given a vertex v on $\mathcal{C}$, we call left-edge (right-edge) of v the edge of $\mathcal{C}$ starting from v and having the exterior of $\mathcal{C}$ on its left (on its right, respectively).

LEMMA 1.4. *An alternating 4-cycle $\mathcal{C}$ in a transversal structure satisfies either of the two following configurations.*

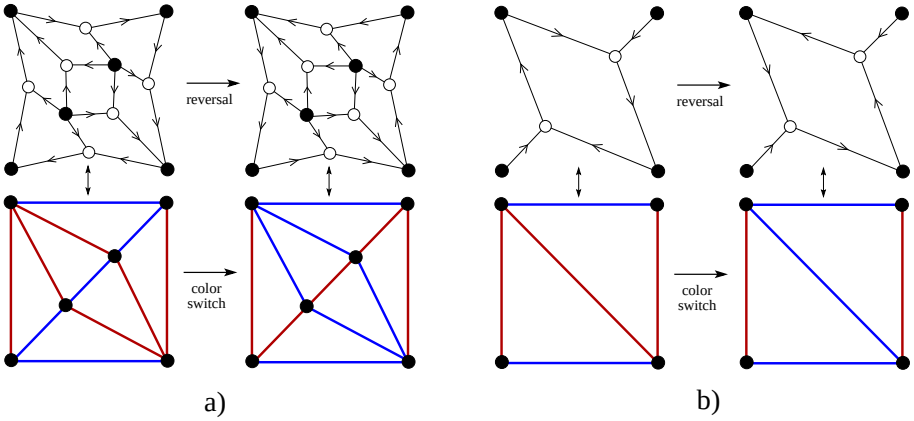


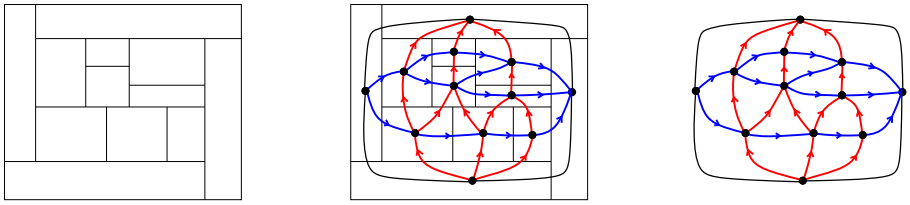
FIGURE 24. The two possible configurations of an essential clockwise circuit $\mathcal{C}$ of $Q(T)$. In each case, an alternating 4-cycle is associated to the circuit; and reversing the circuit orientation corresponds to switching the edge colors inside the alternating 4-cycle.

- *All edges inside $\mathcal{C}$ and incident to a vertex v of $\mathcal{C}$ have the color of the left-edge of v. Then $\mathcal{C}$ is called a* left alternating 4-cycle
- *All edges inside $\mathcal{C}$ and incident to a vertex v of $\mathcal{C}$ have the color of the right-edge of v. Then $\mathcal{C}$ is called a* right alternating 4-cycle.

Proof. Let k be the number of vertices inside $\mathcal{C}$. Condition C1' ensures that there are $4k$ bicolored angles incident to a vertex inside $\mathcal{C}$. Moreover, Euler's relation ensures that there are $2k+2$ faces inside $\mathcal{C}$. Hence Lemma 1.1 implies that there are $4k+4$ bicolored angles inside $\mathcal{C}$. As a consequence, there are four bicolored angles inside $\mathcal{C}$ that are incident to a vertex of $\mathcal{C}$. As $\mathcal{C}$ is alternating, each of the four vertices of $\mathcal{C}$ is incident to at least one bicolored angle. Hence the pigeonhole principle implies that each vertex of $\mathcal{C}$ is incident to one bicolored angle inside $\mathcal{C}$. Moreover, each of the four inner faces (f_1, f_2, f_3, f_4) inside $\mathcal{C}$ and incident to an edge of $\mathcal{C}$ has two bicolored angles, according to Lemma 1.1. As such a face f_i has two angles incident to vertices of $\mathcal{C}$, at least one bicolored angle of f_i is incident to a vertex of $\mathcal{C}$. As there are four bicolored angles inside $\mathcal{C}$ incident to vertices of $\mathcal{C}$, the pigeonhole principle ensures that each face f_1, f_2, f_3, f_4 has exactly one bicolored angle incident to a vertex of $\mathcal{C}$. As each vertex of $\mathcal{C}$ is incident to one bicolored angle inside $\mathcal{C}$, these angles are in the same direction. If they start from $\mathcal{C}$ in clockwise (counterclockwise) direction, then $\mathcal{C}$ is a right alternating 4-cycle (left alternating 4-cycle, respectively). □

THEOREM 1.4. *The set of transversal edge-partitions of a fixed irreducible triangulation is a distributive lattice. The flip operation consists in switching the edge-colors inside a right alternating 4-cycle, turning it into a left alternating 4-cycle.*

Proof. The two possible configurations for an essential clockwise circuit of an α_4-orientation are represented respectively in Figure 24(a) and Figure 24(b). In each case, the essential circuit of the angular map corresponds to a right alternating 4-cycle on the irreducible triangulation. Notice that a clockwise face of $Q(T)$ corresponds to the vertex-empty alternating 4-cycle, whereas essential circuits of length



a) b) c)

FIGURE 25. The dual map of a rectangular layout is naturally endowed with a transversal structure.

8 correspond to all possible alternating 4-cycles with at least one vertex in their interior (Figure 24(a) gives an example). As shown in Figure 24, the effect of reversing a clockwise essential circuit of the angular map is clearly a color switch of the edges inside the associated alternating 4-cycle. □

Definition. Given an irreducible triangulation T, the transversal structure of T with no right alternating 4-cycle is called *minimal*, as it is at the bottom of the distributive lattice.

6.6. Application to cartographic theory. The terminology of *maps* refers to the classical notion of geographic map. Indeed, a geographic map G (e.g., the map of Europe) consists of regions (the countries), two regions being adjacent if they share a border. The topological information of a geographic map is thus represented by a (dual) planar map M_G that has one vertex v for each region R of G. Cartographic theory aims at finding embeddings of a map meeting specific requirements. Typical restrictions can be imposed on the size and on the shape of the regions. Requirements on the size make it possible to have a visual representation of variables such as the population of each country. Requirements on the shape aim at keeping the complexity of the regions as small as possible, so as to make a visual inspection easier.

Rectangular layouts from transversal structures. We focus here on so-called *rectangular layouts*, i.e., all faces of the cartogram are rectangles and at most 3 rectangles meet at a point. This is a typical example of restriction on the shapes of the regions. The following proposition reveals the close connection between rectangular layouts and transversal structures, see Figure 25:

PROPOSITION 1.17 ([**78**]). *A map M admits a representation as a rectangular layout with 4 rectangles on the boundary iff the dual of M is an irreducible triangulation. Given a rectangular layout of M, color red and orient from bottom to top the edges of M^* that are dual to horizontal borders, and color blue and orient from left to right the edges of M^* that are dual to vertical borders. Then the induced bicoloration and orientation of inner edges of M^* is a transversal structure. Conversely, given a transversal structure, a rectangular layout consistent with the orientations and colors of the edges can be computed in linear time.*

The algorithm computing a rectangular layout from a transversal structure has been developed by Kant and He [**78**]. The idea is to consider the dual maps M_r and M_b of the red map and blue map of $T := M^*$, the maps M_r and M_b being endowed with the dual bipolar orientations. The abscissa (ordinate) of a vertex v

is then computed as the length of the longest path from the source to v in M_r (M_b, respectively).

In practice, a geographic map is mostly dual to a triangulated map, as the case where more than 3 borders meet at a same point can be considered as degenerated and is avoided using ϵ-perturbations. Moreover, the outer face can be made quadrangular by adding 4 regions N, S, W, and E, to the North, South, West, and East, respectively. Finally the occurence of separating 3-cycles is mostly due to vertices of degree 3 for instance Luxemburg is enclosed by Belgium, Germany, and France. As described in [**104**], this difficulty is bypassed by simple merge/split operations, for instance by considering Belgium and Luxemburg as a single region and splitting them afterwards.

Rectangular cartograms. As discussed above, up to minor modifications, it is possible to consider a geographic map M as the dual of an irreducible triangulation T, and each transversal structure of T yields a rectangular layout of M. In the literature, the terminology of rectangular cartograms refers to rectangular layouts where several quality criteria of visualization are specified [**35, 104**]. For instance, the area of each region S has to be close to a target-value A_S (e.g. A_S is the population of S), and the presence of too thin rectangles has to be avoided. The quality of a rectangular layout R is then measured by a function $f(R)$ combining the various aesthetic criteria to be met.

Optimizing the quality using exhaustive generation. The article by Speckmann and Van Kreveld describes several heuristics to optimize the layout. The principle is to start from a transversal structure of $T := M^*$, compute the associated rectangular layout of M using the Kant-He algorithm, and then modify locally the layout using techniques such as segment moving, so as to increase the value of $f(R)$. The algorithm is launched from several transversal structures of $T := M^*$. These are obtained by trying all orientations and colorations of edges and then keeping the ones satisfying the conditions of transversal structures. Even if the authors limit the exhaustive search by discarding some edge-orientations, the disadvantage of the method is to try a lot of edge-bicolorations that are not transversal structures. We propose another approach, based on our study of transversal structures. The key point is that the set $\mathcal{S}$ of transversal stuctures of T is a distributive lattice, and there exist efficient procedures to generate exhaustively all elements of a distributive lattice (see [**48**] for a discussion). The approach we propose is the following:

(1) Generate exhaustively all the transversal structures of T, and compute the corresponding rectangular layout in each case.
(2) Launch the optimization heuristics starting from all the corresponding rectangular layouts, and keep the rectangular layout maximizing the value of $f(R)$.

Conclusion. To sum up, we have presented in this chapter several combinatorial structures that characterise well known families of planar maps. Here is the list:

Structure	Family
Eulerian orientations	Eulerian maps
Bipolar orientations	2-connected maps
Schnyder woods (edges)	Triangulations
Schnyder woods (half-edges)	Internally 3-connected maps
Transversal structures	Irreducible triangulations

All these structures have a formulation as orientations with specified outdegree distribution on the vertices; hence they satisfy the properties of α-orientations. In particular, the set of structures of a fixed map is a distributive lattice. The structure at the bottom of the lattice is called *minimal*, for instance we have defined the minimal bipolar orientation of a rooted 2-connected map. Moreover, for each of the combinatorial structures, the flip-flop operations of the distributive lattice have a formulation as a simple pattern-switching operation on the map.

CHAPTER 2

Efficient computation

Introduction. In the first chapter, we have presented several combinatorial structures that characterise well known families of planar maps, namely Eulerian orientations, bipolar orientations, Schnyder woods, and transversal structures. This chapter focuses on their efficient computation. The first interest is to provide an algorithmic proof of existence of the combinatorial structures associated to each family of maps.

In addition to be correct, we require that the computation is *efficient*, i.e., proceeds in linear time. Indeed, as will be developed in Chapter 4 and 5, the combinatorial structures are directly used to carry out optimal encoding and compact straight-line drawing of planar maps. Hence, a fast computation of the structures allows us to have an algorithmics on planar maps both with a very good time complexity and space complexity (in terms of compacity of the output).

Results obtained in this chapter. For each of the combinatorial structures {Eulerian orientations, bipolar orientations, Schnyder woods, transversal structures}, a linear time algorithm of computation of the minimal structure is presented.

The algorithm for Eulerian orientations is very standard. It proceeds by shelling and orienting outer cycles greedily. The algorithms computing bipolar orientations, Schnyder woods, and transversal structures also proceed by augmenting the oriented area, but the principle of augmentation is different. Instead of shelling a cycle, the augmentation is done by pushing a front line; at each step a vertex or face incident to the front line is selected, all its incident edges are oriented, and the front line is pushed further away. In the case of bipolar orientations, this gives rise to a simple algorithm due to de Fraysseix *et al* [**36**], which we recall in Section 3 and reformulate on the associated quadrangulation (the angular map).

Our main contribution is to extend the algorithm computing the minimal Schnyder wood of a triangulation —introduced by Brehm [**25**]— to all 3-connected planar maps, with linear time complexity. A linear algorithm computing a Schnyder wood of a 3-connected map is described in [**3**], but the minimal Schnyder wood is not always reachable. Our algorithm relies on similar principles, suitably modified so as to ouput the minimal Schnyder wood. The proof of correctness is difficult and is delayed to the end of the chapter. Finally, we introduce a new algorithm computing transversal structures, which is easily specified to output the minimal one in linear time. Two other linear algorithms of computation of transversal structures have been introduced by Kant and He [**78**], but they do not seem well adapted to compute the minimal one.

Motivations. Computing the minimal Schnyder wood of a 3-connected map is a key ingredient in two recent important algorithms on planar maps:

(1) The first optimal encoder for 3-connected maps (presented in Chapter 4 Section 2.1), which correspond to the incidences vertices/edges/faces of polygonal meshes with spherical topology.
(2) The currently most compact straight-line drawing algorithm for planar maps, developed by Bonichon *et al* [**16**], and analysed in Chapter 5 Section 3.1.

Similarly, computing the minimal transversal structure is crucial to obtain the first optimal encoder for 4-connected triangulations, which correspond to the incidence relations of triangular meshes with spherical topology and without separating 3-cycle. This application is developed in Chapter 4.

Let us mention that our encoders only deal with the combinatorial part of the meshes (the geometric part, i.e., the coordinates of vertices, is to be encoded separately).

1. Computation of α-orientations

Let us first describe a generic algorithm that, given a graph $M = (V, E)$, tests the feasibility of a function $\alpha : V \to \mathbb{N}$ and outputs an α-orientation if α is feasible, in time $\mathcal{O}(|V|^{3/2})$ if the maximal outdegree is bounded. This algorithm has been explained to me by Stefan Felsner, who describes an alternative algorithm based on flows in [**48**]. Notice that the generic algorithm can be called to compute the combinatorial structures presented in Chapter 1, as these have a formulation as α-orientations. However, for these structures, we will describe more efficient algorithms, which have linear time complexity and have the further advantage of outputing directly the minimal structure (for the distributive lattice).

The algorithm. The idea is to associate a planar map $\widetilde{M}$ to M, such that the existence of an α-orientation in M is equivalent to the existence of a perfect matching in $\widetilde{M}$. The map $\widetilde{M}$ is constructed as follows. Each vertex $v \in M$ gives rise to $\alpha(v)$ copies in $\widetilde{M}$; each edge $e = \{v, w\}$ gives rise to a vertex v_e, called an *edge-vertex*, connected to the vertices of $\widetilde{M}$ spawned from v and w. Hence $\widetilde{M}$ has $|E|$ edge-vertices, has $\sum_{v\in V} \alpha(v)$ vertices spawned from vertices of M, and has $\sum_{e=\{v,w\}\in E} \alpha(v)+\alpha(w)$ edges. Notice that each edge of $\widetilde{M}$ connects an edge-vertex and a vertex spawned from a vertex of M, so that $\widetilde{M}$ is bipartite.

Assume that $\widetilde{M}$ is endowed with a perfect matching X. Then each edge-vertex v_e, which corresponds to an edge $e = \{v, w\}$ of M, is incident to a unique edge of X, this edge being either connected to a copy of v or to a copy of w. In the first (second) case, v (w, respectively) is called the *origin-vertex* of e.

Given a perfect matching X of $\widetilde{M}$, the orientation Y of M such that each edge is oriented outward of its origin-vertex is an α-orientation. Indeed, for each vertex $v \in M$, the copies $v_1, \ldots, v_{\alpha(v)}$ spawned from v are matched with edge-vertices $v_{e_1}, \ldots, v_{e_{\alpha(v)}}$ of $\widetilde{M}$. Hence, v has outdegree $\alpha(v)$, the outgoing edges being $e_1, \ldots, e_{\alpha(v)}$.

Conversely, the same principles ensure that a perfect matching of $\widetilde{M}$ can be computed from an α-orientation of M. This yields the following result.

LEMMA 2.1. *The existence of an α-orientation in M is equivalent to the existence of a perfect matching in $\widetilde{M}$. An α-orientation of M can be computed in linear time from a perfect matching of $\widetilde{M}$.*

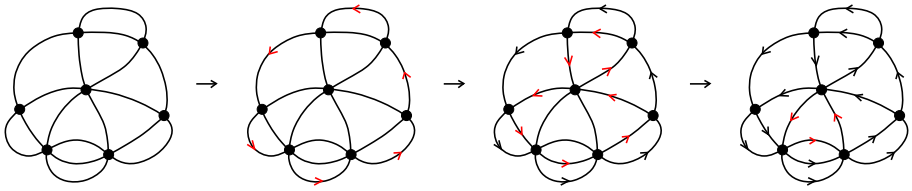


FIGURE 1. Computation of the minimal Eulerian orientation of an Eulerian map.

PROPOSITION 2.1. *Given a map $M = (V, E)$ and a function $\alpha : V \to \mathbb{N}$, testing the feasibility of α and computing an α-orientation of M can be done in time-complexity $\mathcal{O}(|E|^{1/2}m)$, where $m = \sum_{e=\{v,w\}\in E} \alpha(v) + \alpha(w)$. In particular, the complexity is $\mathcal{O}(|V|^{3/2})$ if the maximal outdegree of M has a fixed upper bound.*

Proof. From Lemma 2.1, computing an α-orientation of M reduces to computing a perfect matching of $\widetilde{M}$. An algorithm due to Hopcroft and Karp [**75**] outputs a perfect matching of a bipartite graph (if it exists) in time $\mathcal{O}(|\widetilde{V}|^{1/2}|\widetilde{E}|)$. We have seen that $|\widetilde{E}| = \sum_{e=\{v,w\}\in E} \alpha(v) + \alpha(w)$. Moreover, Euler's relation and the necessary condition $\sum_v \alpha(v) = |E|$ imply that $|\widetilde{V}| = \mathcal{O}(|E|)$. □

2. Computing Eulerian orientations

We describe a simple algorithm to compute the minimal Eulerian orientation of an Eulerian map. We use the well known property that an Eulerian map has no *bridge* (a bridge is an edge having the same face on both sides). The absence of bridge ensures that an edge incident to the outer face on one side is incident to a bounded face on the other side. Hence, the set $E_{\text{out}}(M)$ of edges incident to the outer face of M is partitioned into edge-disjoint simple cycles. In particular, each vertex of M has even degree in $E_{\text{out}}(M)$, hence $M\backslash E_{\text{out}}(M)$ is an Eulerian map. Given all these observations, we introduce the following algorithm of orientation, illustrated in Figure 1.

COMPUTEMINEULERIAN(M):

(1) Orient all edges of $E_{\text{out}}(M)$ with the outer face on their right.
(2) Call COMPUTEMINEULERIAN($M\backslash E_{\text{out}}(M)$).

PROPOSITION 2.2. *Algorithm* COMPUTEMINEULERIAN *outputs the minimal Eulerian orientation of an Eulerian map M and can be implemented to run in linear time.*

Proof. The proof is recursive, as the algorithm. We write shortly $\overline{M} := M\backslash E_{\text{out}}(M)$, $X :=$ COMPUTEMINEULERIAN(M), and $\overline{X} :=$ COMPUTEMINEULERIAN($\overline{M}$). Let us first show that X is Eulerian. The set $E_{\text{out}}(M)$ is partitioned into edge-disjoint simple cycles, which are all oriented counterclockwise at Step 1) of the algorithm COMPUTEMINEULERIAN(M). Hence, each vertex has the same outdegree as indegree in $E_{\text{out}}(M)$. By induction on the number of edges, $\overline{X}$ is Eulerian. Hence, the orientation X, which is obtained as $E_{\text{out}}(M) \cup \overline{X}$, is also Eulerian.

Now we prove recursively that X is equal to the minimal Eulerian orientation $X_{\min}(M)$ of M. The crucial point is that each edge $e = \{v, v'\}$ of $E_{\text{out}}(M)$ has

the outer face on its right in $X_{\min}(M)$ (indeed, any Eulerian orientation is strongly connected due to the existence of an Eulerian tour; if e had the outer face on its left, the cycle formed by e and a path from v' to v would be clockwise, a contradiction.) As a consequence, the edges of $E_{\text{out}}(M)$ have the same orientation in X and in $X_{\min}(M)$. By induction on the number of edges, $\overline{X} = X_{\min}(\overline{M})$. Moreover, $\overline{X}$ is the restriction of X to $\overline{M}$ and $X_{\min}(\overline{M})$ is clearly the restriction of $X_{\min}(M)$ to $\overline{M}$. Hence X and $X_{\min}(M)$ coincide on every edge, i.e., $X = X_{\min}(M)$. Finally, the linear complexity is straightforward; using the half-edge data structure, the outer face of the map formed by the unoriented edges is easily maintained. □

3. Computing bipolar orientations

3.1. Description of the algorithm. Bipolar orientations exist for any rooted 2-connected graph and can be explicitly computed in linear time [**38**]. We focus here on the planar case, i.e., the computation of bipolar orientations of rooted 2-connected maps. In the planar case, a simple algorithm has been introduced by De Fraysseix *et al* [**36**], which we recall here. Given a rooted 2-connected map M with root $e = (s, t)$, the idea is to sweep the map M using an oriented path $\mathcal{P}$ going from s to t, and moving $\mathcal{P}$ progressively to the right. At each step, we denote by $M_{\mathcal{P}}$ the map whose edges are on $\mathcal{P}$ or on the left of $\mathcal{P}$.

To update the path $\mathcal{P}$, we use the concept of *admissible face*. An admissible face f is given by the following property: there exist two vertices v and v' on the contour ∂f of f such that the path $P_1(f) \subset \partial f$ going from v to v' with f on its right is a subpath of $\mathcal{P}$, and the path $P_2(f) \subset \partial f$ going from v to v' with f on its left intersects $\mathcal{P}$ only at v and v'. The update operation consists in choosing an admissible face f, orienting the path $P_2(f)$ from v to v', and replacing $P_1(f)$ by $P_2(f)$ in the path $\mathcal{P}$.

PROPOSITION 2.3. *The orientation algorithm terminates and outputs a bipolar orientation, for any choice of the admissible face at each step.*

Proof. The existence of at least one admissible face at each step is an easy consequence of the absence of separating vertex in M. Hence, the algorithm terminates. It outputs a bipolar orientation, as the following invariant is easily shown to hold all along the computation: "$\mathcal{P}$ is oriented from s to t, the edges of M already oriented are those of $M_{\mathcal{P}}$, and the induced orientation of $M_{\mathcal{P}}$ is bipolar". □

3.2. Computing the minimal bipolar orientation. At each step, the vertices of $\mathcal{P}$ are ordered from left to right, with s as leftmost and t as rightmost vertex. Clearly, admissible faces inherit the left-to-right order.

PROPOSITION 2.4 ([**92**]). *In the algorithm computing bipolar orientations, the choice of the rightmost admissible face at each step yields the minimal bipolar orientation (for the distributive lattice), and the algorithm can be implemented to run in linear time.*

Proof. Omited. The proof of minimality when choosing the rightmost admissible face is given in [**92**]; arguments are close to the ones we will use to prove minimality for our algorithm computing the minimal transversal structure of an irreducible triangulation (see Section 5.2). The linear time complexity is justified in [**36**]; arguments are similar to the ones ensuring that our algorithm computing the minimal transversal structure has linear time complexity (see Section 5.3). □

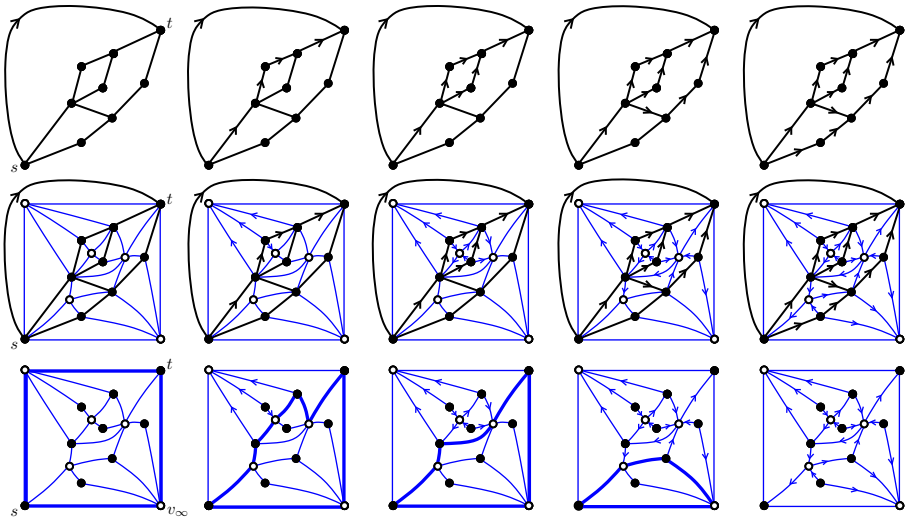


FIGURE 2. Computation of the minimal bipolar orientation of a rooted 2-connected map M (top figure). In the middle figure, the angular map Q is oriented simultaneously. This gives a procedure for computing the minimal 2-orientation of Q, which can be formulated directly on Q by shrinking a cycle (bottom figure).

3.3. Equivalent orientation algorithm on the angular map. Let M be a rooted 2-connected map and Q be its angular map. The generic algorithm presented in Section 3.1 to compute a bipolar orientation of M gives rise to an algorithm $\mathfrak{A}$ computing the associated 2-orientation of Q: we orient each edge e of Q once the two edges of the associated angle of M have been oriented, see Figure 2 (middle). Recall that e is oriented toward its incident black vertex if the angle is extremal, and outward of its incident black vertex if the angle is lateral.

Let us mention that Algorithm $\mathfrak{A}$ can be formalized *directly* as an orientation algorithm on Q using a terminology only dealing with the quadrangulation Q, i.e., without using the underlying bipolar orientation. Write v_∞ for the white vertex of Q corresponding to the outer face of M. A cycle $\mathcal{C}$ of edges of Q containing the outer vertices $\{s, t, v_\infty\}$ is maintained, such that the only separating vertices of $\mathcal{C}$ can be white vertices, and the edges already oriented are those outside of $\mathcal{C}$. A white vertex on $\mathcal{C}\backslash\{v_\infty\}$ is said to be *admissible* if it is not separating and not incident to an internal chord of $\mathcal{C}$ (in fact, an admissible white vertex corresponds to an admissible face of M). At each step, an admissible white vertex v is chosen, the two incident edges of v on $\mathcal{C}$ are oriented outward of v, and all edges incident to v inside $\mathcal{C}$ are oriented toward v, see Figure 3(a)-(b). Then $\mathcal{C}$ is updated, by removing all faces incident to v from the interior of $\mathcal{C}$. Observe that some edges outside of $\mathcal{C}$ might not be oriented yet. Such edges correspond to legs pending from $\mathcal{C}$ and with a black extremity outside of $\mathcal{C}$. We orient these edges outward of their black vertex, see Figure 3(b)-(c), where there are two such legs. According to Proposition 2.4, the choice of the rightmost admissible white vertex at each step yields the minimal 2-orientation of Q. To sum up, the principle of the algorithm is to push a *front-line* —the path $\mathcal{C}\backslash\{v_\infty\}$— by choosing at each step a so-called admissible vertex v on the front-line and orienting the edges incident to v. Moreover, the choice of

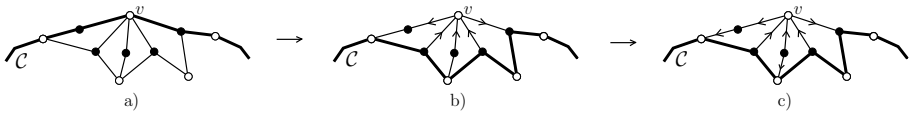


FIGURE 3. A generic step of the algorithm computing a 2-orientation of a quadrangulation.

the rightmost admissible vertex on the front line at each step yields the minimal 2-orientation. These general features are common to all algorithms of orientation presented in the sequel of this chapter, for the computation of Schnyder woods in Section 4 and of transversal structures in Section 5.

4. Computing Schnyder woods

The main contribution of this chapter is a linear algorithm computing the minimal Schnyder wood of a 3-connected map. Our starting point is an algorithm introduced by Brehm to compute the minimal Schnyder wood of a triangulation [**25**]. We first recall the algorithm of Brehm, which is simple and captures the main ideas that are used in the extension to 3-connected maps.

4.1. Computing Schnyder woods of triangulations. The algorithm of Brehm follows the same lines as the algorithm computing the minimal 2-orientation of a quadrangulation, sketched in Section 3.3. It proceeds iteratively; the area not oriented yet is delimited by a cycle and shrinks progressively. The difference is that no vertex bicoloration has to be taken into account, which makes the algorithm presented here simpler.

The algorithm. Let T be a triangulation, with a_1, a_2, and a_3 the outer vertices in clockwise order. The principle of the algorithm is to maintain a simple cycle $\mathcal{C}$ containing the edge (a_2, a_3), called the *base edge*. We always imagine the triangulation as drawn in the plane with the base edge down and the opposite vertex a_1 on top. The vertices of $\mathcal{C}$ are ordered from left to right, with a_3 as leftmost and a_2 as rightmost vertex. A vertex of $\mathcal{C}\backslash\{a_2, a_3\}$ is said to be admissible if it is not incident to an internal chord of $\mathcal{C}$. The cycle $\mathcal{C}$ is initialized with a_2, a_3, and all inner neighbours of a_1. In addition, we label 1 the inner edges incident to a_1 and orient them toward a_1, see Figure 4(b).

At each step, an admissible vertex v is chosen. We orient outward the two edges incident to v on $\mathcal{C}$, giving label 2 to the edge going to the right and label 3 to the edge going to the left; and we orient inward all edges incident to v inside $\mathcal{C}$, giving label 1 to all these edges. The cycle $\mathcal{C}$ is updated by removing from the interior of $\mathcal{C}$ all faces incident to v. As v is not incident to any internal chord, the update of $\mathcal{C}$ is either a simple cycle or is reduced to the base-edge (a_2, a_3). Moreover, as the map is a triangulation, it is easily shown that there exists an admissible vertex at each step, i.e., the algorithm terminates, the cycle $\mathcal{C}$ being shrinked until it is reduced to the base-edge, see Figure 4.

PROPOSITION 2.5 ([**25**]). *Given a triangulation T, the algorithm of orientation outputs a Schnyder wood of T, for any choice of the admissible vertex at each step.*

Proof. The following invariants are maintained all along the computation:

(1) The edges already oriented are those outside of $\mathcal{C}$.

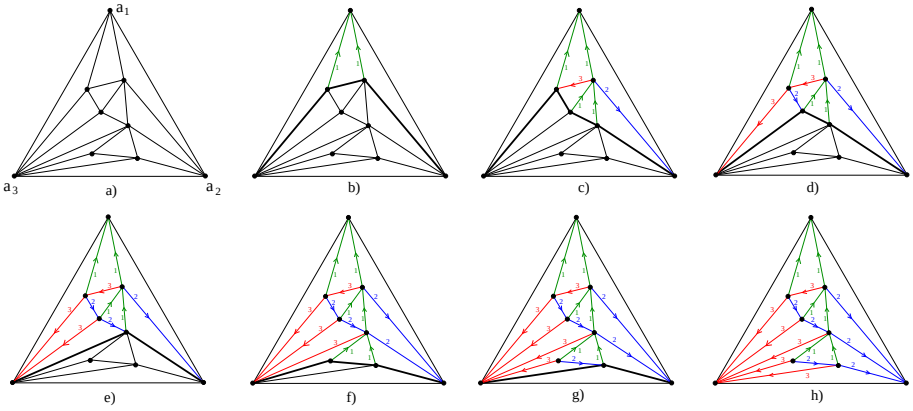


FIGURE 4. Computation of the minimal Schnyder wood of a triangulation.

(2) Each vertex outside of $\mathcal{C}$ satisfies the Schnyder condition.
(3) Each vertex v on $\mathcal{C}$ satisfies the following partial version of Schnyder condition (see Figure 6(b)); one edge incident to v outside of $\mathcal{C}$ is outgoing, and has label 1; all edges incident to v between the outgoing edge and the edge connected to the left neighbour of v are ingoing with label 2, and all edges incident to v between the outgoing edge and the edge connected to the right neighbour of v are ingoing with label 3.

At the end, $\mathcal{C}$ is reduced to the base-edge (a_2, a_3) and the invariants are true; hence, the structure computed is a Schnyder wood, as illustrated in Figure 4.

PROPOSITION 2.6 ([**25**]). *Given a triangulation T, the choice of the rightmost admissible vertex at each step yields the minimal Schnyder wood of T. In that case, the algorithm can be implemented to run in linear time.*

Proof. Let X be the Schnyder wood computed by choosing the rightmost admissible vertex at each step. Recall that essential circuits of Schnyder woods are circuits of length 3 (Proposition 1.7), so the presence of a clockwise circuit in X implies the presence of a clockwise circuit of length 3. Assume there exists a clockwise triangle $\Delta = (x, y, z)$. As shown in the proof of Proposition 1.7, no edge inside Δ has its origin on Δ. Hence, the Schnyder conditions ensure that the edges of Δ in clockwise order have labels $\{1, 2, 3\}$. We assume that the edge with label 1 is (z, x). Clearly, the labels of (x, y), (y, z), and (z, x) imply that x is treated before y and z. Consider the step when x is treated (see Figure 5), and let $\mathcal{C}_0$ be the state of the cycle at this step. The right neighbour of x on $\mathcal{C}_0$ is the extremity of the edge going out of x with label 2, i.e., is the vertex y. As y is on the right of the rightmost admissible vertex x, y is not admissible, hence is incident to an internal chord. Take e as the internal chord of $\mathcal{C}_0$ incident to y and whose other extremity w is leftmost possible. Consider the (closed) area A delimited by $\mathcal{C}_0 \backslash \{a_2, a_3\}$ and the chord e. As T is triangulated, it is easily shown that the area A contains at least one admissible vertex. This property ensures that w is on the left of x on the cycle (otherwise, the admissible vertex in A would be on the right of x, a contradiction). In particular, the vertex x is in A. As (z, x) has label 1, the vertex z is inside $\mathcal{C}_0$ when x is treated, hence is above the chord $\{z, w\}$, see Figure 5. Clearly, e will cease to be an internal chord once all vertices in $A \backslash \{y, w\}$ are treated. At this

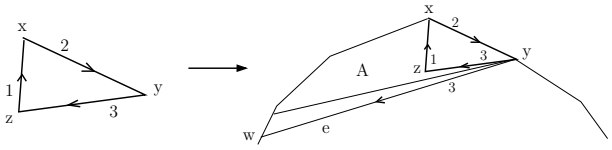


FIGURE 5. The presence of a clockwise 3-cycle yields the contradiction that a vertex has two outgoing edges with label 3.

step, all vertices on the right of y are still blocked, so that y will be the rightmost admissible vertex at this step, with w as left neighbour on $\mathcal{C}$. Hence, the outgoing edge of y with label 3 is (y, w), i.e., $w = z$. This contradicts the fact that z is above $\{y, w\}$. Finally, the linear time complexity will be justified for the extended algorithm computing the minimal Schnyder wood of a 3-connected map, see the paragraph on implementation page 72. □

4.2. Extension to 3-connected maps. We describe in this section a linear-time algorithm to compute the minimal Schnyder wood of an outer-triangular 3-connected map G. Recall that the minimal Schnyder wood of G corresponds by definition to the minimal α_3-orientation of the derived map G'. In addition, using the rules illustrated in Figure 13, the minimal Schnyder wood of G is mapped to the minimal Schnyder wood of its dual map G^*. As any suspended internally 3-connected map is the dual of an outer-triangular 3-connected map, restricting our attention to outer-triangular 3-connected maps is sufficient in order to compute the minimal Schnyder wood of any suspended internally 3-connected map.

In itself our algorithm for 3-connected maps is only slightly more involved than the algorithm for triangulations, and is close to an algorithm due to Di Battista [**3**], which computes a Schnyder wood of a 3-connected map, but might not reach the minimal one. The correctness proof is much more involved than for triangulations (see the discussion in the beginning of Section 6).

4.2.1. *Principle of the algorithm.* Let G be an outer-triangular 3-connected planar map and let G' be its derived map and G^* its dual map. We denote by a_1, a_2 and a_3 the outer vertices of G in clockwise order around G. The idea for the orientation algorithm is again to maintain a simple cycle of edges of G. At each step k, the cycle, denoted by $\mathcal{C}_k$, is shrinked by choosing a so-called *admissible* vertex v on $\mathcal{C}_k$, and by removing from the interior of $\mathcal{C}_k$ all faces incident to v. The admissible vertex is always different from a_2 or a_3, so that the edge (a_2, a_3), called *base-edge*, is always on $\mathcal{C}_k$. The edges of G' ceasing to be on $\mathcal{C}_k$ or in the interior of $\mathcal{C}_k$ are oriented so that the following invariants remain satisfied.

Orientation invariants:

- For each edge e of G outside of $\mathcal{C}_k$, the 4 edges of G' incident to the edge-vertex v_e associated to e have been oriented at a step $j < k$ and v_e has outdegree 1.
- All other edges of G' are not oriented yet.

Moreover, the edges of G' that correspond to half-edges of G also receive a label in $\{1, 2, 3\}$, so that the following invariants for labels remain satisfied:

Label invariants:

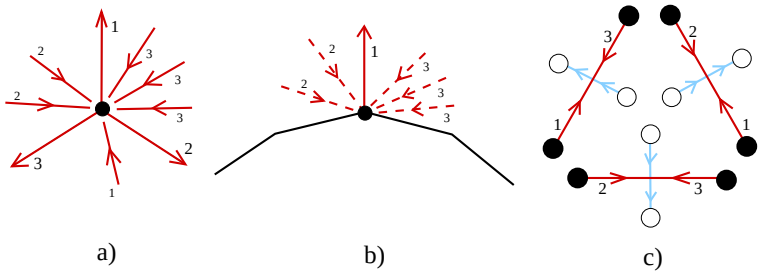


FIGURE 6. The invariants for the labels of the half-edges of G maintained during the algorithm.

- At each step k, every vertex v of G outside of $\mathcal{C}_k$ has one outgoing half-edge for each label 1, 2 and 3 and these outgoing edges appear in clockwise order around v. In addition, all edges between the outgoing edges with labels i and $i+1$ are incoming with label $i-1$, see Figure 6(a).
- Let v be a vertex of G on $\mathcal{C}_k$ having at least one incident edge of G' outside of $\mathcal{C}_k$. Then exactly one of these edges, denoted by e'_1, is going out of v. In addition it has label 1. The edges of G' incident to v and between e'_1 and its left neighbour on $\mathcal{C}_k$ are incoming with label 2; and the edges incident to v in G' between e'_1 and its right neighbour on $\mathcal{C}_k$ are incoming with label 3, see Figure 6(b).
- For each edge e of G outside of $\mathcal{C}_k$, let e' be the unique outgoing edge of its associated edge-vertex v_e. Two cases can occur:
  - If e' is a half-edge of G then the two edges of G' incident to v_e and forming the edge e are identically labelled. This corresponds to the case where e is "simply oriented".
  - If e' is a half-edge of G^*, we denote by $1 \leq i \leq 3$ the label of the edge of G' following e' in clockwise order around v_e. Then the edge of G' following e' in counter-clockwise order around v_e is labelled $i+1$, see Figure 6(c). This corresponds to the case where e is "bi-oriented".

Actually, these labels, which are the labels of the Schnyder wood of G, are not needed to perform the algorithm of orientation, but they will be very useful to prove that it really computes the minimal α_3-orientation.

In the following, we write G_k for the submap of G obtained by removing all vertices and edges outside of $\mathcal{C}_k$ (at step k). In addition, we order the vertices of $\mathcal{C}_k$ from left to right, with a_3 as leftmost vertex and a_2 as rightmost vertex. In other words, a vertex $v \in \mathcal{C}_k$ is on the left of a vertex $v' \in \mathcal{C}_k$ if the path of $\mathcal{C}_k$ going from v to v' without passing by the edge (a_2, a_3) has the interior of $\mathcal{C}_k$ on its right.

4.2.2. *Description of the main iteration.* Let us now describe the kth step of the algorithm, during which the cycle $\mathcal{C}_k$ is shrinked so that the invariants for orientation and labelling remain satisfied. The description requires some definitions.

Definitions. A vertex of $\mathcal{C}_k$ is said to be *active* if it is incident to at least one edge of $G \backslash G_k$. Otherwise, the vertex is said to be *passive*. By convention, before the first step of the algorithm, the vertex a_1 is considered as active and its incident half-edge directed toward the infinite face is labelled 1.

For each pair of vertices v_1, v_2 of $\mathcal{C}_k$ ordered so that v_1 is on the left of v_2, the path on $\mathcal{C}_k$ going from v_1 to v_2 without passing by the edge (a_2, a_3) is denoted by $[v_1, v_2]$. We also use the notation $]v_1, v_2[= [v_1, v_2] \backslash \{v_1, v_2\}$.

A pair (v_1, v_2) of vertices of $\mathcal{C}_k$ is *separating* if there exists an inner face f of G_k such that v_1 and v_2 are incident to f but the edges of $[v_1, v_2]$ are not all incident to f. Such a face is called a *separating face* and the triple (v_1, v_2, f) is called a *separator*. The area delimited by the border path $[v_1, v_2]$ and by the path of edges of f going from v_1 to v_2 with the interior of f on its right is called the *separated area* of (v_1, v_2, f) and denoted by $\mathrm{Sep}(v_1, v_2, f)$.

A vertex v on $\mathcal{C}_k$ is said to be *blocked* if it belongs to a separating pair. It is easily checked that a vertex is blocked iff it is incident to a separating face of G_k. In particular, a non blocked vertex does not belong to any separating pair of vertices. By convention, the vertices a_2 and a_3 are always considered as blocked. A vertex v on $\mathcal{C}_k$ is *admissible* if it is active and not blocked.

Finally, for each vertex v of $\mathcal{C}_k$, we define its *left-connection vertex* $\mathrm{left}(v)$ as the leftmost vertex on $\mathcal{C}_k$ such that the vertices of $]\mathrm{left}(v), v[$ all have only two incident edges in G_k. The path $[\mathrm{left}(v), v]$ is called the *left-chain* of v and the first edge of $[\mathrm{left}(v), v]$ is called the *left-connection edge* of v. Similarly, we define the *right-connection vertex*, the *right-chain*, and the *right-connection edge* of v. An important remark is that all vertices of $]\mathrm{left}(v), v[$ and of $]v, \mathrm{right}(v)[$ are active. Indeed, these vertices have two incident edges in G_k but they have at least 3 incident edges in G because G is 3-connected. Hence, they have at least one incident edge in $G \backslash G_k$.

Operations of step k. First, we choose the rightmost admissible vertex of $\mathcal{C}_k$ and we call $v^{(k)}$ this vertex. (We will prove in Lemma 2.3 that there always exists an admissible vertex on $\mathcal{C}_k$ as long as G_k is not reduced to the edge (a_2, a_3).) Notice that this admissible vertex can not be a_2 nor a_3 because a_2 and a_3 are blocked.

We denote by $f_1, \ldots, f_m$ the bounded faces of G_k incident to $v^{(k)}$ from right to left. We also denote by $e_1, \ldots, e_{m+1}$ the edges of G_k incident to $v^{(k)}$ from right to left. Hence, for each $1 \leq i \leq m$, f_i corresponds to the sector between e_i and e_{i+1}.

An important remark is that the right-chain of $v^{(k)}$ is reduced to one edge. Indeed, if there exists a vertex v in $]v^{(k)}, \mathrm{right}(v^{(k)})[$, then v is active as already explained. In addition, v is incident to only one inner face of G_k, which is f_1. As f_1 is incident to $v^{(k)}$ and as $v^{(k)}$ is non blocked, f_1 is not separating. Hence v is not blocked. Thus v is admissible and is on the right of $v^{(k)}$, in contradiction with the fact that $v^{(k)}$ is the rightmost admissible vertex on $\mathcal{C}_k$.

We perform the following operations on the edges of G' incident to the edge-vertices on the left-chain of $v^{(k)}$ and on the edges $e_1, \ldots e_m$, see Figure 7:

- **Inner edges**: For each edge e_i with $2 \leq i \leq m$, we denote by v_{e_i} the corresponding edge-vertex of G'. Orient the two edges of G' forming e_i toward $v^{(k)}$ and give label 1 to these two edges. Orient the two other incident edges of v_{e_i} toward v_{e_i}, so that v_{e_i} has outdegree 1.
- **Left-chain**: For each edge e of the left-chain of $v^{(k)}$ —traversed from $v^{(k)}$ to $\mathrm{left}(v^{(k)})$— different from the left-connection edge, bi-orient e and give label 3 (resp. label 2) to the first (resp. second) traversed half-edge. Choose the unique outgoing edge of the edge-vertex v_e associated to e to be the edge going out of e toward the interior of $\mathcal{C}_k$
- **Left-connection edge**: If $\mathrm{left}(v^{(k)})$ is passive, bi-orient the left-connection edge e of $v^{(k)}$, give label 1 to the half-edge incident to $\mathrm{left}(v^{(k)})$ and label 3 to the other half-edge, and choose the unique outgoing edge of the

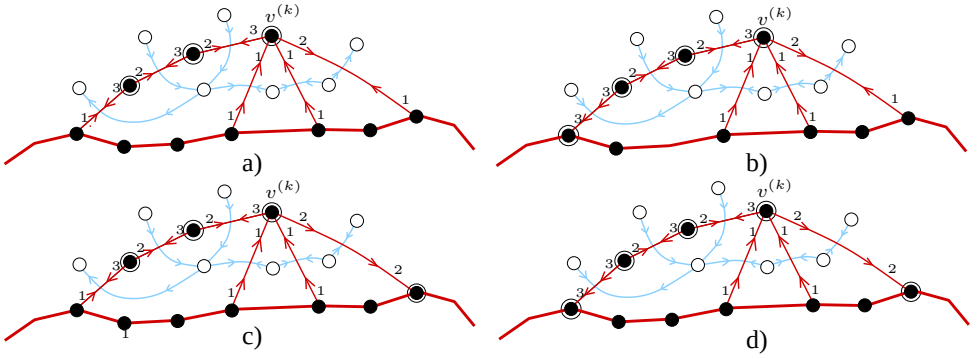


FIGURE 7. The operations performed at step k of the algorithm, whether left$(v^{(k)})$ and right$(v^{(k)})$ are passive-passive (Fig. a) or active-passive (Fig. b) or passive-active (Fig. c) or active-active (Fig. d). Active vertices are surrounded.

edge-vertex v_e to be the edge going out of v_e toward the exterior of $\mathcal{C}_k$. If left$(v^{(k)})$ is active, label 3 and orient toward left$(v^{(k)})$ the two edges of G' forming e, and orient the two other incident edges of v_e toward v_e.

- **Right-connection edge**: The edge e_1, which is the right-connection edge of $v^{(k)}$, is treated symmetrically as the left-connection edge. If right$(v^{(k)})$ is passive, bi-orient e_1, give label 1 to the half-edge incident to right$(v^{(k)})$ and label 2 to the other half-edge, and choose the unique outgoing edge of the edge-vertex v_{e_1} to be the edge going out of v_{e_1} toward the exterior of $\mathcal{C}_k$. If left$(v^{(k)})$ is active, label 2 and orient toward right$(v^{(k)})$ the two edges of G' forming e_1, and orient the two other incident edges of v_{e_1} toward v_{e_1}.

After these operations, all faces incident to $v^{(k)}$ are removed from the interior of $\mathcal{C}_k$, producing a (shrinked) cycle $\mathcal{C}_{k+1}$. As a_2 and a_3 are blocked on $\mathcal{C}_k$, $\mathcal{C}_{k+1}$ still contains the edge (a_2, a_3). In addition, if $\mathcal{C}_{k+1}$ is not reduced to (a_2, a_3), the property of 3-connectivity of G and the fact that the chosen vertex $v^{(k)}$ is not incident to any separating face easily ensure that $\mathcal{C}_{k+1}$ is a simple cycle, i.e., it does not contain any separating vertex.

It is also easy to get convinced from Figure 6 and Figure 7 that the operations performed at step k maintain the invariants of orientation and labeling.

The purpose of the next two lemmas is to prove that the algorithm terminates.

LEMMA 2.2. *Let (v_1, v_2, f) be a separator on $\mathcal{C}_k$. Then there exists an admissible vertex in $]v_1, v_2[$.*

Proof. Let (v_1', v_2', f') be a minimal separator for the (non empty) set of separators whose separated area is included or equal to the separated area of (v_1, v_2, f) (minimality being with respect to the inclusion-relation on the separated areas). Observe that v_1' and v_2' are in $[v_1, v_2]$.

Assume that no vertex of $]v_1', v_2'[$ is active. Then the removal of v_1' and v_2' disconnects $\mathrm{Sep}(v_1', v_2', f)$ from $G \backslash \mathrm{Sep}(v_1', v_2', f)$. This is in contradiction with 3-connectivity of G, because these two sets are easily proved to contain at least one vertex different from v_1' and v_2'.

Hence, there exists an active vertex v in $]v'_1, v'_2[$, also in $]v_1, v_2[$. If v was incident to a separating face, this face would be included in the separated area of (v'_1, v'_2, f'), which is impossible by minimality of (v'_1, v'_2, f'). Hence, the active vertex v is not blocked, i.e., is admissible.

LEMMA 2.3. *As long as $\mathcal{C}_k$ is not reduced to (a_2, a_3), there exists an admissible vertex on $\mathcal{C}_k$.*

Proof. Assume that there exists no separating pair of vertices on $\mathcal{C}_k$. In this case, an active vertex on $\mathcal{C}_k$ different from a_2 and a_3 is admissible. Hence we just have to prove the existence of such a vertex. At the first step of the algorithm, there exists an active vertex on $\mathcal{C}_1 \backslash \{a_2, a_3\}$ because a_1 is active by convention. At any other step, there exists an active vertex on $\mathcal{C}_k \backslash \{a_2, a_3\}$, otherwise the removal of a_2 and a_3 would disconnect $G_k \backslash \{a_2, a_3\}$ from $G \backslash G_k$, in contradiction with 3-connectivity of G.

If there exists at least one separator (v_1, v_2, f), Lemma 2.2 ensures that there exists an admissible vertex v in $]v_1, v_2[$.

Last step of the algorithm. Lemma 2.3 implies that, at the end of the iterations, only the edge $e = \{a_2, a_3\}$ remains. To complete the orientation, bi-orient e and label 3 (resp. label 2) the half-edge of e whose origin is a_2 (resp. a_3); the outgoing edge of the edge-vertex v_e (associated to e) is chosen to be the edge going out of v_e toward the infinite face. We also label respectively 2 and 3 the half-edges incident to a_2 and a_3 and directed toward the infinite face.

Figure 8 illustrates the execution of the algorithm on an example. On this figure, the edges of $\mathcal{C}_k$ are black and bolder. In addition, the active vertices are surrounded and the rightmost admissible vertex is doubly surrounded.

THEOREM 2.1. *The algorithm computes the minimal α_3-orientation of the derived map.*

Section 6 is dedicated to the proof of this theorem.

Remark. As stated in Theorem 2.1, our orientation algorithm computes a particular α_3-orientation, namely the minimal one. The absence of clockwise circuit is due to the fact that among all admissible vertices, the rightmost one is chosen at each step. The algorithm is easily adapted to other choices of admissible vertices: the only difference is that the right-connection chain of the chosen admissible vertex might not be reduced to an edge, in which case it must be dealt with in a symmetric way as the left-connection chain (that is, 2 becomes 3 and left becomes right in the description of edge labellings and orientations). This yields a "generic" algorithm that can produce any α_3-orientation of G'. Indeed, given a particular α_3-orientation X of G', it is easy to prove the existence of a scenario (i.e., a suitable choice of the admissible vertex at each step) that outputs X. In contrast, some α_3-orientations of G' —in particular the minimal one— might be missed by the algorithm developed by Di Battista *et al* [**3**]. Their algorithm is close to ours, however the operations allowed at each step are more limited; with our terminology, each update is either such that the left-connection chain is reduced to one edge or such that the chosen vertex $v^{(k)}$ is incident to a unique inner face of G_k.

Remark. Observe that the triple made of a_2, a_3, and the inner face of G incident to (a_2, a_3) is a separator until the last-but-one step of the algorithm. Only at this particular step, we need to state that a_2 is blocked by convention, so that only the edge (a_2, a_3) remains at the last step. In the following proofs (in particular that of Lemma 2.8), a blocked vertex (even a_2 and a_3) will always be assumed to belong to

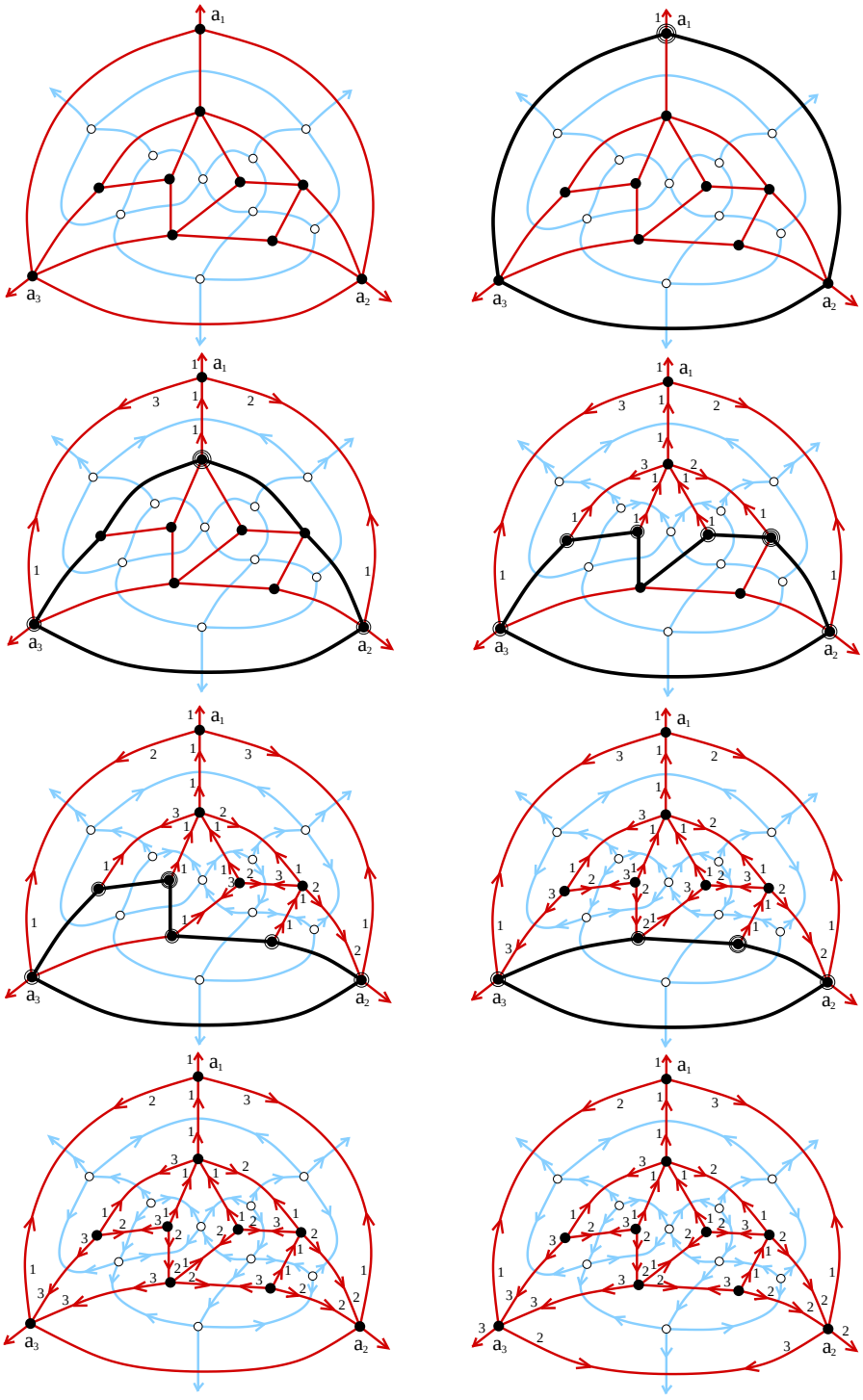


FIGURE 8. Execution of the algorithm computing the minimal Schnyder wood of an outer-triangular 3-connected map.

a separator and we leave to the reader the easy verification that the proofs remain correct by dealing with the particular case of the last but one step.

Implementation. Following [**77**] (see also [**25**] for the case of triangulations), an efficient implementation is obtained by maintaining, for each vertex $v \in \mathcal{C}_k$, the number $s(v)$ of separating faces incident to v. Thus, a vertex is blocked iff $s(v) > 0$. Notice that a face f is separating iff the numbers $v(f)$ and $e(f)$ of vertices and edges (except (a_2, a_3)) incident to f and belonging to $\mathcal{C}_k$ verify $v(f) > e(f) + 1$. Thus, it is easy to test if a face is separating, so that the parameters $s(v)$ are also easily maintained. The data structure we use is the half-edge structure, which allows us to navigate efficiently in the graph. The *pointer* is initially on a_1, which is the rightmost admissible vertex at the first step. During the execution, once the vertex $v^{(k)}$ is treated, the pointer is moved to v the right neighbour of $v^{(k)}$ on C_k. The crucial point is that, if v is blocked, then no vertex on the right of v can be admissible (because of the nested structure of separating faces). Thus, in this case, the pointer is moved to the left until an admissible vertex is encountered. Notice also that v is active after $v^{(k)}$ is treated. Thus, if v is not blocked, then v is admissible at step $k + 1$. In this case, the nested structure of separating faces ensures that the rightmost admissible vertex at step $k + 1$, if not v, is either the right-connection vertex v' of v, or the left neighbour of v' on C_{k+1} (in the case where v' is not admissible). Notice that, in the case where v is not blocked, the pointer is moved to the right but the edges traversed will be immediately treated (i.e., removed from C_{k+1}) at step $k+1$. This ensures that an edge can be traversed at most twice by the pointer: once from right to left and later once from left to right. Thus, the complexity is linear.

5. Computing transversal structures

Let us now describe a simple iterative algorithm to compute the minimal transversal structure of an irreducible triangulation. Two different algorithms computing such transversal structures were already presented in [**78**]. The advantage of our algorithm is that it is easily specified to output the minimal transversal structure for the distributive lattice described in Section 6. We will need the minimal transversal structure in the encoding algorithm for 4-connected triangulations (Section 2.2) and analysing the straight-line drawing algorithm for irreducible triangulations (Section 2.3).

5.1. Description of the algorithm. As the previously presented algorithms of orientation, we proceed iteratively by augmenting the area oriented. The principle is to maintain and iteratively shrink a cycle $\mathcal{C}$ of edges of T such that the following invariants are satisfied at each step:

- The cycle $\mathcal{C}$ contains the two edges (S, W) and (S, E).
- No edge interior to $\mathcal{C}$ connects two vertices of $\mathcal{C}\backslash\{S\}$
- All inner edges of T outside of $\mathcal{C}$ are colored and oriented such that Condition C1 is satisfied for each inner vertex of T outside of $\mathcal{C}$.
- For each inner vertex v of T on $\mathcal{C}$, a partial version of C1 holds: in clockwise order around v, the edges incident to v and exterior to $\mathcal{C}$ form: a (possibly empty) interval of ingoing blue edges, a non empty interval of outgoing red edges, and a (possibly empty) interval of outgoing blue edges.

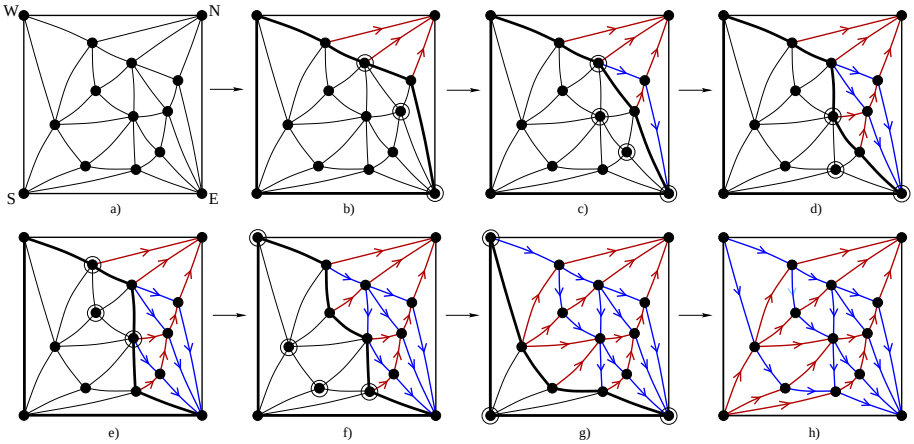


FIGURE 9. An example of execution of the algorithm computing the minimal transversal structure. Vertices of the rightmost admissible path are surrounded.

We initialise the cycle $\mathcal{C}$ with vertices S, W, E and all inner neighbours of N, color red all inner edges incident to N and orient them toward N, see Figure 9(b). Observe that the vertices of $\mathcal{C}$ different from S can be ordered from left to right with W as leftmost and E as rightmost vertex. For two vertices v and v' of $\mathcal{C}\backslash\{S\}$ with v on the left of v', we write $[v, v']$ for the unique path on $\mathcal{C}$ that goes from v to v' without passing by S.

To explain how to update (shrink) $\mathcal{C}$ at each step, we need a few definitions. An *internal path* of $\mathcal{C}$ is a path $\mathcal{P}$ of connecting two vertices of $\mathcal{C}$ via vertices inside $\mathcal{C}$. We write $\mathcal{C}_\mathcal{P}$ the concatenation of $\mathcal{P}$ and $[v, v']$. The path $\mathcal{P}$ is said to be *admissible* if the following conditions are satisfied:

- The paths $\mathcal{P}$ and $[v, v']$ have both at least two edges.
- Each edge interior to $\mathcal{C}_\mathcal{P}$ connects a vertex of $\mathcal{P}\backslash\{v, v'\}$ to a vertex of $[v, v']\backslash\{v, v'\}$. In particular, the interior of $\mathcal{C}_\mathcal{P}$ contains no vertex.
- The cycle $\mathcal{C}'$ obtained from $\mathcal{C}$ by replacing $[v, v']$ by $\mathcal{P}$ is such that no interior edge of $\mathcal{C}'$ connects two vertices of $\mathcal{C}'\backslash\{S\}$.

The update operation is the following: find an admissible internal path $\mathcal{P}$ of $\mathcal{C}$ and write v and v' for its left extremity and right extremity on $\mathcal{C}$; then, color each internal edge of $\mathcal{C}_\mathcal{P}$ in red and orient it toward $[v, v']\backslash\{v, v'\}$. Color all edges of $[v, v']$ in blue and orient them from v to v'; finally update $\mathcal{C}$ by replacing in $\mathcal{C}$ the path $[v, v']$ by the path $\mathcal{P}$.

It is shown in Lemma 2.4 (which uses the absence of separating 3-cycles of T) that the algorithm terminates, i.e., that at each step the cycle $\mathcal{C}$ has an admissible internal path and can be updated (shrinked). After the last update operation, $\mathcal{C}$ is empty and the invariants are satisfied, which implies that the obtained orientation and coloration of inner edges of T is a transversal structure. Figure 9 illustrates the complete execution of the algorithm on an example.

This generic algorithm can easily be adapted to obtain an algorithm, called COMPUTEMINIMAL(T), which computes the minimal transversal structure. Observe that, at each step of the algorithm, admissible paths of $\mathcal{C}$ can be ordered from left to right, by saying that $\mathcal{P}_1 \geq \mathcal{P}_2$ if the left extremity and the right extremity

of $\mathcal{P}_1$ are (weakly) on the left respectively of the left extremity and of the right extremity of $\mathcal{P}_2$. Although this order is only partial, it can easily be shown to admit a unique minimum, called rightmost admissible path of $\mathcal{C}$. Algorithm ComputeMinimal(T) consists in choosing the rightmost admissible path at each step of the iterative algorithm described above, see also Figure 9, where the execution respects this choice.

Proposition 2.7. *Given an irreducible triangulation T,* ComputeMinimal*(T) outputs the minimal transversal structure of T. In addition,* ComputeMinimal*(T) can be implemented to run in linear time.*

The next section focuses on the proof that the computed transversal edge-partition is the minimal one. Then, Section 5.3 details how the implementation can be done with a linear time complexity.

5.2. Correctness of the algorithm. First we prove in Lemma 2.4 that the generic algorithm terminates, hence it outputs a transversal structure. Then we prove in Proposition 2.7' that the transversal structure given by ComputeMinimal(T) is minimal, i.e., has no alternating 4-cycle.

Lemma 2.4. *At each step of the generic algorithm, the cycle $\mathcal{C}$ has an admissible path. Hence the algorithm terminates.*

Proof. If there is no vertex interior to $\mathcal{C}$, we are at the last step of the algorithm, represented on Figure 9(g). At this step, the path made of the two edges (W, S) and (S, E) is clearly admissible.

If there is a vertex interior to $\mathcal{C}$, we define a 2-chord of $\mathcal{C}$ as a path of two edges inside $\mathcal{C}$ connecting two vertices v and v' of $\mathcal{C}\backslash\{S\}$. Given an internal path $\mathcal{P}$, we call *associated cycle* of $\mathcal{P}$ the cycle made of the concatenation of $\mathcal{P}$ and of $[v, v']$, where v and v' are respectively the left and the right extremity of $\mathcal{P}$. A 2-chord is said *separating* if the interior of its associated cycle contains at least one vertex. Observe that the set of separating 2-chords is not empty because the associated cycle of the 2-chord (W, S, E) is the whole cycle $\mathcal{C}$, whose interior has at least one vertex. Let $\mathcal{P}$ be a minimal separating 2-chord, i.e., the interior of the associated cycle $\mathcal{C}(\mathcal{P})$ of $\mathcal{P}$ does not contain the interior of the associated cycle of any other separating 2-chord. Let w_l and w_r be the left and right extremity of $\mathcal{P}$. Observe that $[w_l, w_r]\backslash\{w_l, w_r\}$ contains at least one vertex, otherwise $\mathcal{C}(\mathcal{P})$ would be a separating 3-cycle. Consider a vertex v of $[w_l, w_r]\backslash\{w_l, w_r\}$. Let $v_1, \dots, v_k$ be the neighbours of v in the interior of $\mathcal{C}(\mathcal{P})$ and in counterclockwise order around v. It is easy to see that $v_1, \dots, v_k$ are in the interior of $\mathcal{C}(\mathcal{P})$, by minimality of $\mathcal{P}$. Let e_{left} be the leftmost incident edge of v_1 connected to a vertex of $\mathcal{C}\backslash\{S\}$ and e_{right} be the rightmost incident edge of v_k connected to a vertex of $\mathcal{C}\backslash\{S\}$. Let v_{left} and v_{right} be the respective extremities of e_{left} and e_{right} on $\mathcal{C}\backslash\{S\}$. Consider the path $\mathcal{P}_v$ made of the concatenation of e_{left}, of the chain connecting $v_1 \to v_2 \dots \to v_k$, and of e_{right}, see Figure 10.

We now prove that $\mathcal{P}_v$ is admissible, i.e., that it satisfies the three conditions defined in Section 5.1. The first condition is clearly satisfied. If the second condition was not satisfied, then $v_{\text{left}} \to v_1 \to v$ or $v \to v_k \to v_{\text{right}}$ would be a separating 2-chord, contradicting the minimality of $\mathcal{P}$. Assume that the third condition is not satisfied. Then the cycle $\mathcal{C}'$ obtained from $\mathcal{C}$ by replacing $[v_{\text{left}}, v_{\text{right}}]$ by $\mathcal{P}_v$ contains an internal chord e. As $\mathcal{C}$ contains no internal chord, e is incident to at least one of the interior neighbours of v and is exterior to the cycle delimited by $\mathcal{P}_v$ and by

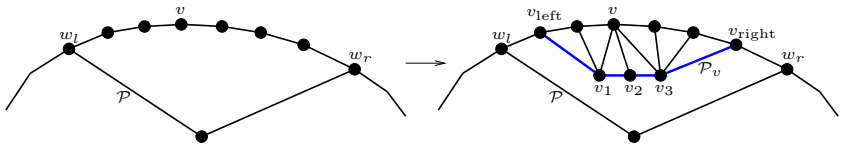


FIGURE 10. There exists an admissible internal path in the interior of a cycle associated to a separating 2-chord.

$[v_{\text{left}}, v_{\text{right}}]$. In fact, e can not connect two interior neighbours of v. Otherwise, let i and j be the indices of these two neighbours, with $i < j$. First observe that $i < j+1$ because v_i and v_{i+1} are already adjacent on $\mathcal{P}_v$ and a triangulation has no multiple edges. Thus the 3-cycle made of e, $\{v, v_i\}$ and $\{v, v_j\}$ is separating because its interior contains the vertex v_{i+1}. Hence e connects an interior neighbour v_i (with $1 \leq i \leq k$) of v to a vertex $\overline{v}$ on $\mathcal{C}\backslash\{S\}$. Then two cases can occur: if $\overline{v}$ is on the left or equal to v_{left}, then the origin v_i of e can not be v_1 because the leftmost edge connecting connecting v_1 to $\mathcal{C}$ is e_{left}, which belongs to $\mathcal{P}_v$. Hence the 2-chord made of e and of $\{v_i, v\}$ is separating because its associated cycle contains the vertex v_1. This is in contradiction with the minimality of $\mathcal{P}$. Similarly the case where e is on the right or equal to v_{right} is excluded. Finally, $\mathcal{P}_v$ is admissible, so that there exists at least one admissible path.

Now we concentrate on the proof that the transversal structure computed by COMPUTEMINIMAL(T) is minimal.

LEMMA 2.5. *Let X be a transversal structure of T and let $\mathcal{P}$ be an oriented blue path connecting W to E. Then there can be no edge exterior to $\mathcal{P}$ connecting two vertices of $\mathcal{P}$.*

Proof. Condition C1 of transversal structures ensures that such an edge e has to be colored blue. By acyclicity of the orientation of the blue edges, e has to be oriented in the parallel direction of $\mathcal{P}$. Assume that e is parallel to $\mathcal{P}$ on the left of $\mathcal{P}$ (the other case is treated similarly). Let $\mathcal{P}_e$ be the part of $\mathcal{P}$ between the origin and the end-vertex of e, and let C_e be the cycle delimited by e and $\mathcal{P}_e$. Then $\mathcal{P}_e$ has length at least 2, otherwise T would have a double edge. Let v be a vertex of $\mathcal{P}_e$ different from the two extremities of e. Then Proposition 1.11 ensures that there exists a simple red path from v to N that starts into the interior of C_e. As the path goes to N, it has to reach C_e at another vertex, in a way that contradicts Condition C1 (the reason is that all vertices on C_e are also on $\mathcal{P}_e$). □

Remark. With the terminology of partial orders, this lemma states that the acyclic graph on blue edges (and the same holds for red edges) has no transitive edges, i.e., it is the Hasse diagram of the partial order induced on the vertices of $T\backslash\{S, N\}$.

LEMMA 2.6. *Let X be a transversal structure of T and let $\mathcal{P}$ be an oriented blue path connecting W to E, with its associated cycle $\mathcal{C}(\mathcal{P})$ defined as the concatenation of $\mathcal{P}$ and of the path $E \to S \to W$. Then, for any inner face f of the blue-map T_b of T having its left lateral path included in $\mathcal{P}$, the right lateral path of f is an admissible path for $\mathcal{C}(\mathcal{P})$, as defined in Section 5.1.*

Proof. Let v and v' be the origin and end-vertex of the right lateral path $\mathcal{P}_r(f)$ of f. Observe that the part of $\mathcal{P}$ between v and v', which we denote by $[v, v']$, is the left lateral path of f. We have to check that the three conditions of an admissible path are satisfied by $\mathcal{P}_r(f)$. The fact that $\mathcal{P}_r(f)$ and $[v, v']$ have at least two edges

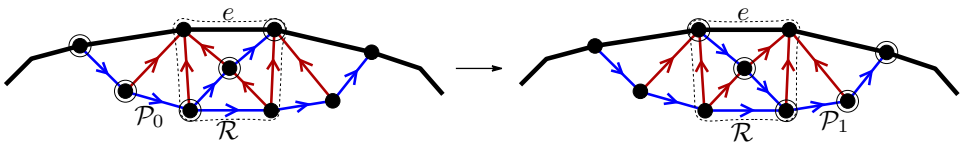


FIGURE 11. The presence of a right alternating 4-cycle in X_0 yields the presence of an admissible path on the right of the chosen path.

(first condition) and the fact that the replacement of $[v, v']$ by $\mathcal{P}_r(f)$ in the path $\mathcal{P}$ creates no internal chord (third condition) are a direct consequence of Lemma 2.5. The second condition is immediately checked by transversality of the two bipolar orientations. □

Remark. This result ensures in particular that any transversal structure X of T can be computed by the generic algorithm, by making a suitable choice of the admissible path at each step. Indeed, any sweeping of the inner faces of the blue map of T (for the transversal structure X) yields a possible execution of the generic algorithm, where the right lateral path of the swept face of T_b is chosen at each step.

Proposition 2.7'. *Given an irreducible triangulation T,* ComputeMinimal*(T) outputs the minimal transversal structure of T.*

Proof. We denote by X_0 the transversal structure computed by ComputeMinimal(T). Assume that X_0 has a right alternating 4-cycle $\mathcal{R}$. Let X_1 be the transversal structure obtained from X_0 by switching the colors of the edges interior to $\mathcal{R}$ and orienting these edges so that Condition C1 is satisfied, see Figure 11. Let e be the (oriented) blue edge of $\mathcal{R}$ having the interior of $\mathcal{R}$ on its right. We consider the step of ComputeMinimal(T) during which e is treated (i.e., colored, oriented and removed from $\mathcal{C}$). Denote by $T_b(X_0)$ the blue-map of T for the transversal structure X_0. Let $\mathcal{P}_0$ be the right lateral path of the face of $T_b(X_0)$ on the right of e. Similarly, consider the blue-map $T_b(X_1)$ of T for X_1; and denote by $\mathcal{P}_1$ the right lateral path of the face of $T_b(X_1)$ on the right of e. Lemma 2.6 ensures that, at the step where e is treated by ComputeMinimal(T), $\mathcal{P}_0$ and $\mathcal{P}_1$ are both admissible paths. As the transversal structure output by ComputeMinimal(T) is X_0, $\mathcal{P}_0$ is the admissible path used to update $\mathcal{C}$ at the step where e is treated. Hence, by definition of ComputeMinimal(T), $\mathcal{P}_0$ is the rightmost admissible path on $\mathcal{C}$. As illustrated in Figure 11, the fact that $\mathcal{R}$ is a right alternating 4-cycle ensures that $\mathcal{P}_1$ is more on the right than $\mathcal{P}_0$, yielding a contradiction. □

5.3. Implementation. We explain here how to implement ComputeMinimal(T) so that the complexity is linear. Let us first give a brief overview. The idea is to maintain a list of *nested* paths incident to the (iteratively shrinked) cycle $\mathcal{C}$, such that a path of the list becomes the rightmost admissible path as soon as all paths which it encloses are treated. The nested structure of the paths indicates that the suitable data structure to store them is a stack.

In all this section, we use the same notations as in Section 5.1. In particular, we denote by $\mathcal{C}$ the cycle iteratively shrinked so as to compute the minimal transversal structure; and for two vertices v and v' of $\mathcal{C}\backslash\{S\}$ with v on the left of v', we denote by $[v, v']$ the path on $\mathcal{C}\backslash\{S\}$ going from v to v'.

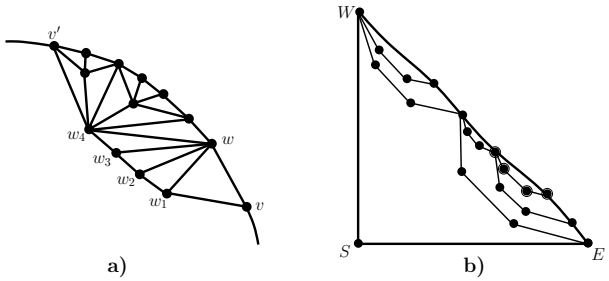


FIGURE 12. The matching path of a vertex v (a). The nested arrangement of admissible matching paths (b). The vertices of the righmost admissible path are surrounded.

We now introduce the terminology of matching path. Let v be a vertex of $\mathcal{C}\backslash\{S\}$ different from W and from the right neighbour of W on $\mathcal{C}\backslash\{S\}$. Denote by w the left neighbour of v on $\mathcal{C}\backslash\{S\}$ and write $w_1, \ldots, w_k$ for the neighbours of w in the interior of $\mathcal{C}$, taken in cw-order. Let $1 \leq i \leq k$ be the smallest index such that w_i is connected by an edge to a vertex of $\mathcal{C}\backslash\{S\}$ on the left of w (such an index exists because w_k is connected to the left neighbour of w). Denote by v' the leftmost vertex of $\mathcal{C}\backslash\{S\}$ connected to w_i by an edge. Then the path $v, w_1, \ldots, w_k, v'$ is called the *matching path* of v, see Figure 12(a). The vertices $w_1, \ldots, w_k$ are called the *internal vertices* of the matching path. The matching path of a vertex v is said to be *admissible* if it does not share any internal vertex with the matching path of a vertex on the right of v on $\mathcal{C}\backslash\{S\}$. As a consequence, admissible matching paths do not share any internal vertex, so that they are *nested*, as illustrated in Figure 12(b). An important remark is that the rightmost admissible path is the matching path of its right extremity. In addition, it is admissible. Precisely, as illustrated in Figure 12(b), the rightmost admissible path is the rightmost one among admissible matching paths that enclose no other admissible matching path.

The implementation of COMPUTEMINIMAL we present here uses a list L of vertices and a stack K of admissible matching paths, so that the following invariants are maintained throughout the execution of the algorithm:

- The list L contains the vertices of $\mathcal{C}\backslash\{S\}$ ordered from right to left, and there is a pointer on a vertex of the list.
- The stack K contains the admissible matching paths whose right extremity is on the right (on $\mathcal{C}\backslash\{S\}$) of the currently marked vertex of L.

As in Section 5.1, the cycle $\mathcal{C}$ is initialized with vertices S, W, E and all interior neighbours of N. In addition, all inner edges incident to N are initially colored red and oriented toward N. The list L is initialized with all neighbours of N ordered from E to W and the pointed vertex of L is E. The stack K is initially empty.

At each step, the pointed vertex v of L is considered. Two cases can arise. If K is not empty and v is the left extremity of the admissible path $\mathcal{P}$ on top of K, then $\mathcal{P}$ is the rightmost admissible path. Then we perform the following operations, where w denotes the right extremity of $\mathcal{P}$: 1) Color blue and orient the edges of $[v, w]$ from v to w. 2) Color red and orient from $\mathcal{P}$ to $[v, w]$ the edges inside the cycle delimited by $\mathcal{P}$ and $[v, w]$. 3) Replace $[v, w]$ by $\mathcal{P}$ in the list L (according to the right-to-left order). 4) Move the pointer of L to the right extremity of $\mathcal{P}$. 5) Pop out the path $\mathcal{P}$ from the stack K.

If K is empty or v is not the left extremity of the admissible path on top of K, we perform the following operations: 1) Compute the matching path of v and mark as visited all internal vertices of the matching path; if, during the computation, an already visited vertex is encountered inside $\mathcal{C}$, stop at once. 2) If no visited vertex has been encountered inside $\mathcal{C}$, add the matching path at the top of K. 3) Move the pointer of L to the vertex following v in the list (i.e., the left neighbour of v on $\mathcal{C}\backslash\{S\}$).

Figure 13 shows the complete execution of the algorithm on the triangulation that was already used as example in Figure 9. It is easily checked that the invariants are maintained and that the algorithm terminates (indeed each step either shrinks $\mathcal{C}$ or moves the marked vertex of L to the left).

Proposition 2.7". *Algorithm* ComputeMinimal*(T) can be implemented to run in linear time.*

Proof. Each blue edge of T is traversed twice: the first time on an admissible matching path, the second time on the cycle $\mathcal{C}$. Each red edge is traversed at most once, as an edge of a matching path whose computation is interrupted. Euler relation ensures that a triangulation with n vertices has $3n-7$ edges, so that the complexity is also linear in the number of vertices. □

6. Appendix: proof of Theorem 2.1

Let G be an outer-triangular 3-connected map. We denote by X_0 the α_3-orientation of the derived map G' computed by the algorithm presented in Section 4. The aim of the whole section is to show Theorem 2.1, i.e., that X_0 is the minimal α_3-orientation of the derived map G'.

Our proof follows the principle of the proof by Brehm [**25**] for triangulations, see Proposition 2.5. Recall that, for a triangulation, the argument is the following (see the proof of Proposition 1.7): the presence of a clockwise circuit implies the presence of an essential clockwise circuit which is, in the case of a triangulation, a 3-circuit (x, y, z). Then the clockwise orientation of (x, y, z) determines unambiguously (up to rotation) the labels of the 3 edges of (x, y, z). These labels determine an order of treatment of the 3 vertices x, y and z not compatible with the fact that the admissible vertex chosen at each step is the rightmost one.

In the general case of 3-connected maps, which we consider here, the proof is more involved but follows the same lines. This time there is a finite set of configurations for an essential circuit of an α_3-orientation (for a triangulation this set is restricted to the triangle). A common caracteristic is that the presence of a clockwise circuit $\mathcal{C}$ for each of these configurations implies the presence of three paths $\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{P}_3$ of edges of G whose concatenation forms a simple cycle in G (in the case of a triangulation, the three paths are reduced to one edge). In addition, the fact that $\mathcal{C}$ is clockwise determines unambiguously the labels and orientations of the edges of $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}_3$. Writing v_1, v_2 and v_3 for the respective origins of these three paths, our proof (as in the case of triangulations, but with quite an amount of technical details) relies on the fact that the labels of $\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{P}_3$ imply an order of processing of v_1, v_2 and v_3 which is not compatible with the fact that the admissible vertex chosen at each step is the rightmost one.

6.1. The algorithm outputs an α_3-orientation. By construction of the orientation, each primal vertex of the derived map G' has one outgoing edge for each label 1, 2 and 3, hence it has outdegree 3. By construction also, each edge-vertex of G' has outdegree 1. Hence, to prove that X_0 is an α_3-orientation, it just remains to prove that each dual vertex of G' has outdegree 3 in X_0.

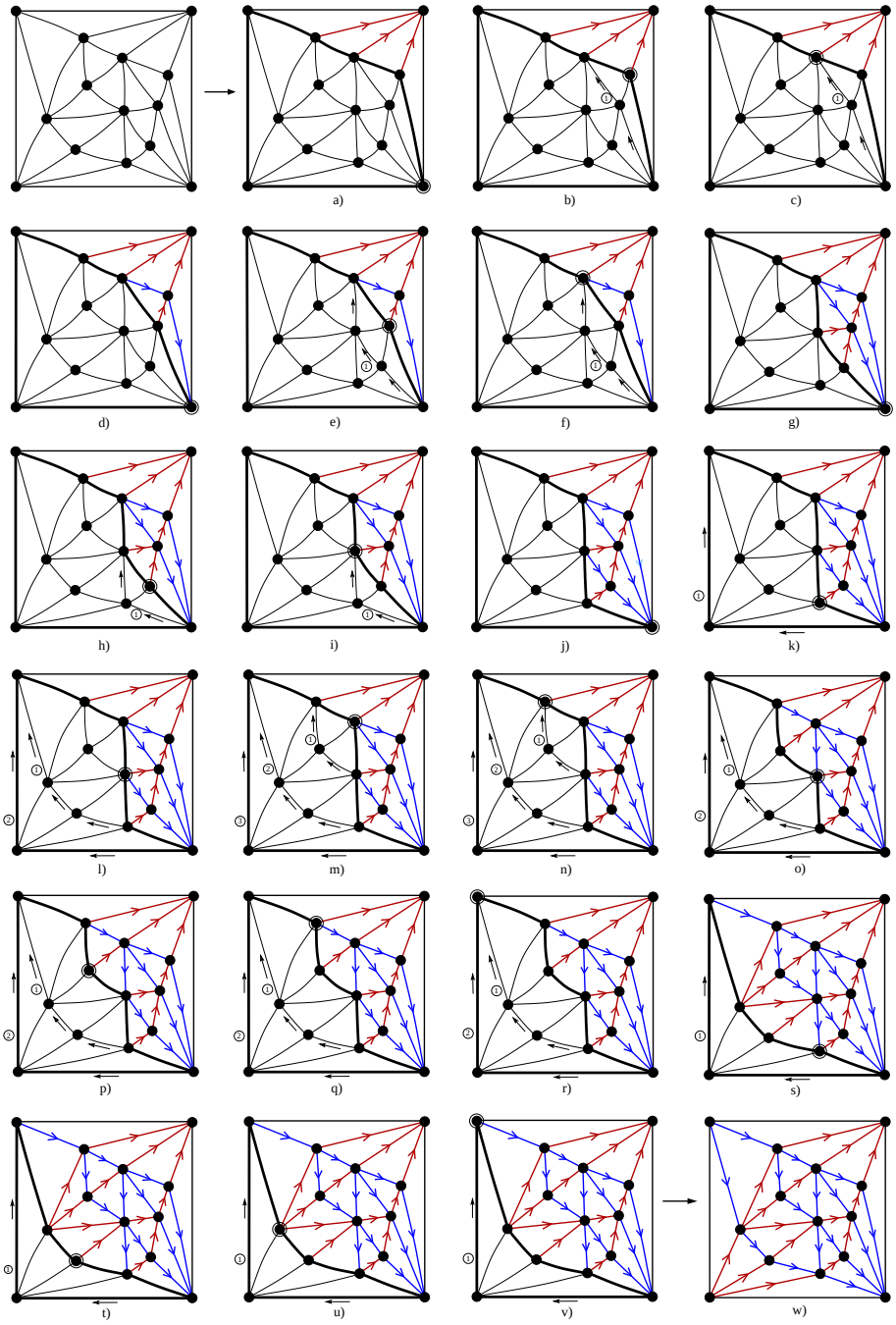


FIGURE 13. The complete execution of the linear time implementation of COMPUTEMINIMAL on an example. At each step, the edges of $\mathcal{C}$ are bold black, the pointed vertex of L is surrounded, and the admissible paths that are in the stack K are bordered by small arrows and have a label indicating their position in the stack.

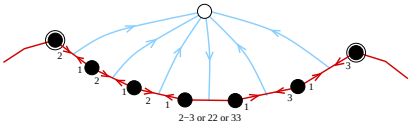


FIGURE 14. The dual vertex of a face f has one outgoing edge connected to the lower path of f.

Let f be an inner face of G and v_f the corresponding dual vertex in G^*. Let k be the step during which f has been merged with the infinite face of G. During this step, a sequence of consecutive edges of f has been removed. This path of removed consecutive edges is called the *upper path* of f. The path of edges of f that are not in the upper path of f is called the *lower path* of f. By construction of the orientation (see Figure 7), exactly two edges of G' connecting v_f to an edge-vertex of the upper path of f are going out of v_f: these are the edge-vertices corresponding to the two extremal edges of the upper path.

Hence it just remains to prove that exactly one edge of G' connecting v_f to an edge-vertex of the lower path of f is going out of v_f. First, observe that the lower path P of f is a non empty path of edges on $\mathcal{C}_{k+1}$, such that the two extremities v_l and v_r of the path are active and all vertices of $]v_l, v_r[$ are passive on $\mathcal{C}_{k+1}$, see Figure 7. The fact that exactly one edge of G' connecting v_f to an edge-vertex of P is going out of v_f is a direct consequence of the following lemma, see Figure 14.

LEMMA 2.7. *At a step k of the algorithm, let v_1 and v_2 be two active vertices on $\mathcal{C}_k$ such that all vertices of $]v_1, v_2[$ are passive. Then the path $[v_1, v_2]$ on $\mathcal{C}_k$ is partitioned into*

- *A (possibly empty) path $[v_1, v]$ whose edges are bi-oriented in the finally computed orientation X_0, the left half-edge having label 2 and the right half-edge label 1.*
- *An edge $e = [v, v']$ that is, in X_0, either simply oriented with label 2 from v to v', or simply oriented with label 3 from v' to v, or bi-oriented, with label 2 on the half-edge incident to v and label 3 on the half-edge incident to v'.*
- *A (possibly empty) path $[v', v_2]$ such that, in X_0, each edge of $[v', v_2]$ is bi-oriented, with label 1 on the left half-edge and label 3 on the right half-edge.*

Proof. The proof is by induction on the length of $[v_1, v_2]$. Assume that this length is equal to 1. Then $[v_1, v_2]$ is reduced to an edge. If v_1 is removed at an earlier step than v_2, then the edge $\{v_1, v_2\}$ is simply oriented with label 2 from v_1 to v_2. If v_2 is removed at an earlier step than v_1, then the edge $\{v_1, v_2\}$ is simply oriented with label 3 from v_2 to v_1. If v_1 and v_2 are removed at the same step, then $\{v_1, v_2\}$ is bi-oriented, with label 2 on v_1's side and label 3 on v_2's side, see Figure 7.

Assume that the length of $[v_1, v_2]$ is at least 2. Observe that the outer path $[v_1, v_2]$ remains unchanged as long as none of v_1 or v_2 is removed. This remark follows from the fact that all vertices of $]v_1, v_2[$ are passive, so that no vertex of $[v_1, v_2]$ can be treated as long as none of v_1 or v_2 is treated.

Then, two cases can arise: if v_1 is removed before v_2, the right neighbour v of v_1 becomes active and the edge $\{v_1, v\}$ is bi-oriented, with label 2 on v_1's side and label 1 on v's side, see Figure 7. Similarly if v_2 is removed before v_1, the left neighour v of v_2 becomes active and the edge $\{v, v_2\}$ is bi-oriented with label 3 on v_2's side and label 1 on v's side.

The result follows by induction, with a recursive call to the path $[v, v_2]$ in the first case and to the path $[v_1, v]$ in the second case.

6.2. The algorithm outputs the minimal α_3-orientation.

6.2.1. *Definitions and preliminary lemmas.* Let us first define what we call *maximal bilabelled paths*. Let v be a vertex of G. For $1 \leq i \leq 3$, the i-path of v is the unique path $P_v^i = (v_0, \ldots, v_m)$ of edges of G starting at v and such that each edge (v_p, v_{p+1}) is the

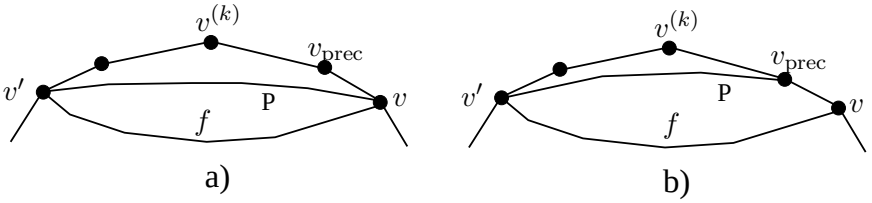


FIGURE 15. The two possible configurations related to the next active vertex on the right of $v^{(k)}$.

outgoing edge of v_p with label i (i.e., the edge of G containing the outgoing half-edge of v_p with label i). Acyclicity properties of Schnyder woods ensure that P_v^i ends at the outer vertex a_i, see [**48**]. For $1 \leq i \leq 3$ and $1 \leq j \leq 3$ with $i \neq j$, we define the *maximal $i-j$ path* starting at v as follows. Let $l \leq m$ be the maximal index such that the subpath $(v_0, \dots, v_l)$ of P_v^i only consists of bi-oriented edges with labels $i-j$. Then the maximal $i-j$ path starting at v is defined to be the path $(v_0, \dots, v_l)$ and is denoted by P_v^{i-j}.

At a step $k \geq 2$, let $v^{(k)}$ be the chosen vertex, i.e., the rightmost admissible vertex on $\mathcal{C}_k$. First, observe that there exists an active vertex on the right of $v^{(k)}$. Indeed, the rightmost vertex a_2 is active as soon as $k \geq 2$. In addition a_2 is non admissible on $\mathcal{C}_k$ because it is blocked, so that a_2 is different from $v^{(k)}$. Hence, a_2 is an active vertex on the right of $v^{(k)}$.

We define the *next active vertex on the right* of $v^{(k)}$ as the unique vertex v on the right of $v^{(k)}$ on $\mathcal{C}_k$ such that all vertices of $]v^{(k)}, v[$ are passive.

LEMMA 2.8. *At a step $k \geq 2$, let $v^{(k)}$ be the chosen vertex. Let v be the next active vertex on the right of $v^{(k)}$. Let v_{prec} be the left neighbour of v on $\mathcal{C}_k$. Then, in the finally computed orientation X_0, each edge of $[v^{(k)}, v_{\text{prec}}]$ is bi-oriented, with label 2 on its left side and label 1 on its right side. The edge $e = \{v_{\text{prec}}, v\}$ is either simply oriented with label 2 from v_{prec} to v or bi-oriented, with label 2 on v_{prec}'s side and label 3 on v's side. In other words, $P_{v^{(k)}}^{2-1} = [v^{(k)}, v_{\text{prec}}]$ and the outgoing edge of v_{prec} with label 2 is (v_{prec}, v).*

Proof. To prove this lemma, using the result of Lemma 2.7, we just have to prove that $\{v_{\text{prec}}, v\}$ is neither bi-oriented with label 1 on v_{prec}'s side and label 3 on v's side, nor simply oriented with label 3 from v to v_{prec}, see Figure 14.

First, as the active vertex v is on the right of $v^{(k)}$, it can not be admissible, so that v is blocked. As a consequence there exists a vertex v' and a face f such that v, v' and f form a separator. Lemma 2.2 ensures that there exists an admissible vertex in $]v', v[$. Hence the vertex v' is on the left of $v^{(k)}$ on $\mathcal{C}_k$, otherwise $v^{(k)}$ would not be the rightmost admissible vertex. Let P be the path of f going from v to v' with the interior of f on its left. Two cases can arise: 1) the first edge of P is different from (v, v_{prec}), so that v_{prec} is above P, see Figure 15(a). Clearly, v remains blocked as long as all vertices above P have not been treated. Hence, v_{prec} will be treated at an earlier step that v. As v is active, it implies (see Figure 7) that (v_{prec}, v) is simply oriented with label 2 from v_{prec} to v. 2) the first edge of P is (v, v_{prec}), see Figure 15(b). Observe that v_{prec} can not be equal to v'. Indeed v is on the right of $v^{(k)}$, so that v_{prec} is on the right or equal to $v^{(k)}$, whereas v' is on the left of $v^{(k)}$. Hence, P has length greater than 1. As a consequence, when f will cease to be separating, v_{prec} will only be incident to f. Figure 7 ensures that, when such a vertex is treated, the edge connecting this vertex to its right neighbour is always bi-oriented and bi-labelled 2-3, which concludes the proof.

LEMMA 2.9. *At a step $k \geq 2$, let $v^{(k)}$ be the rightmost admissible vertex and v the next active vertex on the right of $v^{(k)}$. Let v^{3-2} be the extremity of P_v^{3-2} in X_0 and e the outgoing edge of v^{3-2} with label 3. If e is bi-oriented, it is bi-labelled 3-1 and we define $v_1 = v^{3-2}$. Otherwise, e is simply oriented and we define v_1 as the extremity of e.*

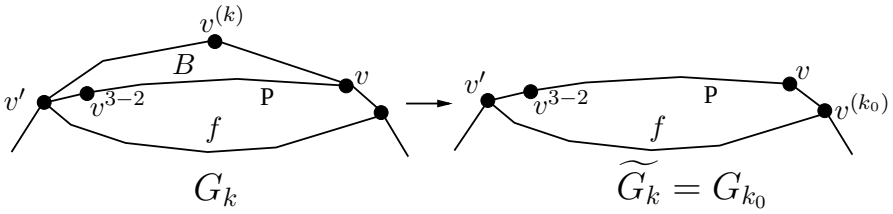


FIGURE 16. The path between v and v^{3-2} will consist of bi-oriented edges bilabelled 3-2.

Then v_1 belongs to $\mathcal{C}_k$ and is on the left of $v^{(k)}$.

Proof. First, observe that each vertex v'' such that the pair $\{v'', v\}$ is separating is on the left of $v^{(k)}$, otherwise, Lemma 2.2 ensures that there exists an admissible vertex in $]v'', v[$, in contradiction with the fact that $v^{(k)}$ is the rightmost admissible vertex.

Observe also that the set of separators (v'', v, f) involving v and endowed with the inclusion-relation on the separated areas is not only a partial order but a total order. In particular, for two separators (v_1'', v, f_1) and (v_2'', v, f_2), if v_1'' is on the left of v_2'', then the separated area of (v_2'', v, f_1) is strictly included in the separated area of (v_1'', v, f_2). In addition, this set is non empty because v is the next active vertex on the right of $v^{(k)}$, hence v is blocked.

Let (v', v, f) be the maximal separator for this totally ordered set. Then the separated area of (v', v, f) contains all separating faces incident to v except f. Let $P(f)$ be the path of edges of f going from v to v' with the interior of f on its left. We denote by B the separated area of (v', v, f). Let $\widetilde{G_k}$ be the submap of G obtained by removing B from G_k. Let $\widetilde{\mathcal{C}_k}$ be the contour of $\widetilde{G_k}$.

We claim that f is not separating in $\widetilde{G_k}$. Otherwise, there would exist a vertex v_2 on the right of v such that (v, v_2, f) is a separator or there would exist a vertex v_3 on the left of v' such that (v_3, v', f) is a separator: the first case is in contradiction with the fact that all separators $\{v, v_2\}$ involving v are such that v is on the right of v_2. The second case is in contradiction with the fact that (v', v, f) is the maximal separator involving v.

We claim that only vertices of B will be removed from step k on, until all vertices of B are removed. Indeed, all separating faces incident to vertices on the right of v are faces of $\widetilde{G_k}$, hence they will remain separating as long as not all vertices of B are removed. As all vertices on the right of v are either blocked or passive, it is easy to see inductively that all these vertices will keep the same status until all vertices of B are removed.

Let k_0 be the first step where all vertices of B have been removed. Then $G_{k_0} = \widetilde{G_k}$. Hence f is not separating anymore on $\mathcal{C}_{k_0}$, but all other faces of $\widetilde{G_k}$ that are separating at step k are still separating at step k_0. We have seen that the separating faces incident to v at step k are the face f and faces in B. In addition, all faces of G_{k_0}, except f, have kept their separating-status between step k and step k_0. Hence v is admissible on $\mathcal{C}_{k_0}$, and the rightmost admissible vertex $v^{(k_0)}$ at step k_0 is a vertex incident to f. It is either v or a vertex of f on the right of v (on $\mathcal{C}_{k_0}$) such that $[v, v^{(k_0)}]$ only consists of edges incident to f (otherwise f would be separating), see Figure 16, where $v^{(k_0)}$ is the right neighbour of v.

Moreover, the left-connection vertex of $v^{(k_0)}$ is v'. Otherwise there would be a vertex of f on $\widetilde{\mathcal{C}_k}$ and on the left of v'. This vertex would also be on $\mathcal{C}_k$ (because only vertices of B are removed to obtain $\widetilde{G_k}$ from G_k), in contradiction with the fact that (v', v, f) is the maximal separator of $\mathcal{C}_k$ involving v.

Then two cases can arise whether v' is passive or active on $\mathcal{C}_{k_0}$: 1) v' is passive on $\mathcal{C}_{k_0}$. Then v' is not incident to any edge of $G \backslash G_{k_0}$. In particular v' is not incident to any edge of $B \backslash G_{k_0}$. Hence the right neighbour of v' on $\mathcal{C}_{k_0}$ and on $\mathcal{C}_k$ are the same vertex, that

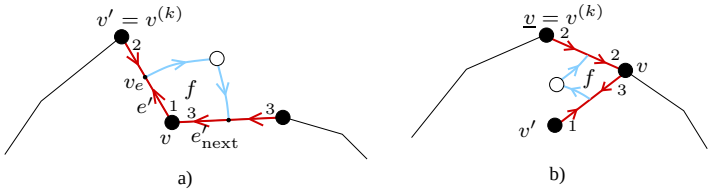


FIGURE 17. Configuration of a face f of G' whose contour is a clockwise circuit and such that the outgoing edge of the unique primal vertex of f has label 1 (Fig. a) and label 3 (Fig. b).

is, the vertex v_1 preceding v' on P. Observe that v_1 is on the left of $v^{(k)}$ on $\mathcal{C}_k$ (indeed, v_1 can not be equal to $v^{(k)}$ at step k because v_1 is incident to f, which is separating at this step). Then, by definition of v_1 and by construction of the orientation (see Figure 7), $P^{3-2}_{v^{(k_0)}}$ is equal to $[v_1, v^{(k_0)}]$ taken from right to left, and (v_1, v') is bi-oriented bi-labelled $3-1$ from v_1 to v'. As $v \in [v_1, v^{(k_0)}]$ on $\mathcal{C}_{k_0}$, $[v, v^{(k_0)}] \subset [v_1, v^{(k_0)}]$, so that P^{3-2}_{v} is equal to $[v, v^{(k_0)}]$ taken from right to left. As $\{v_1, v'\}$ is bi-oriented bi-labelled $3-1$ from v_1 to v', this concludes the proof for the first case (i.e., $v_1 = v^{3-2}$).
2) v' is active on $\mathcal{C}_{k_0}$. In this case, upon taking v_1 to be the vertex v', a similar argument as for the previous paragraph applies: indeed v_1 is a vertex on $\mathcal{C}_k$ on the left of $v^{(k)}$, and P^{3-2}_{v} is the path on $\mathcal{C}_{k_0}$ going from v to the right neighbour of v_1 on $\mathcal{C}_{k_0}$, and the edge connecting the right neighbour of v_1 to v_1 is simply oriented with label 3 toward v_1 (see Figure 7).

LEMMA 2.10. *The vertices a_1, a_2 and a_3 can not belong to any clockwise circuit.*

Proof. Let us consider a_1 (the cases of a_2 and a_3 can be dealt with identically). The outgoing edge of a_1 with label 1 is directed toward the infinite face. The outgoing edges of a_1 with labels 2 and 3 connect respectively a_1 to two edge-vertices whose unique outgoing edge is directed toward the infinite face. Hence each directed path starting at a_1 finishes immediately in the infinite face.

6.2.2. *Possible configurations for a minimal clockwise circuit of X_0.*

LEMMA 2.11. *Let f be an inner face of G'. Then the contour of f is not a clockwise circuit in X_0.*

Proof. Assume that the contour of f is a clockwise circuit. We recall that the contour of f has two edge-vertices, one dual vertex, and one primal vertex v. Let i be the label of the edge e' of f going out of v. The edge e' is the first half-edge of an edge e of G. We denote by v_e the edge-vertex of G' associated to e and by v' the vertex of G such that $e = \{v, v'\}$. As the contour of f is a clockwise circuit, the unique outgoing edge of v_e follows the edge $\{v_e, v\}$ in counter-clockwise order around v_e. Hence, according to Figure 6(c), the edge e is bi-oriented and the second half-edge of e has label $i+1$. We denote by e_{next} the edge of G following e in clockwise order around v. The edge e'_{next} of G' following e' in clockwise order around v is the edge of f directed toward v. Hence, the rules of labelling (Figure 6(a)) ensure that e'_{next} has label $i-1$. As e'_{next} is the second half-edge of e_{next}, this ensures that e_{next} is simply oriented with label $i-1$ toward v.

We now deal separately with the three possible cases $i = 1, 2, 3$:

— Case $i = 1$: The edge e is bi-labelled 1-2 from v to v' and e_{next} is simply oriented with label 3 toward v, see Figure 17(a). Let k be the step of the algorithm during which the vertex v' is removed from G_k. Figure 7 ensures that, if v' is not equal to the rightmost admissible vertex $v^{(k)}$, then the outgoing edge with label 2 of v' is bi-oriented, with label 3 on the other half-edge, which is not the case here. Hence $v' = v^{(k)}$. In addition, as $\{v', v\}$ is bi-labelled 2-1 from v' to v, the vertex v is passive on $\mathcal{C}_k$. Hence, writing $e_{v\rightarrow}$

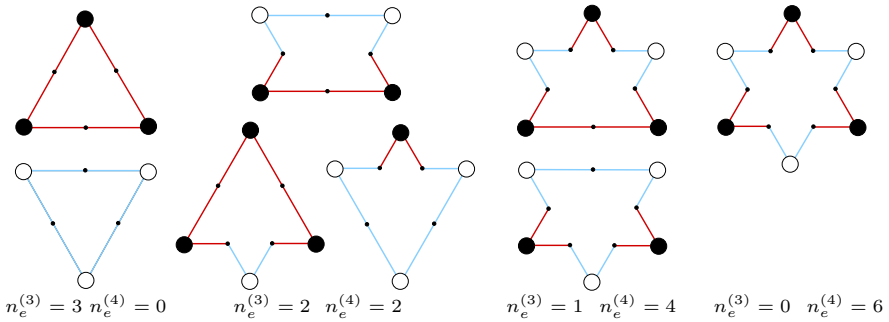


FIGURE 18. The possible configurations for a minimal clockwise circuit of X_0.

for the edge of $\mathcal{C}_k$ whose left extremity is v, there is no edge of $G \backslash G_k$ between e and $e_{v\rightarrow}$ in clockwise order around v, so that $e_{v\rightarrow} = e_{\text{next}}$.

We claim that $k \geq 2$. Otherwise v' would be equal to a_1. As $e = \{v, v'\}$ is bi-labelled 1-2 from v to v', v would be equal to a_2. But according to Lemma 2.10, a_2 can not belong to any clockwise circuit.

Hence $k \geq 2$ and we can use Lemma 2.8. In particular, this lemma ensures that $e_{v\rightarrow}$ is the outgoing edge of v with label 2. We obtain here a contradiction with the fact that e_{next} is going toward v with label 3 and that $e_{v\rightarrow} = e_{\text{next}}$.

— Case $i = 2$: The edge e is bi-labelled 2-3 from v to v' and e_{next} is simply oriented with label 1 toward v. Let k be the step during which v is removed from G_k. By construction of the orientation (see Figure 7), at step k the vertex v belongs to $]\text{left}(v^{(k)}), v^{(k)}[$ and e_{next} is the outgoing edge of v with label 3. This is in contradiction with the fact that e_{next} is simply oriented toward v with label 1.

— Case $i = 3$: The edge e is bi-labelled 3-1 from v to v' and e_{next} is simply oriented with label 2 toward v, see Figure 17(b). Let $\underline{v}$ be the origin of e_{next} and let k be the step during which $\underline{v}$ is removed from G_k. As e_{next} is simply oriented with label 2 from $\underline{v}$ to v, we have $\underline{v} = v^{(k)}$ and $v = \text{right}(v^{(k)})$. Lemma 2.8 ensures that v is the next active vertex on the right of $v^{(k)}$ on $\mathcal{C}_k$. In addition, $k \geq 2$, otherwise $v^{(k)} = a_1$, in contradiction with the fact that the outgoing edge of a_1 with label 2 is bi-oriented. Hence, we can use Lemma 2.9: in our case, the next active vertex on the right of $v^{(k)}$ is v and the path P_v^{3-2} is empty because the outgoing edge with label 3 of v is bi-labelled 3-1. Hence the vertex denoted by v_1 in the statement of Lemma 2.9 is here v. Lemma 2.9 ensures that v is a vertex of $\mathcal{C}_k$ on the left of $v^{(k)}$, in contradiction with the fact that v is the right neighbour of $v^{(k)}$ on $\mathcal{C}_k$. □

LEMMA 2.12 (Felsner [**48**]). *The possible configurations of an essential circuit of X_0 are illustrated in Figure 18.*

Proof. Felsner [**48**, Lem.17] shows that an essential circuit $\mathcal{C}$ of an α_3-orientation has no inner edge whose origin is on $\mathcal{C}$. In addition, if $\mathcal{C}$ is not the contour of a face, he shows that all edge-vertices have either one incident edge or two incident edges inside $\mathcal{C}$, which implies that the length of $\mathcal{C}$ is 6, 8, 10, or 12. The only possible configurations are those listed in Figure 18. As X_0 has no clockwise circuit of length 4 according to Lemma 2.11, this concludes the proof. □

6.2.3. *No configuration of Figure 18 can be a clockwise circuit of X_0.* We have restricted the number of possible configurations for a clockwise circuit of X_0 to the list represented in Figure 18. In this section, we give a method which ensures that the presence of a clockwise circuit for each configuration of Figure 18 yields a contradiction. The method is based on the use of Lemma 2.8, of Lemma 2.9, and of the following lemma:

LEMMA 2.13. *At a step k, let v and v' be two vertices on $\mathcal{C}_k$ such that v is on the left of v'. Assume that there exists a path $P = (v_0, \dots, v_l)$ of edges of G such that $v_0 = v$, $v_l = v'$, and for each $0 \leq i \leq l-1$, the edge (v_i, v_{i+1}) is the outgoing edge of v_i with label 1 in X_0. Then $P = [v, v']$ on $\mathcal{C}_k$ and all edges of P are bi-oriented bilabelled 1-3.*

Proof. Proving that $P = [v, v']$ comes down to proving that all edges of P are on $\mathcal{C}_k$. By construction of the orientation (see Figure 7), for each vertex w of G, the extremity $w_1 \in G$ of the outgoing edge of w with label 1 is removed at an earlier step than w. Moreover, a vertex in $G \backslash G_k$ is removed at a step $j < k$. Hence, if w is in $G \backslash G_k$, then w_1 is also in $G \backslash G_k$. Hence, if P passes by a vertex outside of G_k, it can not reach $\mathcal{C}_k$ again. By definition of an active vertex of $\mathcal{C}_k$, the extremity of its outgoing edge with label 1 is a vertex of $G \backslash G_k$. Hence none of the vertices $v_0, \dots v_{l-1}$ can be active, otherwise P would pass by a vertex outside of G_k and could not reach $\mathcal{C}_k$ again, in contradiction with the fact that P ends at a vertex of $\mathcal{C}_k$.

Hence, all vertices of $\mathcal{C}_k$ encountered by P before reaching v' are passive. It just remains to prove that the outgoing edge with label 1 of each passive vertex of $\mathcal{C}_k$ is an edge of $\mathcal{C}_k$ and will be bi-oriented and bilabelled 1-3 in X_0.

Let w be a passive vertex of $\mathcal{C}_k$ and let w_l and w_r be respectively the left and the right neighbour of w on $\mathcal{C}_k$. We claim that the outgoing edge of w with label 1 is the edge (w, w_l) if w_l will be removed before w_r and is the edge (w, w_r) if w_r will be removed before w_l. Indeed, as long as none of w_l or w_r is removed, w remains passive and keeps w_l and w_r as left and right neighbour. Let k_0 be the first step where w_l or w_r is removed. By construction of the orientation, two vertices v_1 and v_2 on the contour of $\mathcal{C}_{k_0}$ such that $]v_1, v_2[$ contains a passive vertex can not be removed at the same step. Hence, at step k_0, either w_l or w_r is removed. Assume that the removed vertex at step k_0 is w_l. Then, at step k_0, (w, w_l) is given a bi-orientation and receives label 1 on w's side and label 2 on w_l's side, see Figure 7. Similarly, if the removed vertex is w_r then, at step k_0, (w, w_r) is bi-orientated and receives label 1 on w's side and label 3 on w_r's side.

Finally, it is easy to see that only this second case can happen in the path P, because the starting vertex of P is on the left of the end vertex of P on $\mathcal{C}_k$. □

LEMMA 2.14. *None of the configurations of Figure 18 can be the contour of a clockwise circuit in X_0.*

Proof. We take here the example of the third configuration of the case $\{n_e^{(3)} = 2, n_e^{(4)} = 2\}$ of Figure 18 and show why this configuration can not be a clockwise circuit in X_0. Let $\mathcal{C}$ be a clockwise circuit corresponding to such a configuration. Then $\mathcal{C}$ contains two successive dual edges e_1^* and e_2^* —in counter-clockwise order around $\mathcal{C}$— and a unique primal vertex which we denote by $v_{\mathcal{C}}$. Let M' be the submap of G' obtained by removing all edges and vertices outside of $\mathcal{C}$. Let M be the submap of G obtained by keeping only the edges whose associated edge-vertex belongs to M' and by keeping the vertices incident to these edges. As $\mathcal{C}$ is an essential circuit, no edge inside $\mathcal{C}$ has its origin on $\mathcal{C}$, see [**48**, Lem.17]. The rules of labelling (see Figure 6), the fact that all edge-vertices have outdegree 1, and the fact that no edge goes from a vertex of $\mathcal{C}$ toward the interior of $\mathcal{C}$ determine unambiguously the labels and orientations of all the edges of the contour of M in X_0, up to the label of the outgoing edge of $v_{\mathcal{C}}$ on $\mathcal{C}$. Figures 19(a), 19(b) and 19(c) represent the respective configurations when the label of the outgoing edge of $v_{\mathcal{C}}$ on $\mathcal{C}$ is 1, 2 or 3.

First, we deal with the case of Figure 19(a). Let $\hat{v}$ (resp. $\widehat{v_0}$) be the primal vertex outside of $\mathcal{C}$ and adjacent to the edge-vertex associated to e_2^* (resp. e_1^*). Let $\widehat{v'}$ be the primal vertex inside of $\mathcal{C}$ and adjacent to the edge-vertex associated to e_2^*. Let k be the step at which $\hat{v}$ is removed from G. As already explained in preceding proofs (for example in the proof of Lemma 2.11), it is easy to see that $k \geq 2$ and that $\hat{v}$ is the chosen vertex $v^{(k)}$. Hence we can use Lemma 2.8 and Lemma 2.9. Lemma 2.8 and the configuration of Figure 19(a) ensure that $\widehat{v'}$ is the right neighbour of $\hat{v}$ on $\mathcal{C}_k$ and that $\widehat{v_0}$ is the next active vertex on the right of $\hat{v}$ on $\mathcal{C}_k$. Moreover, the configuration of Figure 19(a) ensures that $\widehat{v_1}$ corresponds to the vertex v_1 in the statement of Lemma 2.9. Hence Lemma 2.9 ensures

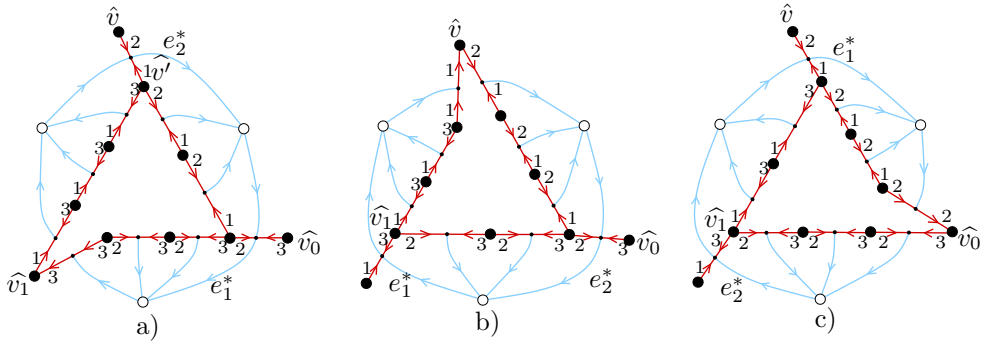


FIGURE 19. The 3 possible cases for the contour of the map M associated to the third configuration of the case $\{n_e^{(3)} = 2, n_e^{(4)} = 2\}$ in Figure 18.

that $\widehat{v_1}$ is on $\mathcal{C}_k$ on the left of $\hat{v}$. We see on Figure 19(a) that there is an oriented path P going from $\widehat{v_1}$ to $\hat{v}$ such that each edge of the path is leaving with label 1. Lemma 2.13 ensures that all edges of P are bilabelled 1-3, in contradiction with the fact that $(\widehat{v'}, v)$ is bilabelled 1-2.

We deal with the case of Figure 19(b) similarly. We define $\hat{v} := v_{\mathcal{C}}$ and denote by $\widehat{v_0}$ the primal vertex outside of $\mathcal{C}$ and adjacent to the edge-vertex associated to e_2^*. We denote by $\widehat{v_1}$ the primal vertex inside of $\mathcal{C}$ and adjacent to the edge-vertex associated to e_1^*. Let k be the step where $\hat{v}$ is removed. Then it is easy to see that $k \geq 2$ and $\hat{v} = v^{(k)}$. Hence we can use Lemma 2.8 and Lemma 2.9. Lemma 2.8 and the configuration of Figure 19(b) ensure that $\widehat{v_0}$ is the next active vertex on the right of $\hat{v}$ on $\mathcal{C}_k$. Then we see on Figure 19(b) that the vertex $\widehat{v_1}$ corresponds to the vertex v_1 of the statement of Lemma 2.9. Hence, Lemma 2.9 ensures that $\widehat{v_1}$ is on $\mathcal{C}_k$ on the left of $\hat{v}$. We see on Figure 19(b) that there exists an oriented path P going from $\widehat{v_1}$ to $\hat{v}$ such that each edge of P leaves with label 1. But we see that the last edge of P is simply oriented, in contradiction with Lemma 2.13.

Finally, the case of Figure 19(c) can be treated similarly (the vertices $\widehat{v_0}$, $\widehat{v_1}$ and $\hat{v}$ are indicated on the picture).

A similar treatment ensures that none of the other configurations of Figure 18 can be the contour of a clockwise circuit of X_0. □

Finally, Theorem 2.1 follows from Lemma 2.14 and from the fact that all possible configurations for a clockwise circuit of X_0 are listed in Figure 18.

Conclusion. For each of the combinatorial structures {Eulerian orientations, Bipolar orientations, Schnyder woods, Transversal structures}, there exists a linear time algorithm computing the minimal structure of a given map (the structure at the bottom of the distributive lattice).

These algorithms (except for Eulerian orientations) are similar. At each step, a vertex is chosen on a front line, and its incident edges are directed, making the area oriented larger. The systematic choice of the rightmost admissible vertex on the front line yields the minimal structure.

Notice that no generic linear algorithm is known to compute the minimal α-orientation of a planar map, given a feasible function α as input. It seems that the principles we use for our specific combinatorial structures do not generalise easily.

CHAPTER 3

Bijective counting of maps

Introduction. Many families of (rooted) planar maps have strikingly simple enumeration formulas. For instance, the number of rooted triangulations with $n+2$ vertices satisfies

$$c_n = \frac{2(4n-3)!}{n!(3n-1)!}. \tag{4}$$

The pioneer method to obtain such formulas has been developed by Tutte and goes back to the 60's [**113**]. It consists of two steps: 1) the maps are decomposed at the root, yielding an equation satisfied by the associated generating functions; 2) the equation is solved using clever algebraic manipulations (a unified method for solving such equations is described in [**21**]). Nevertheless, the simplicity of the formulas such as (4) asks for a direct combinatorial interpretation. A nice bijective method has been introduced by G. Schaeffer in his PhD thesis [**99**]. In each case, a family of rooted maps is in bijection with a family of rooted trees. The principle consists in performing local operations on the tree, so as to *close* faces progressively and obtain a rooted map of the corresponding family. The method has been applied successfully to count bijectively several families of planar maps [**64, 93, 93, 95, 98**]. It appears in the bijection with rooted triangulations [**95**] that the concept of *minimal* α-*orientation* is crucial to recover the tree from the map. As shown in this chapter, the principle can be generalized, making it possible to recover and reformulate several bijections in a unified framework.

Results obtained in this chapter. In Section 1, we present a general method to count bijectively a family of rooted maps using the concept of minimal α-orientation. In this way, we unify four bijections: with the families of (rooted) Eulerian maps, triangulated maps, triangulations, and quadrangulations. The first description of these bijections has been given in [**56, 93, 95, 98**], respectively.

The bijective method presented in Section 1 seems difficult to apply to the families of 3-connected maps and irreducible triangulations. However, by using some orientations derived from minimal α-orientations, we find another type of bijections to enumerate these two families. Even if the principle is also based on local closure operations on a tree, these bijections are truly of another type, as they work with *unrooted* objects. In particular, if a tree has a rotation symmetry, the associated map inherits the symmetry. The advantage is that these bijections yield enumeration formulas for unrooted maps (as well as for rooted maps) of the considered family. In this way, we get the first bijective proof of the nice formula giving the number of unrooted plane triangulations with n vertices (Proposition 3.5), found by Brown [**27**].

Motivations. This chapter provides bijective proofs of enumerative formulas on planar maps already known from the earlier works of Tutte [**111**] and his collaborators [**90**]. As mentioned above these formulas are surprisingly simple. The advantage of our bijective proofs is to give a direct interpretation of the formulas, where binomial coefficients systematically appear. In addition, our bijections reveal some deep combinatorial properties of planar maps, in particular the crucial role played by minimal α-orientations combined with spanning tree decompositions. Finally, the advantage of bijective proofs is that they often give rise to a very efficient algorithmics on the family of objects considered. These applications will be described in the next chapter.

1. Bijections using root-accessible α-orientations

In this section, we present a general method to associate a family of rooted trees to a family of rooted maps in a bijective way. The method works for families of maps characterized by α-orientations satisfying an *accessibility* property. For such a family $\mathcal{M}$, the tree associated to a map $M \in \mathcal{M}$ is an enriched spanning tree of M computed using the minimal α-orientation of M.

Our presentation defines root-accessible α-orientations, describes the bijective method, and then applies it to unify four bijections under one roof, related to rooted Eulerian maps, quadrangulations, triangulated maps, and triangulations, respectively.

1.1. The bijective method. Root-accessible α-orientations. Consider a rooted map $M = (V, E)$ and a feasible function $\alpha : V \to \mathbb{N}$. Let r be the origin of the root of M, called the *root-vertex* of M. An α-orientation of M is said to be *root-accessible* if, for any vertex v, there exists an oriented path from v to r. In this case, any other α-orientation is also root-accessible, as the accessiblity is an invariant of the function α. Hence, the notion of root-accessibility only depends on the function α; accordingly a feasible function α whose α-orientations are root-accessible is also called *root-accessible*. Root-accessible orientations have been defined and considered by Olivier Bernardi to derive a general bijective method for counting tree-rooted maps [**7**] (i.e., maps endowed with a spanning tree).

Opening edge-bipartition of a map. In the sequel, each family $\mathcal{M}$ of rooted maps we consider is such that a specific function α is feasible and root-accessible for each map of $\mathcal{M}$. Examples of such families are Eulerian maps endowed with Eulerian orientations, triangulations endowed with 3-orientations, quadrangulations endowed with 2-orientations (root-accessibility is obtained by orienting the outer face ccw), as we have seen in Chapter 1. Given a map $M = (V, E) \in \mathcal{M}$ endowed with its minimal α-orientation, we define an *opening edge-partition* of M as a partition $(H, \overline{H})$ of the edge-set E such that the following conditions are satisfied.

(1) The edges of H form a spanning tree of M oriented to the root-vertex r.
(2) For each edge $e \in \overline{H}$, let C_e be the unique cycle formed by e and edges of H. Then e has the interior of C_e on its left.

Notice that the root edge is in $\overline{H}$. The following result is an easy consequence of a bijection of Bernardi between root-accessible orientations and spanning trees of a planar map [**7**]. Whereas the purpose of [**7**] is to count maps endowed with a spanning tree (called tree-rooted maps), we will make use of it to count rooted maps.

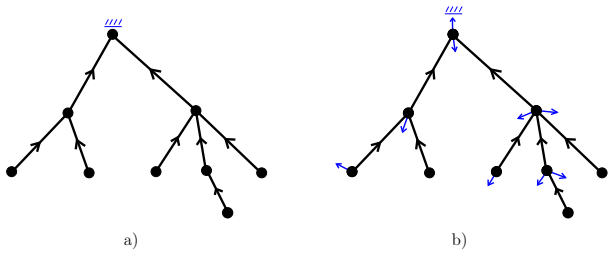


FIGURE 1. A rooted plane tree (Fig.a) and a rooted blossoming tree (Fig.b).

THEOREM 3.1 (Bernardi [**7**]). *A rooted map $M = (V, E)$ endowed with its minimal root-accessible α-orientation admits a unique opening edge-partition $(H, \overline{H})$. The spanning tree formed by edges of H is called the* opening spanning tree *of M.*

This theorem is a powerful tool to count maps. Indeed, it ensures that enumerating the family $\mathcal{M}$ is equivalent to counting opening edge-partitions of maps in $\mathcal{M}$. The idea is that much information on the opening edge-partition is contained in the opening spanning tree, so that we get the flavour of a tree-counting problem. In all examples treated later, an opening edge-partition is encoded by a so-called *blossoming tree*, which are obtained from the opening spanning tree by addition of half-edges.

Definition. A *blossoming tree* is a rooted plane tree T —the edges of T being oriented toward the root— which carries additional half-edges called stems, see Figure 1. A stem is always oriented outward of its incident vertex and end-vertices of stems are not considered as nodes of the tree (such trees have been introduced by Gilles Schaeffer in his PhD [**99**]). In other words, a blossoming tree is a rooted plane tree with some arrows going out of nodes.

In each of the next four sections, we use Theorem 3.1 to recover four bijections (one in each section) between families of blossoming trees and the four families of rooted tetravalent maps, quadrangulations, triangulated maps, and triangulations. For each family $\mathcal{M}$ of rooted maps endowed with a specific root-accessible function α, the main difficulty of our bijective method is to encode the opening edge-partition in a simple way. In each case treated here, the edge-partition is encoded by a blossoming tree. Then we use the outdegree information (i.e., the function α) to characterise and count the trees.

Let us start with the example of tetravalent maps, i.e., the family of maps whose vertices have degree 4. Our presentation is a little sketchy, as we only reformulate bijections that are already known.

1.2. Bijection with rooted tetravalent maps. The family of α-orientations characterising tetravalent maps are Eulerian orientations, i.e., each vertex has indegree 2 and outdegree 2. Notice that these α-orientations are root-accessible because the existence of an Eulerian circuit ensures strong connectivity.

Given a rooted tetravalent map M, let $(H, \overline{H})$ be the opening edge-partition of M, with H the opening spanning tree of M. As H is a spanning tree of M oriented to the root, each vertex different from the root node has outdegree 1 in H and the root node r has outdegree 0 in H. As all vertices have outdegree 2, each vertex $v \neq r$ has outdegree 1 in $\overline{H}$, and r has outdegree 2 in $\overline{H}$. We now

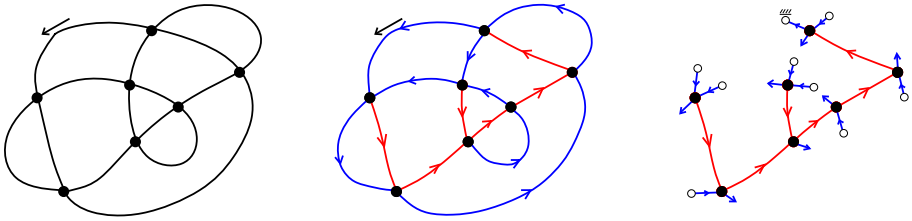


FIGURE 2. Opening of a rooted tetravalent map.

perform the following surgery operations. We cut each non-root edge $e = (v, v')$ of $\overline{H}$ at the middle. The resulting half-edge incident to v is considered as a stem going out of v, and the ingoing half-edge incident to v' is considered as a true edge, precisely a pending edge going out of a new leaf. The root-edge is also cut at its middle, with the difference that the two resulting half-edges are considered as pending edges incident to new leaves. In this way, we obtain a blossoming tree, precisely a tree rooted at a leaf and such that each vertex has degree 3 and carries one stem. Indeed, each vertex has its two ingoing edges leading to its two sons, one outgoing edge leading to its father, and one outgoing edge turned into a stem. Such a tree is called a *blossoming binary tree*. Another way of constructing a blossoming binary tree is to start from a rooted binary tree and place an outgoing stem in one of the 3 angles of each node. Hence the number of blossoming binary trees with n nodes is 3^n multiplied by the number of rooted binary trees with n nodes, which is well known to be $(2n)!/(n!(n+1)!)$ the nth Catalan number.

$$3^n C_n, \text{ where } C_n := \frac{(2n)!}{n!(n+1)!} \text{ is the } n\text{th Catalan number.} \tag{5}$$

Conversely, starting from a blossoming binary tree B, we construct an opening edge-partition of a tetravalent map using the following closure-operations.

Closure of a blossoming binary tree. Traverse the contour of B in ccw order (imagine an ant walking around the tree with the outer face on its right), and associate a cyclic parenthesis word to B, writing an opening parenthesis when a stem is crossed and a closing parenthesis when a leaf is traversed. Clearly a blossoming n-node binary tree has $n+2$ leaves (including the root-leaf) and has n stems. Hence the word has two more)'s than ('s. It is an easy exercise to show that there exists a unique pair of closing parentheses $)_1$ and $)_2$ such that the word between $)_1$ and $)_2$ and the word between $)_2$ and $)_1$ are parenthesis words. For each matching ($\leftrightarrow$), merge the stem corresponding to (and the 1-leg corresponding to) into an edge oriented like the original stem. Finally, let l_1 be the pending edge corresponding to $)_1$ and let l_2 be the pending edge corresponding to $)_2$. If B is rooted at l_1 or l_2, then the tree is called *balanced*. If l_1 is the root, match l_1 with l_2 so as to form an edge e oriented from l_1 to l_2. The edge e is taken as root-edge. The same operation is done if l_2 is the root, upon exchanging the role of l_1 and l_2.

LEMMA 3.1. *The map obtained by performing the closure of a balanced blossoming binary tree with n nodes is a rooted n-vertex tetravalent map endowed with an opening edge-partition $(H, \overline{H})$, where H is the set of original inner edges of the tree (i.e., stems and pending edges are not counted).*

Proof. The outdegree condition of blossoming binary trees ensures that the rooted map $M = \text{Closure}(T)$ is tetravalent and is endowed with an Eulerian orientation. Moreover, the obtained orientation is minimal (proof: there is no clockwise circuit in the beginning, and each closure-edge has the outer face on its right when it is closed, so that no cw circuit can result from a local closure). Finally, as each closure-edge has the outer face on its right when it is closed, the partition $(H, \overline{H})$ defined in the statement of the lemma is an opening edge-partition. □

Clearly, the opening procedure (from a blossoming binary tree to a rooted tetravalent map endowed with an opening edge-partition) and the closure procedure are mutually inverse. This results from the fact that inner edges of a blossoming binary tree correspond to the edge-set H in the opening edge-partition $(H, \overline{H})$ of the associated rooted tetravalent map. As each rooted tetravalent map admits a unique opening edge-partition, we obtain the following result.

PROPOSITION 3.1 (Recover bijection of [**98**]). *There is a bijection between rooted tetravalent maps with n vertices and balanced blossoming binary trees with n nodes. As a consequence, the number t_n of rooted tetravalent maps with n vertices satisfies*

$$t_n = 2 \cdot 3^n \frac{(2n)!}{n!(n+2)!}. \tag{6}$$

Proof. The bijection being already justified, it remains to count balanced blossoming binary trees. We have seen in the description of the closure that, among the $n+2$ leaves of a n-node blossoming binary tree, exactly two of them lead to a (rooted) balanced blossoming binary trees. Accordingly, the proportion of balanced blossoming binary trees is $2/(n+2)$. The result follows from Formula (5). □

Remark. Our formulation of the bijection differs from the original article [**98**] by the use of minimal orientations to calculate the tree from the map. In the original article, the tree is recovered by traversing the outer face and breaking non-disconnecting edges until the remaining figure is a tree. Equivalently, as explained in [**94**], perform a left-to-right breadth-first traversal on the dual map. The formulation of the bijection using minimal orientations is already given in my Masters thesis [**56**], however without using the concept of opening edge-partition.

The same principle can be extended to count any family of rooted Eulerian maps with prescribed distribution of vertex degrees, recovering the general bijective results obtained in [**98**].

1.3. Bijection with rooted quadrangulations. To describe the bijection with rooted quadrangulations, it proves more convenient to work with rooted quadrangulations such that the outer vertex following the root vertex in clockwise order around the outer face has degree 2, see Figure 3(b). Such rooted quadrangulations are called *complete*. This is a minor modification, because rooted quadrangulations with $(n+2)$ vertices are in bijection with rooted complete quadrangulations with $(n+3)$ vertices. Indeed, a rooted quadrangulation is made complete by adding an outer path of length 2 connecting the root vertex to its opposite outer vertex, see Figure 3(a)-(b). Recall that the α-orientations associated with the family of quadrangulations are the 2-orientations, i.e., orientations of the inner edges such that each inner vertex has outdegree 2 and the outer vertices have outdegree 0. Given a rooted complete quadrangulation endowed with its minimal 2-orientation, we first

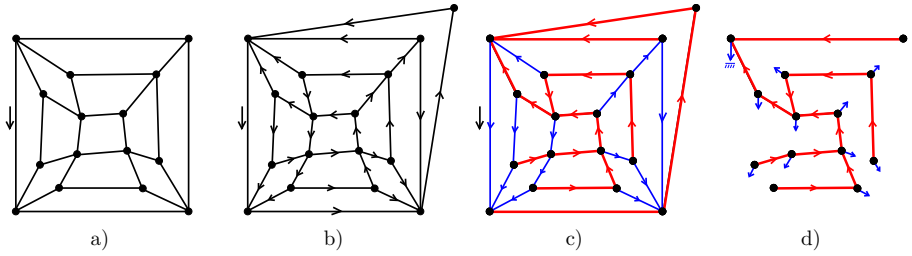


FIGURE 3. Opening of a rooted complete quadrangulation.

make the orientation root-accessible by orienting the outer edges so as to form a counter-clockwise circuit. As we have seen in Section 4, the contour of the outer face is accessible from any inner vertex of a quadrangulation endowed with a 2-orientation; hence, orienting the outer face counterclockwise makes the orientation root-accessible, while still having no clockwise circuit.

Similarly as for tetravalent maps, the bijection to count quadrangulations consists in encoding an opening edge-partition by a blossoming tree. This procedure is called the *opening mapping*.

Opening mapping. Let Q be a rooted complete quadrangulation with $n+3$ vertices and endowed with its minimal 2-orientation. Orient the outer face of Q counter-clockwise and consider the opening edge-partition $(H, \overline{H})$ of Q. Then perform the following operations, see Figure 3(c)-(d).

(1) For each edge e of $\overline{H}$, delete the ingoing half-edge of e, and specify the outgoing half-edge of e as a stem.
(2) Delete the non-root outer edges of Q as well as the non-root outer vertices, see Figure 3(d).

Moreover, specify the root stem as the outgoing half-edge of the root.

Definition. A *rooted 1-stem plane tree* is a plane tree such that each node carries one stem, the tree being rooted at a stem.

LEMMA 3.2. *The opening of a rooted complete quadrangulation with $n+3$ vertices is a rooted 1-stem plane tree with n nodes.*

Proof. Each inner vertex has two outgoing edges: one in H (because a non-root vertex has outdegree 1 in a tree oriented to the root), and one in $\overline{H}$. Hence, each inner vertex carries one stem at the end of the opening. □

The inverse mapping is based on closure operations, in a similar way as for tetravalent maps.

Local closure. Let T be a rooted 1-stem plane tree. Each time we find a succession (stem,edge,edge,edge) in a ccw traversal around the tree, we merge the stem with the extremity of the third edge, so as to *close* a quadrangular face, see Figure 4(a). At each step of the closure, the contour of the outer face consists of a succession of unmatched stems and edges. Accordingly, the contour of the outer face is encoded by a (cyclic) word: during a ccw traversal of the outer face, write a letter s when a stem is crossed and a letter e when traversing an edge. In the beginning this cyclic word w has thus n letters s and $2n-2$ letters e. Then each closure consists in searching a factor $seee$ and replacing it by a letter e. In other words, local closures

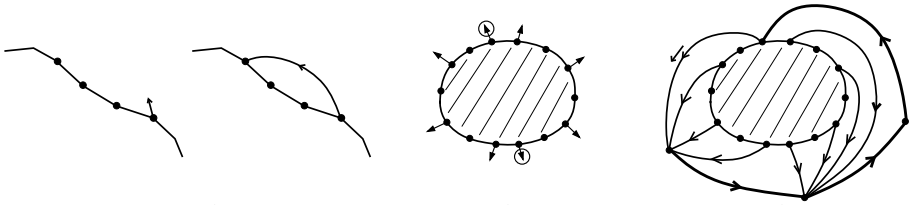


FIGURE 4. Closure of a balanced rooted 1-stem tree into a rooted complete quadrangulation.

correspond to the parenthesis matchings of the word, with an opening parenthesis of weight 3 for each letter s and a closing parenthesis of weight 1 for each letter e. As for tetravalent maps, this ensures that the figure obtained by performing greedily local closures does not depend on the order of the closures.

When all local closures have been performed greedily, the cyclic word w contains no factor $seee$. In fact it is easily shown, based on simple counting arguments [1], that w is now of the form $w = se(see)^*se(see)^*$, i.e., contains two factors se separated by two sequences of factors see, see Figure 4(b). We write $\{s_1, s_2\}$ for the two occurences of s followed by a unique e. Then, we add a branch of length 3 carrying no stem as the rightmost branch of the tree, i.e., the branch is attached to the root vertex in the angle incident to the outer face and delimited to the left by the root stem. Accordingly, in the cyclic word w, we add three letters eee just before the letter s corresponding to the root stem (the added final pattern eee corresponding to the three left sides of the edges of the added branch). It is easily proved that all remaining stems can be closed greedily, yielding a rooted quadrangulation. Moreover the obtained rooted quadrangulation is complete iff the root stem corresponds to s_1 or s_2, see Figure 4(c). In this case the tree is called *balanced*. Similar arguments as for rooted tetravalent maps ensure that the opening and closure mapping are mutually inverse, yielding a bijection between balanced rooted 1-stem plane trees and rooted complete quadrangulations endowed with an opening edge-partition.

PROPOSITION 3.2 (Recover bijection of [**56**]). *Rooted quadrangulations with $n+2$ vertices are in bijection with balanced rooted 1-stem plane trees with n nodes. As a consequence, the number q_n of rooted quadrangulations with $n+2$ vertices satisfies*

$$q_n = 2\frac{(3n-3)!}{n!(2n-1)!}. \tag{7}$$

Proof. A rooted 1-stem plane tree with n nodes is encoded by a word in the following way. Walk around the tree in counter-clockwise order, starting from the angle to the left of the root stem. Write a letter a each time an edge is traversed from top to bottom, and write a letter b each time a stem is crossed or an edge is traversed from bottom to top. Finish the word with a letter b corresponding to the root stem. Then the obtained word W clearly contains $(n-1)$ a's and $(2n-1)$ b's. In addition, by giving weight 2 to each letter a and weight -1 to each letter b, the tree structure ensures that the weight of W is -1 and the weight of each strict prefix

[1]The number of letters s decreases by 1 and the number of letters e decreases by 2 for each local closure, hence $2|w|_s - |w|_e$ remains equal to 2 during the closure.

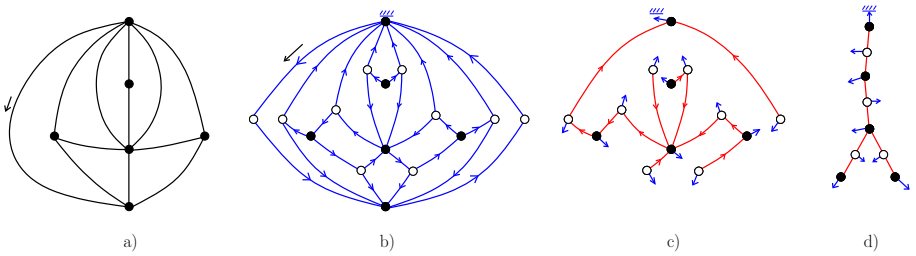


FIGURE 5. Opening of a rooted triangulated map.

of W is nonnegative. Conversely, any word in $\mathfrak{S}(a^{n-1}b^{2n-1})$ satisfying this prefix condition gives rise to a rooted 1-stem plane tree, and the two mappings are inverse of one another. Finally, each cyclic class of words in $\mathfrak{S}(a^{n-1}b^{2n-1})$ contains $3n-2$ words, and contains exactly one word satisfying the prefix condition. Hence, the number of rooted 1-stem trees with n nodes is $|\mathcal{F}_n| = \frac{1}{3n-2}\binom{3n-2}{n-1}$. Finally, as we have seen, a 1-stem plane tree with n nodes (hence n stems) has exactly two stems (s_1 and s_2 in the discussion above) whose rooting yields a balanced tree. Hence the proportion of balanced trees in $\mathcal{F}_n$ is $2/n$. The result follows. □

Remark. Our formulation of the bijection is quite close to the original bijection [**56**], which already makes use of the minimal 2-orientation. The difference is that we make the orientation root-accessible by orienting the outer face ccw, and we slightly change the family of quadrangulations, considering *complete ones*, in order to have a simple conjugation principle for the trees (as we have seen, a proportion $2/n$ is balanced). The end of the closure procedure is also different: addition of two vertices receiving remaining stems in [**56**], addition of a branch of length 3 in our construction.

1.4. Bijection with rooted triangulated maps. Recall that a triangulated map is a loopless map such that all faces have degree 3. The difference with triangulations is that multiple edges are allowed. When all faces have degree 3, a simple argument ensures that the absence of loop is equivalent to the absence of separating vertex. Hence, a triangulated map is a 2-connected map such that all faces have degree 3.

LEMMA 3.3. *Rooted triangulated maps with n vertices are in bijection with the family $\mathcal{Q}_n^{(3)}$ of rooted quadrangulations with n black vertices, all inner white vertices of degree 3, and the two outer white vertices of degree 2.*

Proof. According to Theorem 1.2 (angular mapping), rooted triangulated maps with n vertices are in bijection with rooted quadrangulations with n black vertices and such that all white vertices have degree 3. Upon adding two paths of length 2 in the outer face (see Figure 5(b)), these quadrangulations are in turn in bijection with the family $\mathcal{Q}_n^{(3)}$. □

Let $Q \in \mathcal{Q}_n^{(3)}$ be endowed with its minimal 2-orientation. Orient ccw the outer face of Q (so as to make the orientation root-accessible) and let $(H, \overline{H})$ be the opening edge-partition of Q for this orientation. Then we obtain a blossoming tree by performing the following operations.

Opening mapping.

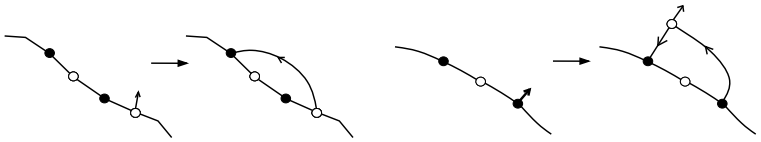


FIGURE 6. The two types of local closure in the bijection related to triangulated maps.

(1) For each edge $e = (v, v') \in \overline{H}$, delete the ingoing half-edge of e, the remaining half-edge is specified as a *stem* going out of v.
(2) Delete the non-root outer edges and the 3 non-root outer vertices (see Figure 5(c)).
(3) For each white vertex v having indegree 0 in H, delete v and the edge of H going out of v.

In addition, specify the root stem as the outgoing half-edge of the root. Operations 1 and 2 are the same as the opening of a rooted complete quadrangulation, as described in Section 1.3. The obtained rooted 1-stem tree has the particular feature that white vertices either have one neighbour, i.e., they are leaves, or have two neighbours, i.e., they can be seen as the middle of an edge connecting two black nodes. As explained later in the description of the closure mapping, the information given by the white leaves is not needed to encode the opening edge-partition. These vertices are thus deleted in the further step 3.

Definition. Let $\mathcal{F}^{(3)} = \cup_n \mathcal{F}_n^{(3)}$ be the family of bicolored 1-stem plane trees such that the root stem is incident to a black node and each white vertex has degree 2. These trees are counted with respect to the number n of black nodes.

LEMMA 3.4. *The opening of a bicolored rooted quadrangulation $Q \in \mathcal{Q}_{n+1}^{(3)}$ is in $\mathcal{F}_n^{(3)}$.*

Proof. Before Step 3 of the opening, the tree H is in the family $\mathcal{F}_n$ defined in Section 1.3. As all white vertices have degree 3 in Q, the nodes of H either have indegree 1 or indegree 0 in H. As the nodes of indegree 0 are deleted at Step 3, the tree finally obtained is in $\mathcal{F}_n^{(3)}$. □

Again the inverse procedure consists in performing greedily local closure operations, this time of two different types, (see Figure 6):

Local closures. Let T be a rooted tree in $\mathcal{F}_n^{(3)}$.

- If a stem s incident to a white node is followed by (at least) 3 edges in a ccw walk around the tree, create an opposite half-edge to s incident to the extremity of the third edge, so as to close a quadrangular face, see Figure 6(a).
- If a stem s incident to a black node is followed by (at least) 2 edges in a ccw traversal around the tree, create a white vertex v carrying a stem and connected by a 1-leg to the extremity of the second edge. Then create an opposite half-edge to s incident to v so as to close a quadrangular face, see Figure 6(b).

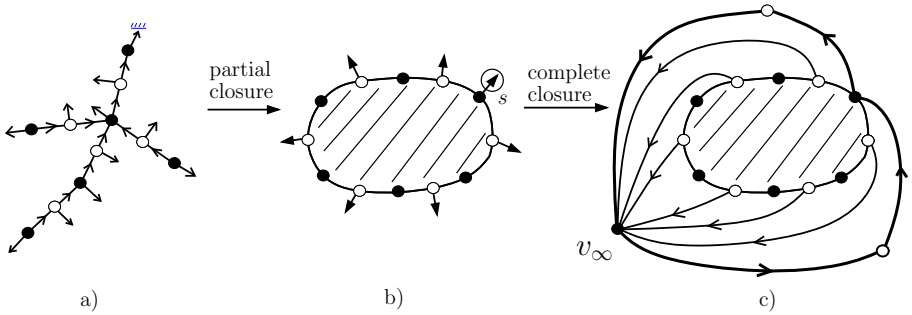


FIGURE 7. Partial closure and then complete closure of a blossoming tree.

The figure F obtained after greedy application of these local closure operations is called the *partial closure* of T. Then we perform a *complete closure* operation in order to get a map in $\mathcal{Q}^{(3)}_{n+1}$. Consider the sequence of unmatched stems in a ccw traversal around the outer face of F. Clearly two consecutive stems s_i, s_{i+1} are separated by at most two edges if s_i is incident to a white vertex and are separated by one edge if s_i is incident to a black vertex. Moreover, a simple counting argument ensures that the number of 1-edge intervals is equal to two, all other intervals being 2-edge intervals. Then the fact that the colors of vertices alternate ensures that the two 1-edge intervals are consecutive and separated by a stem s incident to a black node, and that all other unmatched stems are incident to white nodes. This particular stem is surrounded in Figure 7. The tree is called *balanced* if the root stem is s.

Complete closure. The *complete closure* consists in the following operations (see Figure 7); create a black vertex v_∞ in the outer face, connect to v_∞ all unmatched stems incident to white vertices, and delete the stem s. Finally, add two paths of length 2 connecting the root vertex to the opposite outer vertex so that the two paths form the new contour of the outer face, and orient the outer face ccw. In this way we obtain a quadrangulation $Q \in \mathcal{Q}^{(3)}_n$ endowed with its opening edge-partition. We obtain a bijective construction, as the tree T encodes (via the closure operations) the opening edge-partition of the obtained quadrangulation Q; and, conversely, the tree associated to the quadrangulation is read from the opening edge-partition using the opening mapping.

PROPOSITION 3.3 (Recover bijection of [**93**]). *Rooted triangulated maps with $n+1$ vertices are in bijection with balanced trees of the family $\mathcal{F}^{(3)}_n$. As a consequence the number $t^{(2)}_n$ of rooted triangulated maps with $n+1$ vertices satisfies*

$$t^{(2)}_n = \frac{2^{n-1}(3n-3)!}{(2n-1)!n!}. \tag{8}$$

Proof. The bijection is already justified in the preceding discussion. Thus it remains to count balanced rooted trees in $\mathcal{F}^{(3)}_n$. Clearly, a tree of $\mathcal{F}^{(3)}_n$ can be seen as a rooted 1-stem plane tree with n nodes colored black, where a white vertex is put at the middle of each edge, the white vertex carrying a stem on either side of the edge. Hence, $|\mathcal{F}^{(3)}_n|$ is equal to $2^{n-1}|\mathcal{F}_n| = 2^{n-1}(3n-3)!/((2n-1)!(n-1)!)$. Finally, as we have seen in the description of the closure, exactly one stem (among the n stems

incident to black nodes) yields a balanced tree. Hence, the proportion of balanced rooted trees in $\mathcal{F}_n^{(3)}$ is $1/n$. The result follows. □

Remark. Our bijection is very different from the original bijection, which operates on the dual map of the triangulated map (we operate on the angular map) and does not use orientations. In addition, only the mapping from trees to triangulated maps is explicitly given in [**93**], the bijectivity being proved by induction. Let us mention that Bernardi has found an extremely elegant (and very different) method to count triangulated maps, while simultaneously counting the so-called *Kreweras walks* [**9**].

1.5. Bijection with rooted triangulations. The bijection to count rooted triangulations is very similar to the one used to count rooted quadrangulations, the 3-orientations of triangulations playing the same role as 2-orientations for quadrangulations. First, it is more convenient to work with rooted triangulations of the 4-gon such that the vertex following the root vertex in clockwise order around the outer face has degree 2. Such triangulations of the 4-gon are called *complete*. This is a minor change, as there is a bijection between rooted triangulations with n vertices and rooted complete triangulations of a 4-gon with $n+1$ vertices. Indeed, starting from a rooted triangulation T, the completion-step consists in adding an outer path of length 2 connecting the root vertex r to the vertex following r in clockwise order around the outer face. The outer edge of T now covered by the path of length 2 is called the *covered edge*.

Given a rooted triangulation T endowed with its minimal 3-orientation, we perform the completion step $M := \text{Complete}(T)$, orient the outer (quadrangular) face counter-clockwise, and orient the covered edge out of the root vertex. In this way, the inner vertices have outdegree 3, the root vertex has outdegree 2, and the non-root outer vertices have outdegree 1. We consider the opening edge-partition $(H, \overline{H})$ associated with this orientation and define the following opening procedure:

Opening mapping.

(1) For each edge $e \in \overline{H}$, delete the ingoing half-edge of e.
(2) Delete the non-root outer edges of Q as well as the non-root outer vertices.

Definition. A *rooted 2-stem plane tree* is plane tree such that each node carries two stems, the tree being rooted at a stem.

LEMMA 3.5. *The opening of a rooted complete triangulation of the 4-gon with $n+3$ vertices is a rooted 2-stem plane tree with n nodes.*

Proof. This is an easy consequence of the outdegree condition and the way we have defined the opening, see Figure 3. □

Conversely, starting from a so-called balanced rooted 2-stem plane tree (balanced being defined similarly as in the bijection with quadrangulations), one obtains a rooted complete triangulation of the 4-gon in a very similar way as described for quadrangulations. Figure 8 illustrates the main steps (details are ommited).

PROPOSITION 3.4 (Recover bijection of [**95**]). *Rooted triangulations with $n+2$ vertices are in bijection with balanced 2-stem plane trees with n nodes. As a consequence, the number $t_n^{(3)}$ of rooted triangulations with $n+2$ vertices satisfies*

$$t_n^{(3)} = 2\frac{(4n-3)!}{n!(3n-1)!}. \tag{9}$$

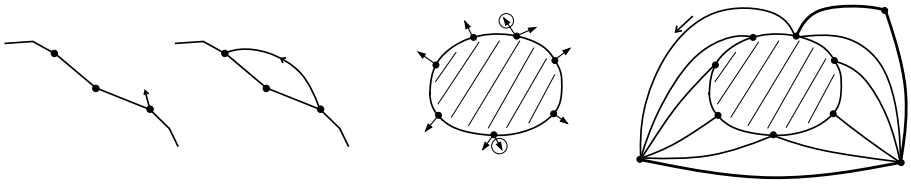


FIGURE 8. Closure of a balanced rooted 2-stem tree into a rooted complete triangulation of the 4-gon: local closure operation (Fig.a), the generic figure obtained after greedily performing local closures (Fig.b), and complete closure (Fig.c). The two stems whose rooting yields a balanced tree are surrounded in Fig.b.

Proof. Rooted triangulations with $n+2$ vertices are in bijection with rooted complete triangulations of the 4-gon with $n+3$ vertices. These are in bijection with balanced rooted 2-stem plane trees with n nodes. In a similar way as in the proof of Proposition 3.2, the number of rooted 2-stem plane trees with n nodes is shown to be $(4n-3)!/((n-1)!(3n-1)!)$, and the proportion of balanced ones is $2/n$. The result follows. □

Remark. Our formulation is close to the original bijection [**95**]. As for quadrangulations (Proposition 3.2), the difference is that we make the orientation root-accessible and slightly change the family in order to have a simple argument to count the family of trees, considering complete triangulations of the 4-gon instead of triangulations. The end of the closure procedure is also different: addition of two vertices receiving remaining stems in [**95**], addition of a branch of length 3 in our construction.

2. Bijections not depending on a root

In Section 1, we have presented a general method, based on root-accessible α-orientations, to establish bijections between families of rooted maps and families of blossoming trees. As we have seen, the most difficult part is often to find a simple characterisation of the family $\mathcal{T}$ of blossoming trees, in order to count these trees. In this section, we focus on the families of 3-connected maps and irreducible triangulations of the 4-gon. These two families are both characterised by specific α-orientations (on the derived map), as we have seen in Chapter 1, and slight modifications in the outer face make these orientations root-accessible. Hence, the method of Section 1 can be applied, but unfortunately we have not been able to find a simple characterisation of the obtained families of blossoming trees.

We overcome this difficulty by introducing bijections of another type to handle these two classes of maps. The originality of these bijections is that they work for *unrooted* objects, so that they have the advantage of yielding enumeration formulas for the number of unrooted (as well as rooted) maps of the family. The mapping from trees to maps still relies on local closures; however the inverse mapping does not work directly with α-orientations but with associated orientations of the half-edges. The two bijective correspondences we present in this section are very similar; the first one is from binary trees to irreducible quadrangulated dissections of the hexagon (which are closely related to 3-connected maps), the second one is from

ternary trees to irreducible triangulations. In contrast to Section 1, we first describe the mapping from trees to maps, and then explain how to recover the tree from the map. We also provide more details than in Section 1, as the bijections described here are new.

2.1. Bijection between binary trees and some dissections.

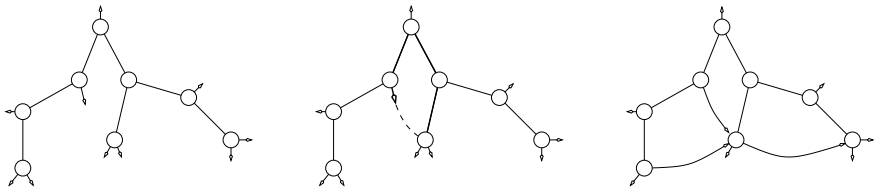
2.1.1. *Closure mapping: from trees to dissections. Binary trees* are plane trees whose vertices have degrees in $\{1, 3\}$. The vertices of degree 3 are called *nodes*, and the vertices of degree 1 are called *leaves*. Edges incident to a leaf are called *stems*, and the other are called *closed edges*. A *rooted binary tree* is a binary tree rooted at a stem. The root-edge of a rooted binary tree thus connects a node, called the *root-node*, to a leaf, called the *root-leaf*. With this definition of rooted binary trees, upon drawing the tree in a top down manner starting with the root-leaf, every node (including the root-node) has a father, a left son and a right son. This (very minor) variation on the usual definition of rooted binary trees will be convenient later on. For $n \geq 1$, we denote respectively by $\mathcal{B}_n$ and $\mathcal{B}'_n$ the sets of binary and rooted binary trees with n nodes (they have $(n+2)$ leaves, as proved by induction on n). These rooted trees are well known to be counted by the Catalan numbers: $|\mathcal{B}'_n| = \frac{1}{n+1}\binom{2n}{n}$.

The nodes of a binary tree can be greedily bicolored (say in black or white) so that adjacent nodes have distinct colors. The bicoloration is unique up to the choice of the color of the first node. As a consequence, rooted bicolored binary trees are either *black-rooted* or *white-rooted*, depending on the color of the root node. The sets of black-rooted (white-rooted) binary trees with i black nodes and j white nodes is denoted by $\mathcal{B}^{\bullet}_{ij}$ (by $\mathcal{B}^{\circ}_{ij}$, respectively); and the whole set of rooted bicolored binary trees with i black nodes and j white nodes is denoted by $\mathcal{B}'_{ij}$.

It will be convenient to view each closed edge of a tree as a pair of opposite *half-edges* (each one incident to one extremity of the edge), and to view each stem as a single half-edge (incident to the node holding the stem). More generally we shall consider maps that have *closed edges* (made of two half-edges) and *stems* (made of only one half-edge). It is then also natural to associate one face to each half-edge (say, the face on its right). In the case of trees, there is only the outer face, so that all half-edges get the same associated face.

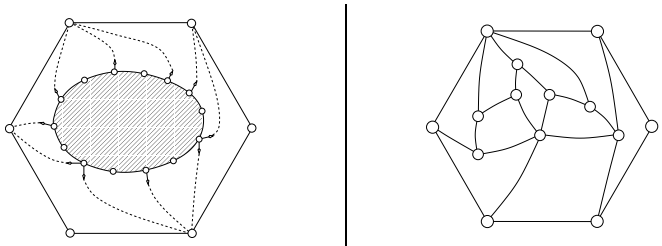
Let us recall the definition of irreducible dissections of the hexagon, which are central here. An *irreducible dissection* of the hexagon —shortly called irreducible dissection— is a planar map with outer face of degree 6, inner faces of degree 4 and no separating 4-cycle. Rooted irreducible dissections are naturally endowed with the unique bicoloration of vertices in black and white such that the origin of the root is black. The set of (rooted) irreducible dissections with n inner vertices is denoted by $\mathcal{D}_n$ ($\mathcal{D}'_n$, respectively), and the set of (rooted) irreducible dissections with i inner black vertices and j inner white vertices is denoted by $\mathcal{D}_{i,j}$ ($\mathcal{D}'_{i,j}$, respectively).

Local and partial closure. Given a map with closed edges and stems (for instance a tree), we define a *local closure* operation, which is based on a counter-clockwise walk around the map: this walk alongside the border of the outer face visits a succession of stems and closed edges, or more precisely, a sequence of half-edges having the outer face on their right-hand side. When a stem is immediately followed in this walk by three closed edges, its local closure consists in the creation of an opposite half-edge for this stem, which is attached to farthest endpoint of the third inner edge: this amounts to completing the stem into a closed edge, so as to create (or *close*) a quadrangular face. This operation is illustrated by Figure 9(b).



(a) A binary tree, (b) a local closure, (c) and its partial closure.

FIGURE 9. The partial closure.



(a) Generic case when $r = 2$ and $s = 2$.

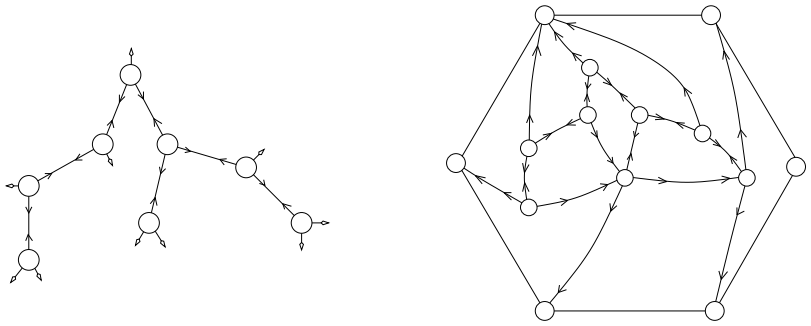
(b) Case of the binary tree of Figure 9(a).

FIGURE 10. The complete closure.

Given a binary tree T, the local closure can be performed greedily until no more local closure is possible. Each local closure creates a new closed edge, maybe making a new local closure possible. It is easy to see that the final map, called the *partial closure* of T, does not depend on the order of the local closures. Indeed, a cyclic parenthesis word is associated to the counter-clockwise contour of the tree, with an opening parenthesis (of weight 3) for a stem and a closing parenthesis for a side of closed edge; then the future local closures correspond to matchings of the parenthesis word. An example of partial closure is shown in Figure 9(c).

Complete closure. Let us now complete the partial closure operation to obtain a dissection of the hexagon with quadrangular faces. An *outer closed half-edge* is a half-edge belonging to a closed edge and incident to the outer face. Observe that a binary tree T with n nodes has $n + 2$ stems and $2n - 2$ outer closed half-edges. Each local closure decreases by 1 the number of stems and by 2 the number of outer closed half-edges. Hence, if k denotes the number of (unmatched) stems in the partial closure of T, there are $2k - 6$ outer closed half-edges. Moreover, stems delimit intervals of inner half-edges on the contour of the outer face, with length at most 2, otherwise a local closure would be possible. Let r be the number of such intervals of length 1 and s be the number of such intervals of length 0 (that is, the number of nodes incident to two unmatched stems). Then r and s are clearly related by the relation $r + 2s = 6$.

The *complete closure* consists in completing all unmatched stems with half-edges incident to vertices of the hexagon in the unique way (up to rotation of the hexagon) that creates only quadrangular bounded faces. Figure 10(a) illustrates the complete closure for the case $(r = 2, s = 2)$, and a particular example is given in Figure 10(b).



(a) A tri-oriented binary tree, (b) and its tri-oriented closure.

FIGURE 11. Examples of tri-orientations.

LEMMA 3.6. *The closure of a binary tree is an irreducible dissection of the hexagon.*

Proof. Assume that there exists a separating 4-cycle $\mathcal{C}$ in the closure of T. Then there are $m \geq 1$ vertices in its interior, and $2m$ edges according to Euler's relation. Let v be a vertex of T that belongs to the interior of $\mathcal{C}$ after the closure. Consider the orientation of edges of T away from v (only for the sake of this proof). Then nodes of T have outdegree 2, except v, which has outdegree 3. This orientation naturally induces an orientation of inner edges of the closure with the same property (except that vertices of the hexagon have outdegree 0). Hence there are at least $2m+1$ edges in the interior of $\mathcal{C}$, in contradiction with the counting given by Euler's relation. □

2.1.2. *Tri-orientations and opening.*

Tri-orientations. In order to define the inverse operation of the closure, we need a better description of the structure induced on the dissection by the original tree. Let us consider orientations of the *half-edges* of a map (in contrast to the usual notion of orientation, where edges are oriented). a half-edge is said to be *inward* if it is oriented toward its origin and *outward* if it is oriented out of its origin. If a map is endowed with an orientation of its half-edges, the *outdegree* of a vertex v is naturally defined as the number of its incident half-edges oriented outward. The (unique) *tri-orientation* of a binary tree is defined as the orientation of its half-edges such that any node has outdegree 3, see Figure 11(a) for an example. A *tri-orientation* of a dissection is an orientation of its inner half-edges (half-edges belonging to inner edges) such that outer and inner vertices have respectively outdegree 0 and 3, and such that two half-edges of a same inner edge can not both be oriented inward, see Figure 11(b). An edge is said to be *simply oriented* if its two half-edges have same direction (that is, one is oriented inward and the other one outward), and *bi-oriented* if they are both oriented outwards.

Let D be a dissection endowed with a tri-orientation. A *clockwise circuit* of D is a simple cycle $\mathcal{C}$ consisting of edges that are either bi-oriented or simply oriented with the interior of $\mathcal{C}$ on their right.

LEMMA 3.7. *A tri-orientation of an element of $\mathcal{D}_n$ has $n-1$ bi-oriented edges and $n+2$ simply oriented edges.*

If a tri-orientation of a dissection has no clockwise circuit, then its bi-oriented edges form a tree spanning the inner vertices of the dissection.

Proof. Let $D \in \mathcal{D}_n$, endowed with a tri-orientation, and let s and r denote the numbers of simply and bi-oriented edges of D. According to Euler's relation (using the degrees of the faces), D has $2n+1$ inner edges, i.e., $2n+1 = r+s$. Moreover, as all inner vertices have outdegree 3, $3n = 2r+s$. Hence $r = n-1$ and $s = n+2$.

If the tri-orientation has no clockwise circuit, the subgraph T induced by the bi-oriented edges has $r = n-1$ edges, no cycle (otherwise the cycle could be traversed clockwise, as all its edges are bi-oriented), and is incident to at most n vertices, which are the inner vertices of D. According to a classical result of graph theory, T is a tree spanning the n inner vertices of D. □

Closure-tri-orientation of a dissection. Let D be a dissection obtained as the closure of a binary tree T. The tri-orientation of T clearly induces *via* the closure operation a tri-orientation of D, called *closure-tri-orientation*. On this tri-orientation, bi-oriented edges correspond to inner edges of the original binary tree, see Figure 11(b).

LEMMA 3.8. *A closure-tri-orientation has no clockwise circuit.*

Proof. Since the vertices of the hexagon have outdegree 0, they can not belong to any circuit. Hence clockwise circuits may only be created during a local closure (not during the complete closure). However closure edges are simply oriented with the outer face on their right, hence may only create counterclockwise circuits. □

This property is indeed quite strong: the following theorem ensures that the property of having no clockwise circuit characterises the closure-tri-orientations. It also ensures the existence of such a tri-orientation for any irreducible dissection. The proof of this theorem is delayed to Section 3.

THEOREM 3.2. *Any irreducible dissection has a unique tri-orientation without clockwise circuit.*

Recovering the tree: the opening mapping. Lemma 3.7 and the present section give all necessary elements to describe the inverse mapping of the closure, which is called the *opening*. Let D be an irreducible dissection endowed with its (unique by Theorem 3.2) tri-orientation without clockwise circuit. Then the opening of D is the binary tree consisting of the inner vertices and the half-edges of D that are oriented outwards.

2.1.3. *The closure is a bijection.* In this section, we show that the opening is the inverse mapping of the closure. By construction of the opening, the following lemma is straightforward:

LEMMA 3.9. *Let $D \in \mathcal{D}_n$ be an irreducible dissection obtained as the closure of a binary tree $T \in \mathcal{B}_n$. Then the opening of D is T.*

LEMMA 3.10. *Let $T \in \mathcal{B}_n$ be a binary tree obtained as the opening of an irreducible dissection $D \in \mathcal{D}_n$. Then the closure of T is D.*

Proof. The proof relies on the definition of an order for removing inward half-edges. Start with the half-edges incident to outer-vertices (that are all oriented inward): this clearly inverses the completion step of the closure. Each further removal must correspond to a local closure, that is, the removed half-edge must have the outer face on its right.

Let M_k be the submap of the dissection induced by remaining half-edges after k removals. Then M_k covers the n inner vertices, and, as long as some inward

half-edge remains, it has at least n edges (see Lemma 3.7). Hence, it has at least a cycle, and a simple one can be extracted from the contour of the outer face of M_k. Since there is no clockwise circuit, at least one of its edges is simply oriented with the interior of the cycle on its left, and the corresponding inward half-edge can be selected for the next removal. □

Assuming Theorem 3.2, the bijection follows from Lemma 3.9 and Lemma 3.10:

THEOREM 3.3 (bijection). *The closure mapping is a bijection between the set $\mathcal{B}_n$ of binary trees with n nodes and the set $\mathcal{D}_n$ of irreducible dissections with n inner vertices. The inverse mapping of the closure is the opening.*

We can state three analogous versions of Theorem 3.3 for rooted objects:

THEOREM 3.4 (bijection, rooted versions). *The closure mapping induces the following bijections between sets of rooted objects:*

$$\begin{aligned} \mathcal{B}'_n \times \{1,\dots,6\} &\equiv \mathcal{D}'_n \times \{1,\dots,n+2\}, \\ \mathcal{B}'_{ij} \times \{1,2,3\} &\equiv \mathcal{D}'_{ij} \times \{1,\dots,i+j+2\}, \\ \mathcal{B}^\bullet_{ij} \times \{1,2,3\} &\equiv \mathcal{D}'_{ij} \times \{1,\dots,2i-j+1\}. \end{aligned}$$

Proof. We define the set $\mathcal{D}''_n$ of so-called *bi-rooted irreducible dissections* of the hexagon. An object of $\mathcal{D}''_n$ is an object of $\mathcal{D}'_n$ endowed with its tri-orientation without clockwise circuit and where a simply oriented edge is marked. Opening and rerooting on the corresponding stem defines a surjection from $\mathcal{D}''_n$ onto $\mathcal{B}'_n$, for which each element of $\mathcal{B}'_n$ has clearly 6 preimages since the dissection could have been rooted at any edge of the hexagon. Moreover, erasing the mark clearly defines a surjection from $\mathcal{D}''_n$ to $\mathcal{D}'_n$, for which each element of $\mathcal{D}'_n$ has $n+2$ preimages, according to Lemma 3.7. Hence, the closure defines a $(n+2)$-to-6 mapping between $\mathcal{B}'_n$ and $\mathcal{D}'_n$. The proof of the $(i+j+2)$-to-3 correspondence between $\mathcal{B}'_{ij}$ and $\mathcal{D}'_{ij}$ is the same.

The $(2i-j+1)$-to-3 correspondence between $\mathcal{B}^\bullet_{ij}$ and $\mathcal{D}'_{ij}$ induced by the closure can be proved similarly, with the difference that the marked simply oriented edge has to have a black vertex as origin. Then the result follows from the fact that an object of $\mathcal{D}'_{ij}$ endowed with its tri-orientation without clockwise circuit has $(2i-j+1)$ simply oriented edges whose origin is a black vertex. □

Let us mention that the $(i+j+2)$-to-3 correspondence between $\mathcal{B}'_{ij}$ and $\mathcal{D}'_{ij}$ is a key ingredient to the planar graph generators presented in Section 4.

The coefficient $|\mathcal{B}'_n|$ is well-known to be the n-th Catalan number $\frac{1}{n+1}\binom{2n}{n}$, and refinements of the standard proofs yield $|\mathcal{B}^\bullet_{ij}| = \frac{1}{2j+1}\binom{2j+1}{i}\binom{2i}{j}$, as detailed below (Section 2.1.5). Theorem 3.4 thus implies the following enumerative results:

COROLLARY 3.1 (counting rooted dissections). *The coefficients counting rooted irreducible dissections have the following expressions,*

$$|\mathcal{D}'_n| = \frac{6}{n+2}|\mathcal{B}'_n| = \frac{6}{(n+2)(n+1)}\binom{2n}{n}, \tag{10}$$

and

$$|\mathcal{D}'_{ij}| = \frac{3}{2i-j+1}|\mathcal{B}^\bullet_{ij}| = \frac{3}{(2i+1)(2j+1)}\binom{2j+1}{i}\binom{2i+1}{j}. \tag{11}$$

These enumerative results have already been obtained by Mullin and Schellenberg [**90**] using algebraic methods. Our result provides a bijective interpretation of these numbers for $m = 2$.

Notice that the cardinality of $\mathcal{D}'_n$ is $S(n,2)/2$ where $S(n,m) = \frac{(2n)!(2m)!}{n!m!(n+m)!}$ is the nth super-Catalan number of order m (these numbers are discussed by Gessel in [**67**]). Our bijection gives an interpretation of these numbers for $m = 2$.

2.1.4. *Counting triangulations.* A nice feature of the closure mapping is that it specializes to a bijection between plane triangulations and a simple subfamily of binary trees. In this way, we get the first bijective proof of the formula giving the number of unrooted plane triangulations with n vertices, found by Brown [**27**], and recover the counting formula for rooted triangulations, already obtained in [**95, 111**] and in Section 1.5.

THEOREM 3.5. *The closure mapping induces a bijection between the set $\mathcal{T}_n$ of (unrooted) plane triangulations with n inner vertices and the set $\mathcal{S}_n$ of bicolored binary trees with n black nodes and no stem (i.e., leaf) incident to a black node.*

The closure-mapping induces the following correspondence between the set $\mathcal{T}'_n$ of rooted triangulations with n inner vertices and the set $\mathcal{S}'_n$ of rooted trees in $\mathcal{S}_n$:

$$\mathcal{S}'_n \times \{1,2,3\} \equiv \mathcal{T}'_n \times \{1,\dots,3n+3\}.$$

Proof. Plane triangulations are exactly 3-connected planar maps such that all faces have degree 3. The *angular dissection* of a triangulation T is the dissection of the hexagon obtained from the angular map of T by splitting the white vertex corresponding to the outer face of T into three white vertices, one in front of each outer edge of T, see Figure 12 such that the three outer white vertices have degree 2 and all inner white vertices have degree 3. It is in fact easily checked that this mapping is a bijection between $\mathcal{T}_n$ and the set of bicolored irreducible dissections with n inner black vertices and with all inner white vertices having degree 3 and the outer white vertices having degree 2. In a tri-orientation, the indegree of each inner white vertex v is $\deg(v) - 3$ and the indegree of each outer white vertex v is $\deg(v)-2$, so that the dissections considered here have no ingoing half-edge incident to a white vertex. Hence the opening of the dissection (by removing ingoing half-edges) is a binary tree with no stem incident to a black node. Conversely, starting from such a binary tree, the half-edges created during the closure mapping are opposite to a stem. As all stems are incident to white vertices, the half-edges created are incident to black vertices. Hence the degree of each white vertex does not increase during the closure mapping, i.e., it remains equal to 3 for inner white vertices and equal to 2 for outer white vertices. This concludes the proof of the bijection $\mathcal{S}_n \equiv \mathcal{T}_n$.

The bijection $\mathcal{S}'_n \times \{1,2,3\} \equiv \mathcal{T}'_n \times \{1,\dots,3n+3\}$ follows easily (see the proof of Theorem 3.4), using the fact that a tree of $\mathcal{S}_n$ has $3n+3$ leaves. □

The bijection, illustrated in Figure 12, makes it possible to count plane unrooted and rooted triangulations, as the subfamily of binary trees involved is easy to enumerate.

PROPOSITION 3.5 (counting triangulations). *For $n \geq 0$, the number of rooted triangulations with n inner vertices is*

$$|\mathcal{T}'_n| = 2\frac{(4n+1)!}{(n+1)!(3n+2)!}.$$

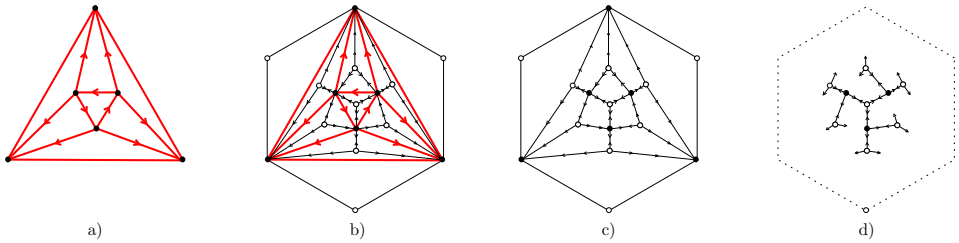


FIGURE 12. The bijection between triangulations and bicolored binary trees with no leaf incident to a black node.

The number of unrooted plane triangulations with n inner vertices is

$$|\mathcal{T}_n| = \frac{2}{3}\frac{(4n+1)!}{(n+1)!(3n+2)!} \qquad \text{if } n \equiv 2 \mod 3,$$

$$|\mathcal{T}_n| = \frac{2}{3}\frac{(4n+1)!}{(n+1)!(3n+2)!} + \frac{4}{3}\frac{(4k+1)!}{k!(3k+2)!} \qquad \text{if } n \equiv 1 \mod 3 \ \ [n=3k+1],$$

$$|\mathcal{T}_n| = \frac{2}{3}\frac{(4n+1)!}{(n+1)!(3n+2)!} + \frac{2}{3}\frac{(4k)!}{k!(3k+1)!} \qquad \text{if } n \equiv 0 \mod 3 \ \ [n=3k].$$

Proof. Let $\mathcal{S}' = \cup_n \mathcal{S}'_n$ be the set of rooted binary trees with no leaf incident to a black node and let $\mathcal{R}' = \cup_n \mathcal{R}'_n$ be the set of rooted binary trees where the root leaf is incident to a black node and all other leaves are incident to white nodes. Let $S(x)$ and $R(x)$ be the generating functions of $\mathcal{S}'$ and $\mathcal{R}'$ with respect to the number of black nodes. Clearly the two subtrees at the (white) root node of a tree of $\mathcal{S}'$ are either empty or in $\mathcal{R}'$. Hence $S(x) = (1+R(x))^2$. Similarly, a tree in $\mathcal{R}'$ decomposes at the (black) root node into two trees in $\mathcal{S}'$, so that $R(x) = xS(x)^2$. Hence, $R(x) = x(1+R(x))^4$ is equal to the generating function of quaternary trees, and $S(x) = (1+R(x))^2$ is equal to the generating function of pairs of quaternary trees (the empty tree being allowed). Using a Lukasiewicz encoding and the cyclic lemma, the number of pairs of quaternary trees with a total of n nodes is easily shown to be $\frac{2}{4n+2}\frac{(4n+2)!}{n!(3n+2)!}$. This expression of $|\mathcal{S}'_n|$ and the $(3n+3)$-to-3 correspondence between $\mathcal{S}'_n$ and $\mathcal{T}'_n$ yield the expression of $|\mathcal{T}'_n|$.

Let us now prove the formula for $|\mathcal{T}_n| = |\mathcal{S}_n|$. Clearly, the only possible non-trivial symmetry of a bicolored binary tree is a rotation of order 3. Let $\mathcal{S}_n^{\text{sym}}$ be the set of trees of $\mathcal{S}_n$ with a rotation symmetry and let $\mathcal{S}_n^{\text{asy}}$ be the set of trees of $\mathcal{S}_n$ with no symmetry. Let $\mathcal{S}_n^{'\text{asy}}$ and $\mathcal{S}_n^{'\text{sym}}$ be the sets of trees of $\mathcal{S}_n^{\text{asy}}$ and $\mathcal{S}_n^{\text{sym}}$ that are rooted. It is easily shown that a tree in $\mathcal{S}_n$ has $3n+3$ leaves. Clearly the tree gives rise to $3n+3$ rooted trees if it is asymmetric and gives rise to $n+1$ rooted trees if it is symmetric. Hence $|\mathcal{S}_n^{\text{asy}}| = |\mathcal{S}_n^{'\text{asy}}|/(3n+3)$ and $|\mathcal{S}_n^{\text{sym}}| = |\mathcal{S}_n^{'\text{sym}}|/(n+1)$. Using $|\mathcal{S}_n| = |\mathcal{S}_n^{\text{sym}}| + |\mathcal{S}_n^{\text{asy}}|$ and $|\mathcal{S}'_n| = |\mathcal{S}_n^{'\text{sym}}| + |\mathcal{S}_n^{'\text{asy}}|$, we obtain

$$|\mathcal{S}_n| = \frac{1}{3n+3}|\mathcal{S}'_n| + \frac{2}{3}|\mathcal{S}_n^{\text{sym}}|.$$

The centre of rotation of a tree in $\mathcal{S}_n^{\text{sym}}$ is either a black node, in which case $n = 3k+1$, or is a white node, in which case $n = 3k$. In the first case, a tree $\tau \in \mathcal{S}_n^{\text{sym}}$ is obtained by attaching to a black node 3 copies of a tree in $\mathcal{S}'_k$. Hence $|\mathcal{S}_{3k+1}^{\text{sym}}| = |\mathcal{S}'_k| = 2\frac{(4k+1)!}{k!(3k+2)!}$. In the second case, a tree $\tau \in \mathcal{S}_n^{\text{sym}}$ is obtained by

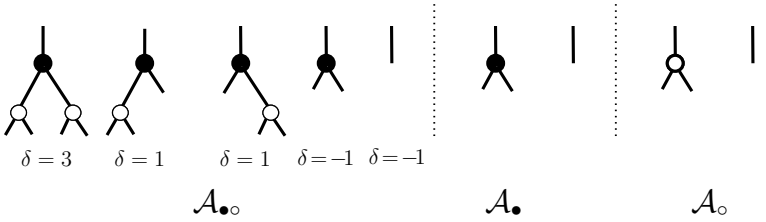


FIGURE 13. The three alphabets for words associated to bicolored binary trees.

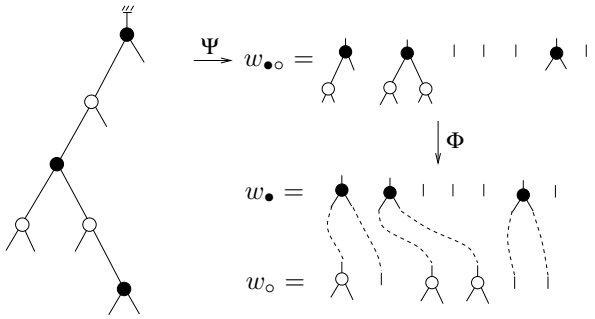


FIGURE 14. A bicolored rooted binary tree, and the corresponding words $w_{\bullet\circ}$, $w_\bullet$, and $w_\circ$.

attaching to a white node 3 copies of a tree in $\mathcal{R}'_k$. Hence $|\mathcal{S}^{\mathrm{sym}}_{3k}| = |\mathcal{R}'_k| = \frac{(4k)!}{k!(3k+1)!}$. The result follows. □

2.1.5. *Appendix of Section 2.1: counting, coding and sampling rooted bicolored binary trees.* This section provides a proof of the counting formula for the number of black-rooted bicolored binary trees with i black nodes and j white nodes. We use well-known methods of tree-encoding by parenthesis words in several parameters. The advantage of such encoding methods is to make it possible to perform random generation and encoding of bicolored binary trees, which will be useful later to encode and sample rooted 3-connected maps.

From a bicolored tree to a pair of words. This section adapts to the family of bicolored binary trees some general methods to encode a family of trees specified by several parameters. Let T be a rooted bicolored binary tree with i black nodes and j white nodes. Doing a depth-first traversal of T from left to right, we obtain a word $w_{\bullet\circ}$ of length $(2j+1)$ on the alphabet $\mathcal{A}_{\bullet\circ}$ represented on Figure 13 (left), see Figure 14 for an example, the mapping being denoted by Ψ. It is easy to see that the sum of the weights δ of the letters of any strict prefix of $w_{\bullet\circ}$ is nonnegative and the sum of the weights of the letters of $w_{\bullet\circ}$ is equal to -1. In addition, $w_{\bullet\circ}$ is the unique word in its cyclic equivalence-class that has these two properties.

Then $w_{\bullet\circ}$ is mapped to a pair $(w_\bullet, w_\circ) := \Phi(w_{\bullet\circ})$ of words such that:

- $w_\bullet$ is a word of length $(2j+1)$ on the alphabet $\mathcal{A}_\bullet$ (represented in Figure 13 middle) with i black-node-letters.
- $w_\circ$ is a word of length $2i$ on the alphabet $\mathcal{A}_\circ$ (represented in Figure 13 right) with j white-node-letters.

Figure 14 illustrates the mapping Φ on an example.

Inverse mapping: from a pair of words to a tree. Conversely, let $(w_\bullet, w_\circ)$ be a pair of words such that $w_\bullet$ is of length $(2j+1)$ on $\mathcal{A}_\bullet$ with i black-node-letters and $w_\circ$ is of length $2i$ on $\mathcal{A}_\circ$ with j white-node-letters. First, we associate to this pair a word $\widetilde{w}_{\bullet\circ}$ of

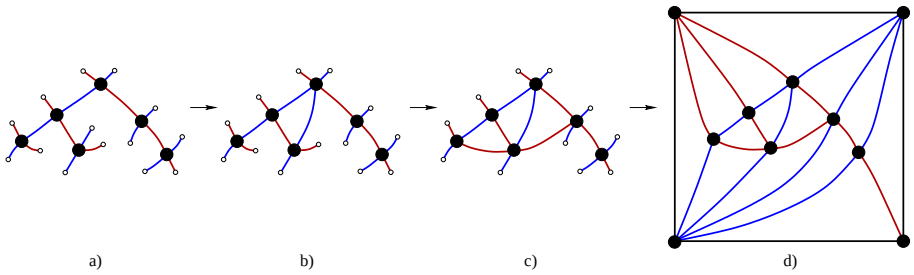


FIGURE 15. Closure of a ternary tree into an irreducible triangulation of the 4-gon.

length $(2j+1)$ on $\mathcal{A}_{\bullet\circ}$ by doing the inverse of the mapping Φ (represented on the right of Figure 14). This word has the property that the sum of the weights of its letters is equal to -1. There is a unique word $w_{\bullet\circ}$ in the cyclic equivalence-class of $\widetilde{w}_{\bullet\circ}$ such that the sum of the weights of the letters of any strict prefix is nonnegative. We associate to $w_{\bullet\circ}$ the binary tree of $\mathcal{B}^{\bullet}_{ij}$ obtained by doing the inverse of the mapping Ψ shown in Figure 14.

This method allows us to sample uniformly objects of $\mathcal{B}^{\bullet}_{ij}$ in linear time and ensures that

$$|\mathcal{B}^{\bullet}_{ij}| = \frac{1}{2j+1}\binom{2j+1}{i}\binom{2i}{j}. \tag{12}$$

2.2. Bijection between ternary trees and irreducible triangulations. The second bijection of the unrooted type we present is from ternary trees to irreducible triangulations. This bijection is very similar to the bijection with binary trees described in the preceding section. It has been the starting point of our discovery of transversal structures. Indeed, as we will see, a natural edge-bicoloration of a ternary tree is transported to the minimal transversal structure of the associated triangulation. In all this section, the set of (rooted) irreducible triangulations is denoted by $\mathcal{T}_n$ ($\mathcal{T}'_n$, respectively).

2.2.1. *Closure mapping from ternary trees to irreducible triangulations.* We define ternary trees similarly as binary trees in Section 2.1. A *ternary tree* is a plane tree whose vertex degrees are in $\{1, 4\}$. Vertices of degree 4 are called *nodes* and vertices of degree 1 are called *leaves*. An edge of A connecting two nodes is called an *closed edge* and consists of two half-edges, one at each extremity of the edge. An edge of the tree connecting a node to a leaf is called a *stem*. We consider stems as consisting of a unique half-edge incident to the node and having not (yet) an opposite half-edge. A ternary tree can be rooted by marking one of its leaves. The root allows us to distinguish the 4 neighbours of each node, taken in ccw-order, into a parent (the neighbour in the direction of the root), a left-child, a middle-child, and a right-child. Thus, what we call *rooted* ternary trees correspond to the classical definition, where each node is either a leaf or has 3 ordered children.

Like the bijection with binary trees, the mapping from ternary trees to irreducible triangulations consists of three main steps: local closure, partial closure and complete closure. Perform a counterclockwise traversal of a ternary tree A. If a stem s and then two closed edges e_1 and e_2 are successively encountered during the

traversal, create a half-edge opposite to the stem s and incident to the farthest extremity of e_2, so as to *close* a triangular face. This operation is called *local closure*, see Figure 15(b).

Now we can restart a counterclockwise traversal around the new figure F, which is identical to A, except that it contains a triangular face and, more important, the stem s has become an closed edge (i.e., an edge with two half-edges). Each time we find a succession (stem, closed edge, closed edge), we perform a local closure, update the figure, and restart, until no local closure is possible. This greedy execution of local closures is called the *partial closure* of A, see Figure 15(c). It can easily be shown that the figure F obtained by partial closure of A does not depend on the order of execution of the local closures.

At the end of the partial closure, the number n_s of unmatched stems and the number n_e of sides of closed edges incident to the outer face of F satisfy the relation $n_s - n_e = 4$. Indeed, this relation is satisfied on A because a ternary tree with n nodes has $n-1$ closed edges and it can be proved inductively on n that it has $2n+2$ leaves (so that $n_s = 2n+2$ and $n_e = 2n-2$); and the relation $n_s - n_e = 4$ remains satisfied throughout the partial closure because a local closure decreases n_s and n_e by 1, so that $n_s - n_e$ is unchanged. When no local closure is possible anymore, two consecutive unmatched stems on the contour of the outer face of F are separated by at most one side of closed edge. Hence, the relation $n_s = n_e + 4$ implies that the unmatched stems of F are partitioned into four intervals I_1, I_2, I_3, I_4, where two consecutive stems of an interval are separated by a side of closed edge, and where the last stem of I_i is incident to the same vertex as the first stem of $I_{(i+1) \bmod 4}$, see Figure 15(c).

The last step of the closure mapping, called *complete closure* consists of the following operations: draw a 4-gon (v_1, v_2, v_3, v_4) outside of F; then, for $i \in \{1, 2, 3, 4\}$, create a half-edge for each stem s of the interval I_i, this new half-edge being opposite to s and incident to the vertex v_i. Clearly, this creates only triangular faces, so that the obtained object is a triangulation of the 4-gon, see Figure 15(d).

Now we explain how the closure mapping is related to transversal edge-partitions. The edges of a ternary tree A (both closed edges and stems) can be colored blue or red so that any angle incident to an node of A is bicolored, see Figure 15(a). This bicoloration, unique up to the choice of the colors, is called the *edge-bicoloration* of A. The following invariant (I) is maintained throughout the partial closure: *any angle incident to the outer face is bicolored.* By definition of the edge-bicoloration, Invariant (I) is true on A. It remains satisfied after a local closure. Indeed, let (s, e_1, e_2) be the succession (stem, closed edge, closed edge) intervening in the local closure. Invariant (I) implies that s and e_2 have the same color. As we give to the new created half-edge h the same color as its opposite half-edge s (in order to have unicolored edges), s and e_2 have the same color. The effect of the local closure on the angles of the outer face is the following, where we write v for the incident vertex of h and h' for the ccw-follower of e_2 around v: the angle (e_1, e_2) disappears from the outer face, and the angle (e_2, h') is replaced by the angle (h, h'). As e_2 has the same color as h, the bicolored angle (e_2, h') is thus replaced by the bicolored angle (h, h'), so that (I) remains true after the local closure.

It also follows from this proof that the condition for inner vertices (i.e., the incident edges form four intervals alternating in color) remains satisfied throughout the closure, because the number of bicolored angles around a vertex is not increased, and is initially equal to 4. At the end of the partial closure, Invariant (I) ensures

that all stems of the intervals I_1 and I_3 are of one color, and all stems of the intervals I_2 and I_4 are of the other color. Hence, the condition of transversal edge-partitions for outer vertices is satisfied after the complete closure, see Figure 15(d). Thus, the closure mapping transports the edge-bicoloration of A into a transversal edge-partition of the associated triangulation of the 4-gon. The following lemma makes this statement more precise.

PROPOSITION 3.6. *The closure of a ternary tree A with n nodes is an irreducible triangulation T with n inner vertices.*

The closure transports the edge-bicoloration of A into the minimal transversal edge-partition of T.

Proof. As explained above, T is endowed with a transversal edge-partition by the closure mapping. Assume a contrario that T has a separating 3-cycle Δ. Observe that Lemma 1.1 page 45 was stated and proved without the irreducibility condition. Hence, when the four border edges of T are colored blue, each inner face of T has exactly two bicolored angles. Let $k \geq 1$ be the number of vertices inside Δ. Then Euler's relation implies that there are $2k + 1$ faces inside Δ, so that there are $4k + 2$ bicolored angles inside Δ. Moreover, Condition C1 implies that there are $4k$ bicolored angles incident to a vertex in the interior of Δ. Hence there are exactly two bicolored angles inside Δ incident to a vertex of Δ. However, for each of the three edges $\{e_1, e_2, e_3\}$ of Δ, the incident face of e_i inside Δ has at least one of its two bicolored angles incident to e_i. Hence, we have the contradiction that there are at least three bicolored angles inside Δ and incident to a vertex of Δ.

Now we show that the transversal edge-partition of T transported by the closure mapping is minimal, i.e., has no right alternating 4-cycle. Let $\mathcal{C}$ be an alternating 4-cycle of T. This cycle has been closed during a local closure involving one of the four edges of $\mathcal{C}$. Let e be this edge and let v be the origin of the stem whose completion has created the edge e. The fact that the closure of a stem is always performed with the infinite face on its right ensures that e is the right-edge of v on $\mathcal{C}$, as defined in Section 6. A second observation following from Invariant (I) is that, when a stem s is merged, the angle formed by s and by the edge following s in counterclockwise order around the origin of s is a bicolored angle. This ensures that $\mathcal{C}$ is a left alternating 4-cycle. □

2.2.2. *Inverse mapping: the opening.* In this section, we describe the inverse of the closure mapping, so as to recover a ternary tree from an irreducible triangulation. As we have seen in the proof of Invariant (I), during a local closure, the new created half-edge h is such that the cw-consecutive half-edge has the same color as h. Hence, throughout the closure of a ternary tree endowed with its edge-bicoloration, the following invariants are maintained for an half-edge h incident to an inner vertex of T:

- If the angle formed by h and its cw-consecutive half-edge is unicolored, then h has been created during a local closure.
- If the angle is bicolored, then h is one of the 4 original half-edges of A incident to v.

These invariants indicate how to inverse the closure. Given an irreducible triangulation T of the 4-gon, the *opening* of T consists of the following steps, see Figure 16 for an example:

(1) Endow T with its minimal transversal edge-partition.

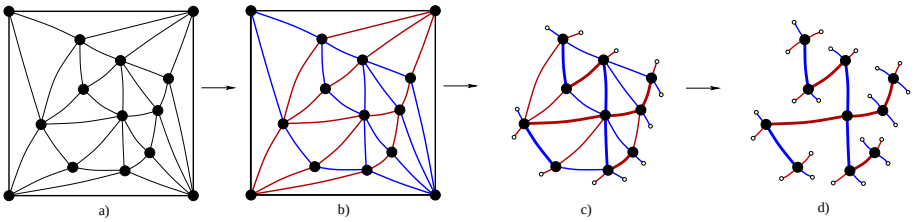


FIGURE 16. The opening algorithm performed on an example.

(2) Remove the outer quadrangle of T and all half-edges of T incident to the quadrangle.
(3) Remove all half-edges whose clockwise-consecutive half-edge has the same color.

The opening mapping has been defined so that the following lemma is satisfied.

LEMMA 3.11. *Let A be a ternary tree and let T be the irreducible triangulation obtained by doing the closure of A. Then the opening of T is A.*

Hence, the closure Φ and the opening Ψ are such that $\Psi \circ \Phi = \mathrm{Id}$. To prove that the opening is the inverse of the closure, it remains to prove that $\Phi \circ \Psi = \mathrm{Id}$, which is more difficult. First, we will show that the opening of an irreducible triangulation T is always a ternary tree. Then, we will prove that the closure of this ternary tree is T.

First we need a few definitions. Let T be an irreducible triangulation endowed with its minimal transversal edge-partition. Let h be a half-edge of T belonging to an inner edge of T and let v be the origin of h. Then we orient h using the following rules:

- If v is an outer vertex of T, orient h toward v.
- If v is an inner vertex of T, orient h outward of v if the angle formed by h and its cw-consecutive half-edge h is a bicolored angle; orient h toward v if the angle is unicolored.

Each inner vertex of T clearly has outdegree 4 for this orientation (because exactly 4 of the incident angles are bicolored). We call this orientation of the inner half-edges of T the *4-orientation* of T. By definition of the 4-orientation, the opening of a triangulation can be seen equivalently as the operation of removing the outer 4-gon and all ingoing half-edges.

Let e be an inner edge of T. An important remark is that the two half-edges of e can can not be simultaneously directed toward their respective origin. Indeed, assume a contrario that there is such an edge e, and consider the cycle $\mathcal{C}$ bordered by the two triangular faces f_1 and f_2 incident to e. Condition C2' implies easily that both extremities of e are inner vertices, so that the 4 edges of $\mathcal{C}$ are inner edges of T. The fact that f_1 and f_2 have two bicolored angles (Lemma 1.1) implies that the 4-cycle bordering the two triangular faces incident to e would be a right alternating 4-cycle, which is impossible.

Hence, only two cases arise:

- If exactly one half-edge of e is ingoing, then e is called a *stem-edge*. In this case the other half-edge of e is outgoing, so that the two half-edges of e have the same direction.

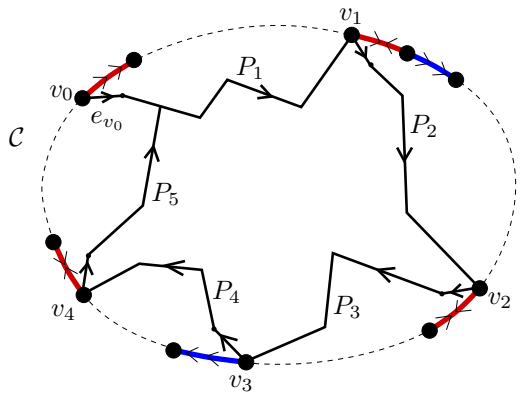


FIGURE 17. The existence of a clockwise cycle on the 4-orientation of T implies the existence of a clockwise cycle on the minimal α_4-orientation of $Q(T)$.

- Otherwise the preceding discussion ensures that the two half-edges of e are outgoing. In this case e is called a *tree-edge*. A tree-edge can be considered as a bi-oriented edge for the 4-orientation of T.

We define a clockwise circuit of the 4-orientation of T as a simple cycle $\mathcal{C}$ of inner edges of T such that each edge e of $\mathcal{C}$ is either a tree-edge (i.e. a bi-oriented edge) or a stem-edge having the interior of $\mathcal{C}$ on its right.

LEMMA 3.12. *The 4-orientation of T has no clockwise circuit.*

Proof. Assume that there is a clockwise circuit $\mathcal{C}$ in the 4-orientation of T. For a vertex v on $\mathcal{C}$, denote by h_v the half-edge of $\mathcal{C}$ going out of v when doing a clockwise traversal of $\mathcal{C}$. As $\mathcal{C}$ is a clockwise cycle for the 4-orientation of T, h_v is directed outward of v. Hence the angle $\mathcal{A}$ formed by h_v and its clockwise-consecutive half-edge around v is a bicolored angle for the minimal transversal edge-partition of T. Hence, the edge of $Q(T)$ associated with the angle $\mathcal{A}$ is going out of v for the α_4-orientation $O_{\min}$ of $Q(T)$ with no cw circuit (so-called minimal). We denote this edge by e_v. Note also that, in the interior of $\mathcal{C}$, e_v is the ccw-most edge of $Q(T)$ incident to v.

We use this observation to build iteratively a clockwise circuit of $O_{\min}$ (see Figure 17), yielding a contradiction. Let v_0 be a vertex on $\mathcal{C}$ and let $\mathcal{P}_{v_0}$ be the straight path (see the definition in page 48) starting at e_{v_0} for the α_4-orientation $O_{\min}$. Lemma 1.3 ensures that $\mathcal{P}(v_0)$ can not pass twice by the same vertex and must end at E or W. In particular, $\mathcal{P}(v_0)$ has to reach $\mathcal{C}$ at a vertex v_1 different from v_0. We denote by P_1 the part of $\mathcal{P}(v_0)$ between v_0 and v_1. We denote by Λ_1 the part of the clockwise circuit $\mathcal{C}$ between v_1 and v_0 and we call $\mathcal{C}_1$ the cycle obtained by concatenating P_1 and Λ_1. Then, let $\mathcal{P}(v_1)$ be the straight path starting at e_{v_1}. The fact that e_{v_1} is the most counterclockwise incident edge of v_1 in the interior of $\mathcal{C}$ ensures that $\mathcal{P}(v_1)$ starts in the interior of $\mathcal{C}_1$. Then the path $\mathcal{P}(v_1)$ has to reach $\mathcal{C}_1$ at a vertex $v_2 \neq v_1$. We denote by P_2 the part of the path $\mathcal{P}(v_1)$ between v_1 and v_2. If v_2 belongs to P_1, then the concatenation of the part of P_1 between v_2 and v_1 and of the part of P_2 between v_1 and v_2 is a clockwise circuit, in contradiction with the minimality of $O_{\min}$. Hence v_2 is on Λ_1 strictly between v_1 and v_0. We denote by $\overline{P}_2$ the concatenation of P_1 and P_2; and denote by Λ_2 the

part of $\mathcal{C}$ going from v_2 to v_0. As v_2 is strictly between v_1 and v_0, Λ_2 is strictly included in Λ_1. Finally, we denote by $\mathcal{C}_2$ the cycle made of the concatenation of $\overline{P}_2$ and Λ_2. Similarly as for the path $\mathcal{P}_{v_1}$, the path $\mathcal{P}_{v_2}$ must start in the interior of $\mathcal{C}_2$.

Then we continue iteratively, see Figure 17. At each step k, we consider the straight path $\mathcal{P}(v_k)$ starting at e_{v_k}. This path starts in the interior of the cycle $\mathcal{C}_k$. It reaches again $\mathcal{C}_k$ at a vertex v_{k+1}. The vertex v_{k+1} can not belong to $\overline{P}_k$, otherwise a clockwise circuit of $O_{\min}$ would be created. Hence v_{k+1} is strictly between v_k and v_0 on $\mathcal{C}$, i.e., it is on $\Lambda_k \backslash \{v_k, v_0\}$. In particular the path Λ_{k+1} going from v_{k+1} to v_0 on $\mathcal{C}$, is strictly included in the path Λ_k going from v_k to v_0 on $\mathcal{C}$. Thus, Λ_k shrinks strictly at each step. Hence, there must be a step k_0 where $\mathcal{P}(v_{k_0})$ reaches $\mathcal{C}_{k_0}$ at a vertex on $\overline{P}_{k_0}$, thus creating a clockwise circuit of $O_{\min}$ and yielding a contradiction. □

LEMMA 3.13. *The tree-edges of T form a spanning tree of the inner vertices of T.*

Proof. Denote by H the graph consisting of the tree-edges of T and their incident vertices. A first observation is that H has no cycle because such a cycle of (bi-oriented) edges of T would be a clockwise circuit in the 4-orientation of T, which is impossible according to Lemma 3.12. Let n be the number of inner vertices of T. Observe that H can not be incident to the border vertices of T, so that H can cover at most the set of inner vertices of T. A well-known result of graph theory is the following: if an acyclic graph H covers a subset of a set E of n vertices and if H has $(n-1)$ edges, then H is a tree covering exactly all vertices of E. Hence it remains to show that H has $(n-1)$ edges. Let s be the number of stem-edges and t be the number of tree-edges of T. As T has n inner vertices, there are $4n$ outgoing half-edges in the 4-orientation of T. Moreover, each stem-edge has contribution 1 to the number of outgoing half-edges and each tree-edge has contribution 2 to the number of outgoing half-edges. Hence, $s + 2t = 4n$. Finally, Euler relation ensures that T has $(3n+1)$ inner edges, so that $s + t = 3n + 1$. These two equalities ensure that $t = n - 1$, which concludes the proof that H is a spanning tree of the inner vertices. □

LEMMA 3.14. *The opening of an irreducible triangulation T is a ternary tree.*

Proof. As we have seen from the definition of the 4-orientation, the opening of a triangulation can be seen as the operation of removing the outer 4-gon and then removing all ingoing half-edges. The obtained figure consists of the tree-edges, which form a spanning tree according to Lemma 3.13, and of the half-edges that have lost their opposite half-edge. The edges of the first and second type correspond respectively to the inner edges and to the stems of the obtained tree. In addition, after removing all ingoing half-edges, each vertex has degree 4, so that the obtained tree satisfies the degree-conditions of a ternary tree. □

To prove that the closure-mapping is a bijection whose inverse is the closure, it remains to prove the following lemma:

LEMMA 3.15. *Let T be an irreducible triangulation and let A be the ternary tree obtained by doing the opening of T. Then the closure of A is T.*

Proof. First it is clear that the operation of complete closure (transition between Figure 15(c) and Figure 15(d)) is the inverse of Step 2 of the opening algorithm. Let F be the figure obtained from T after Step 2 of the opening mapping.

To prove that the partial closure of A is F, it is sufficient to find a chronological order of deletion of the ingoing half-edges of F (for the 4-orientation) such that the inverse of each half-edge deletion is a local closure. This is done in a way similar to the proof of Lemma 3.10. □

Finally Lemma 3.11 and Lemma 3.15 yield the following theorem:

THEOREM 3.6 (bijection). *The closure mapping is a bijection between the set of ternary trees with n nodes and the set of irreducible triangulations of the 4-gon with n inner vertices. The inverse mapping of the closure is the opening.*

THEOREM 3.7 (bijection, rooted version). *The closure mapping induces a 4-to-$(2n+2)$ correspondence between the set $\mathcal{A}'_n$ of rooted ternary trees with n nodes and the set $\mathcal{T}'_n$ of rooted irreducible triangulations with n inner vertices. In other words,*

$$\mathcal{A}'_n \times \{1, \ldots, 4\} \equiv \mathcal{T}'_n \times \{1, \ldots, 2n+2\}.$$

Proof. It can easily be proved by induction on the number of nodes that a ternary tree with n nodes has $2n+2$ leaves. Hence, when rooting the ternary tree obtained by doing the opening of an object of $\mathcal{T}'_n$, there are $2n+2$ possibilities to place the root. Conversely, starting from a rooted ternary tree with n nodes, there are four possibilities to place the root on the object of $\mathcal{T}'_n$ obtained by doing the closure of the tree, because the root has to be placed on one of the four border edges. □

PROPOSITION 3.7 (counting irreducible triangulations). *For $n \geq 0$, the number of rooted irreducible triangulations of the 4-gon with n inner vertices is*

$$|\mathcal{T}'_n| = 4\frac{(3n)!}{n!(2n+2)!}.$$

The number of unrooted irreducible triangulations with n inner vertices is

$$|\mathcal{T}_n| = \frac{(3n)!}{n!(2n+2)!} + \frac{1}{2}\frac{(3k)!}{k!(2k+1)!} \qquad \text{if } n \equiv 0 \mod 2 \quad [n = 2k]$$

$$|\mathcal{T}_n| = \frac{(3n)!}{n!(2n+2)!} + \frac{1}{2}\frac{(3k+1)!}{k!(2k+2)!} + \frac{1}{2}\frac{(3k')!}{k'!(2k'+1)!} \qquad \text{if } n \equiv 1 \mod 4 \quad [n = 2k+1 = 4k'+1],$$

$$|\mathcal{T}_n| = \frac{(3n)!}{n!(2n+2)!} + \frac{1}{2}\frac{(3k+1)!}{k!(2k+2)!} \qquad \text{if } n \equiv 3 \mod 4 \quad [n = 1+2k].$$

Proof. The enumerative formula follows from $|\mathcal{T}'_n| = \frac{4}{2n+2}|\mathcal{A}'_n|$ and from the well known fact that $|\mathcal{A}'_n| = (3n)!/((2n+1)!n!)$ (which can be derived from Lagrange inversion formula applied to the generating function $A(z) = z(1+A(z))^3$). The formula for $|\mathcal{T}_n| = |\mathcal{A}_n|$ follows from the enumeration of unrooted ternary trees, which is easily obtained by considering the possible rotation symmetries (order 2 around a vertex or an edge, order 4 around a vertex), in a way similar to the proof of Proposition 3.5. □

The formula for *rooted* irreducible triangulations can easily be obtained from the series counting rooted triangulations of the 4-gon by using a composition scheme, see [**111**]. To our knowledge, the counting formula for *unrooted* irreducible triangulations is new. However, a composition scheme should make it possible to count irreducible triangulations with a given rotation symmetry (order 2 around

a vertex or an edge, order 4 around a vertex), starting from triangulations of the 4-gon with a given rotation symmetry, which have been counted by Brown [**27**].

3. Appendix: proof of Theorem 3.2

This section is devoted to the proof of Theorem 3.2, which states that each irreducible dissection has a unique tri-orientation without clockwise circuit.

Theorem 1.3 ensures uniqueness of the orientation without clockwise circuit of a graph with prescribed outdegree for each vertex. However, this does not directly imply uniqueness in Theorem 3.2, because a tri-orientation has *bi-oriented* edges.

To use Theorem 1.3, we work with the derived map G' of an irreducible dissection D, as defined in Section 1.3.2. We have defined derived maps only for a subset of irreducible dissections, namely for bicolored complete irreducible dissections. As a consequence, a first step toward proving Theorem 3.2 is to reduce its proof to the proof of existence and uniqueness of a so-called complete-tri-orientation (a slight adaptation of the definition of tri-orientation) without clockwise circuit for any bicolored complete irreducible dissection.

Then we prove that a complete-tri-orientation without clockwise circuit of a bicolored complete irreducible dissection D is transposed injectively into an α_3-orientation without clockwise circuit of its derived map G'. By injectivity of the transposition and by uniqueness of the α_3-orientation without clockwise circuit of G', this implies uniqueness of a tri-orientation without clockwise circuit of D.

Conversely, we prove that an α_3-orientation without clockwise circuit of G' is transposed into a complete-tri-orientation without clockwise circuit of D. As $\alpha-3$ is feasible for any suspended 3-connected map, there exists an α_3-orientation without clockwise circuit of G', this implies the existence of a complete-tri-orientation without clockwise circuit of D, concluding the proof of Theorem 3.2.

3.1. Reduction to the case of bicolored complete dissections.

Introduction. The object of this section is to reduce the proof of Theorem 3.2 to the class of so-called *complete bicolored irreducible dissections*, which are defined as dissections such that the three outer white vertices have degree 2. We state the following proposition where the term "complete-tri-orientation", to be defined later, is a slight adaptation of the notion of tri-orientation.

Proposition 3.8. *The existence and uniqueness of a complete-tri-orientation without clockwise circuit for any bicolored complete irreducible dissection implies the existence and uniqueness of a tri-orientation without clockwise circuit for any irreducible dissection, i.e., implies Theorem 3.2.*

The rest of this subsection is devoted to the proof of Proposition 3.8. The proof is done in two steps. First, reduce the proof of Theorem 3.2 to the existence and uniqueness of a tri-orientation without clockwise circuit for any bicolored complete irreducible dissection. Then, prove that this reduces to the existence and uniqueness of a complete-tri-orientation without clockwise circuit for any bicolored complete irreducible dissection.

Completion of a bicolored irreducible dissection. For any bicolored irreducible dissection D, we define its *completed dissection* D^c as follows . For each white vertex v of the hexagon, we denote by e_1 and e_2 the two edges of the hexagon incident to v and denote by v_1 and v_2 the respective extremities of e_1 and e_2. We perform the following operation: if v has at least 3 incident edges in the dissection, a new white vertex v' is created outside of the hexagon and is linked to v_1 and to v_2 by two new edges e'_1 and e'_2, see Figure 18.

The dissection obtained is a bicolored dissection of the hexagon such that the three white vertices of the hexagon have two incident edges, see the transition between Figure 19(a) and Figure 19(b) (ignore here the orientation of edges on this figure). Hence, the obtained dissection is complete, i.e., the three outer white vertices have degree 2.

Lemma 3.16. *The completion D^c of a bicolored irreducible dissection D is a bicolored complete irreducible dissection.*

Proof. The dissection D^c is complete by construction. Hence, we just have to prove that D^c is irreducible. As D is irreducible, if a separating 4-cycle C appears in D^c when the completion is performed, then it must contain a white vertex v' of the hexagon of D^c added during the completion. Hence, using the notation of the definition of the completion, two edges of C are the edges e'_1 and e'_2 incident to v' in D^c. The two other edges e''_1 and e''_2 of C form a path of length 2 connecting the extremities v_1 and v_2 of e'_1 and e'_2 and passing by the interior of D (otherwise, C would enclose a face). As D is irreducible, the 4-cycle $\mathcal{C}'$ of D consisting of the edges e_1, e_2, e''_1 and e''_2 delimits a face. Hence e_1 and e_2 are incident to the same inner face of D, which implies that the white vertex v on the border of D and associated to v' has degree 2. But, by definition of the completion, the creation of a new vertex v' associated to v is performed only if v has at least 3 incident edges, yielding a contradiction. □

Tri-orientations. Let D be a bicolored irreducible dissection and let D^c be its completed bicolored dissection. We define a mapping Φ from the tri-orientations of D^c to the tri-orientations of D. Given a tri-orientation Y of D^c, we remove the half-edges that have been added to obtain D^c from D, erase the orientation of the edges of the hexagon of D, and orient inward all inner half-edges incident to a vertex of the hexagon of D. We obtain thus a tri-orientation $\Phi(Y)$ of D, see the transition between Figure 19(b) and Figure 19(a).

LEMMA 3.17. *Let $\widetilde{Y}$ be a tri-orientation of D^c without clockwise circuit. Then the tri-orientation $\Phi(\widetilde{Y})$ of D has no clockwise circuit.*

For each tri-orientation Y of D without clockwise circuit, there exists a tri-orientation $\widetilde{Y}$ of D^c without clockwise circuit such that $\Phi(\widetilde{Y}) = Y$.

Proof. The first point is trivial as the tri-orientation $\Phi(\widetilde{Y})$ is just obtained by removing some edges and some orientations of edges.

For the second point, the preimage $\widetilde{Y}$ is constructed as follows. Consider each white vertex v of the hexagon of D which has degree at least 3. Let $(h_1, \dots, h_m)$ $(m \geq 3)$ be the series of half-edges incident to v in D in counter-clockwise order around v, with h_1 and h_2 belonging respectively to the edges e_1 and e_2 of the hexagon that are incident to v (using the notation of Figure 18). As $m \geq 3$, the vertex v gives rise to a new vertex v' with two incident edges e'_1 and e'_2 such that the edges e_1, e_2, e'_1 and e'_2 form a new face f. Then the edges e_1 and e_2 become internal edges of D^c and have thus to be oriented in a tri-orientation of D^c.

We orient the two half-edges of e_1 and e_2 respectively toward v_1 and toward v_2, see Figure 18. Then the vertex v has degree 2 after these two orientations, and we have to give to v a third outgoing half-edge. The judicious choice to avoid the appearance of a clockwise circuit is to orient h_3 outward, see Figure 18. Indeed, assume that these new orientations of half-edges create a simple clockwise circuit $\mathcal{C}$, then the circuit must pass by v. It arrives at v by one of the half-edges h_i directed toward v, i.e., $i \geq 4$. Moreover, it must go out of v using the half-edge h_3 (indeed, if the circuit uses h_1 or h_2 to go out of v, then it arrives at a vertex on the hexagon, i.e., a vertex having outdegree 0). Hence, the interior of the clockwise circuit $\mathcal{C}$ must contain all faces incident to v that are on the right of v when we traverse v from h_i and go out using h_3. Hence, the interior of $\mathcal{C}$ must contain the new face f of D^c, see Figure 18. But f is on the border of the hexagon. Hence, the circuit $\mathcal{C}$, containing f, must pass by the edges of f that are on the border of the dissection D^c. As these edges are not oriented by definition of a tri-orientation, we have a contradiction. Thus, we have constructed a tri-orientation $\widetilde{Y}$ of D^c without clockwise circuit and such that $\Phi(\widetilde{Y}) = Y$. An example of this construction can be seen as the transition between Figure 19(a) and Figure 19(b). □

LEMMA 3.18. *The existence and uniqueness of a tri-orientation without clockwise circuit for any bicolored complete irreducible dissection implies the existence and uniqueness of a tri-orientation without clockwise circuit for any irreducible dissection, i.e., implies Theorem 3.2.*

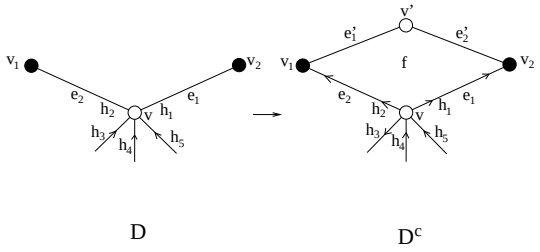


FIGURE 18. From a tri-orientation Y of D without clockwise circuit, the rules of construction of a tri-orientation $\widetilde{Y}$ of D^c without clockwise circuit such that $\Phi(\widetilde{Y}) = Y$.

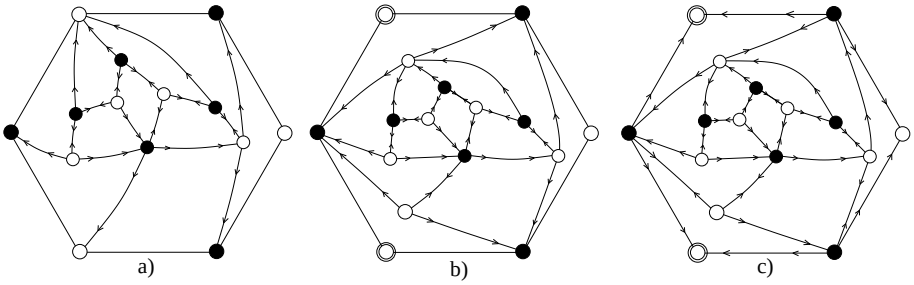


FIGURE 19. A bicolored irreducible dissection D endowed with a tri-orientation Y without clockwise circuit (Figure a). Its associated completed dissection D^c (the two added white vertices are surrounded) with a tri-orientation $\widetilde{Y}$ without clockwise and preimage of Y by Φ (Figure b). The dissection D^c with a complete-tri-orientation without clockwise circuit and preimage of $\widetilde{Y}$ by Ψ (Figure c).

Proof. This is a clear consequence of Lemma 3.16 and Lemma 3.17. □

Complete-tri-orientations. A *complete-tri-orientation* of a bicolored complete irreducible dissection D is an orientation of the half-edges of D that satisfies the following conditions (very similar to the conditions of a tri-orientation): all black vertices and all inner white vertices of D have outdegree 3, the three white vertices of the hexagon have outdegree 0, and the two half-edges of an edge of D can not both be oriented inward. The difference with the definition of tri-orientation is that the half-edges of the hexagon are oriented, with prescribed outdegree for the outer vertices. Similarly as in a tri-orientation, edges of D are distinguished into simply-oriented edges and bi-oriented edges.

LEMMA 3.19. *Let $D \in \mathcal{D}_n$ be a bicolored complete irreducible dissection endowed with a complete-tri-orientation without clockwise circuit. Then the subgraph T of D consisting of the bi-oriented edges of D is a tree incident to all vertices of D except the three outer white vertices.*

Proof. We reason similarly as in Lemma 3.7. We denote by r and s the respective number of bi-oriented and simply oriented edges of D. From Euler's relation (using the degrees of the faces of D), D has $2n+7$ edges, i.e., $r+s = 2n+7$. In addition, the n inner vertices and the three black (resp. white) vertices of the hexagon of D have outdegree 3 (resp. 0). Hence, $2r+s = 3(n+3)$. Thus, $r = n+2$ and $s = n+5$. Hence, the subgraph T has $n+2$ edges, has no cycle (otherwise, it would yield a clockwise circuit of D), and is incident to

at most $(n+3)$ vertices, which are the inner vertices and the three outer black vertices of D. A classical result of graph theory ensures that T is a tree spanning these $(n+3)$ vertices. □

LEMMA 3.20. *Let $D \in \mathcal{D}_n$ be a bicolored complete irreducible dissection endowed with a complete-tri-orientation Y without clockwise circuit. Then, each black vertex v of the hexagon has two of its outgoing half-edges on the hexagon, belonging to simply oriented edges; and the third outgoing half-edge of v belongs to a bi-oriented inner edge.*

Proof. The fact that the two edges of the hexagon incident to v are simply oriented out of v follows from the fact that the three white vertices of the hexagon have outdegree 0. The second assertion is a consequence of the fact that the subgraph T consisting of the bi-oriented edges of D is a tree spanning all vertices of D except the three outer white vertices. Hence, there is a bi-oriented edge e incident to each black vertex v of the hexagon and this edge consitutes the third outgoing edge of v. □

Let D be a bicolored complete irreducible dissection. We define a mapping Ψ from the complete-tri-orientations without clockwise circuit of D to the tri-orientations of D, defined as follows. Let Y be a complete-tri-orientation of D without clockwise circuit. We proceed as follows to obtain $\Psi(Y)$: erase the orientation of the edges of the hexagon of D; then for each black vertex v of the hexagon, change the orientation of the unique outgoing inner half-edge h of v. According to Lemma 3.20, h belongs to a bi-oriented edge e, so that the change of orientation of h turns e into an edge simply oriented toward v. Thus, the obtained orientation $\Psi(Y)$ satisfies the conditions of a tri-orientation.

LEMMA 3.21. *Let D be a bicolored complete irreducible dissection. Let $\widetilde{Y}$ be a complete-tri-orientation of D without clockwise circuit. Then the tri-orientation $\Psi(\widetilde{Y})$ of D has no clockwise circuit.*

For each tri-orientation Y of D without clockwise circuit, there exists a complete-tri-orientation $\widetilde{Y}$ of D without clockwise circuit such that $\Psi(\widetilde{Y}) = Y$.

Proof. The first point is trivial. For the second point, we reason similarly as in Lemma 3.17. For each black vertex v of the hexagon of D, let $(h_1, \ldots, h_m)$ $(m \geq 3)$ be the sequence of half-edges of D incident to v in counter-clockwise order around v, with h_1 and h_2 belonging to the two edges e_1 and e_2 of the hexagon of D that are incident to v. To construct the preimage $\widetilde{Y}$ of Y, we orient the two half-edges of e_1 and e_2 toward their incident white vertex. The third outgoing half-edge is chosen to be h_3, which is the "leftmost" inner half-edge of v. An argument similar as in the proof of the second point of Lemma 3.17 ensures that this choice is judicious to avoid the creation of a clockwise circuit. An example of this construction is the transition between Figure 19(b) and Figure 19(c). □

Finally, Proposition 3.8 follows directly from Lemma 3.21 and Lemma 3.18.

Proposition 3.18 reduces the proof of Theorem 3.2 to the set of bicolored complete irreducible dissections. From now on, we will work with these dissections.

3.2. Transposition rules for orientations. Let D be a bicolored complete irreducible dissection and let G' be its derived map. We associate to a complete-tri-orientation of D an orientation of the edges of G' of D in the following way.

Let e be an edge of G' and let v be the extremity of e which is a primal or dual vertex (the other extremity is an edge-vertex). Let h be the half-edge of D following e in counterclockwise order around v. Then we give to e the orientation of h, see Figure 20.

LEMMA 3.22. *Let D be a bicolored complete irreducible dissection endowed with a complete-tri-orientation without clockwise circuit. Then the orientation of edges of the derived map G' of D obtained using the transposition rules has the following properties:*

- *Each primal or dual vertex of G' has outdegree 3.*
- *Each edge-vertex of G' has outdegree 1.*

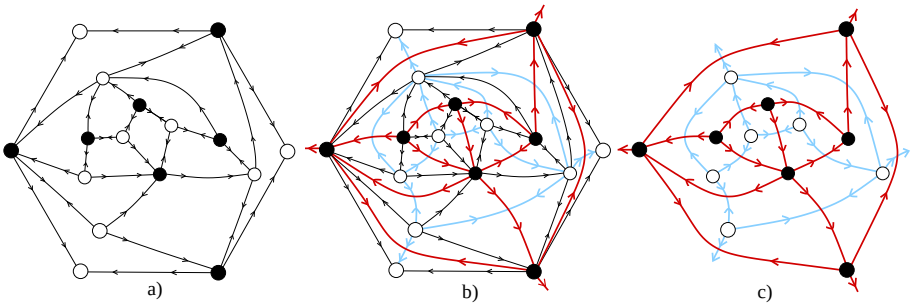


FIGURE 20. The construction of the derived map of a bicolored complete irreducible dissection. The dissection is endowed with a complete-tri-orientation without clockwise circuit, and the derived map is endowed with the orientation obtained from the transposition rules for orientations.

In other words, the orientation of edges of G' obtained by the transposition rules is an α_3-orientation.

Proof. The first point is trivial.

For the second point, let f be an inner face of D and v_f the associated edge-vertex of G' (we recall that v_f is the intersection of the two diagonals of f). The transposition rules for orientation ensures that the outdegree of v_f in G' is the number n_f of inward half-edges of D incident to f (i.e., having the interior of f on their right). Hence, to prove that each edge-vertex of G' has outdegree 1, we have to prove that $n_f = 1$ for each inner face f of D. Observe that n_f is a positive number, otherwise the contour of f would be a clockwise circuit in the complete-tri-orientation, which is impossible. Let n be the number of inner vertices of D. Using Euler's relation, it is easy to prove that D has $(n+2)$ inner faces and has $(4n+14)$ half-edges. By definition of a complete-tri-orientation, $3(n+3)$ half-edges are outgoing. Hence, $(n+5)$ half-edges are ingoing. Among these $(n+5)$ ingoing half-edges, exactly three are incident to the outer face (see Figure 19c). Hence, D has $(n+2)$ half-edges incident to an inner face, so that $\sum_f n_f = n+2$. As $\sum_f n_f$ is a sum of $(n+2)$ positive numbers adding to $(n+2)$, the pigeonhole's principle ensures that $n_f = 1$ for each inner face f of D. □

3.3. Uniqueness of a tri-orientation without clockwise circuit. The following lemma completes Lemma 3.22 and is crucial to establish the uniqueness of a tri-orientation without clockwise circuit for any irreducible dissection.

LEMMA 3.23. *Let D be a bicolored complete irreducible dissection endowed with a complete-tri-orientation Y without clockwise circuit. Let G' be the derived map of D. Then the α_3-orientation X of G' obtained from Y by the transposition rules has no clockwise circuit.*

Proof. Assume that X has a clockwise circuit $\mathcal{C}$. Each edge of G' connects an edge-vertex and a vertex of the original dissection D. Hence, the circuit $\mathcal{C}$ consists of a sequence of pairs $(\underline{e}, \overline{e})$ of consecutive edges of G' such that $\underline{e}$ goes from a vertex $\underline{v}$ of the dissection toward an edge-vertex v' of G' and $\overline{e}$ goes from v' toward a vertex $\overline{v}$ of the dissection. Let $(e'_1, \dots, e'_m)$ be the sequence of edges of G' between $\underline{e}$ and $\overline{e}$ in clockwise order around v', so that $e'_1 = \underline{e}$; and $e'_m = \overline{e}$ and let $(v_1, \dots, v_m)$ be their respective extremities, so that $v_1 = \underline{v}$ and $v_m = \overline{v}$. Note that $2 \leq m \leq 4$.

As each edge-vertex has outdegree 1 in X and as e'_m is going out of v', the edges $e'_1, \dots, e'_{m-1}$ are directed toward v'. Hence,the transposition rules for orientations ensure that the edges (v_i, v_{i+1}), for $1 \leq i \leq m-1$, are all bi-oriented or oriented from v_i to

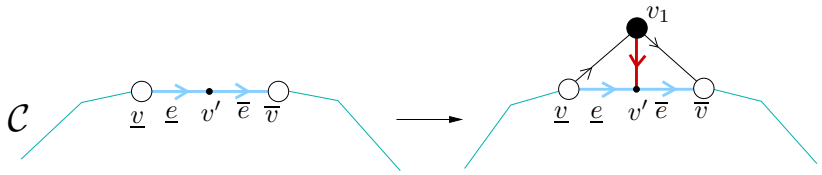


FIGURE 21. An oriented path of edges of the dissection can be associated to each pair $(\underline{e}, \overline{e})$ of consecutive edges of $\mathcal{C}$ sharing an edge-vertex.

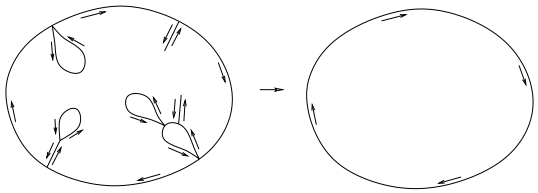


FIGURE 22. A simple clockwise circuit can be extracted from an oriented path enclosing a bounded simply connected region on its right.

v_{i+1} in the complete-tri-orientation Y of D. Hence,we can go from $\underline{v}$ to $\overline{v}$ passing by the exterior of $\mathcal{C}$ and using only edges of D, see Figure 21 for an example, where $m = 3$.

Concatenating the paths of edges of D associated to each pair $(\underline{e}, \overline{e})$ of $\mathcal{C}$, we obtain an oriented path of edges of D enclosing the interior of $\mathcal{C}$ on its right. Then, a simple clockwise circuit can be extracted from this closed path, see Figure 22. As the complete-tri-orientation Y has no clockwise circuit, this yields a contradiction. □

PROPOSITION 3.9. *Each irreducible dissection has at most one tri-orientation without clockwise circuit.*

Proof. Let D be a bicolored complete irreducible dissection and G' its derived map. A first important remark is that the transposition rules for orientations clearly define an injective mapping. In addition, Lemma 3.23 ensures that the image of a complete-tri-orientation of D without clockwise circuit is an α_3-orientation of G' without clockwise circuit. Hence, injectivity of the mapping and uniqueness of an α_3-orientation without clockwise circuit of G' ensure that D has at most one complete-tri-orientation without clockwise circuit. Finally, Proposition 3.8 ensures that each irreducible dissection has at most one tri-orientation without clockwise circuit. □

3.4. Existence of a tri-orientation without clockwise circuit.

Inverse of the transposition rules. Let D be a bicolored complete irreducible dissection and G' its derived map. Given an α_3-orientation of G', we can associate to this orientation an orientation of the half-edges of D by performing the inverse of the transposition rules: each half-edge h of D receives the orientation of the edge of G' that follows h in clockwise order around its origin, see Figure 20(b).

LEMMA 3.24. *Let D be an irreducible dissection and let G' be the derived map of D, endowed with its minimal α_3-orientation. Then the inverse of the transposition rules for orientations yields a complete-tri-orientation of D.*

Proof. The inverse of the transposition rules is clearly such that a vertex has the same outdegree in the obtained orientation of half-edges of D as in the α_3-orientation of G'.

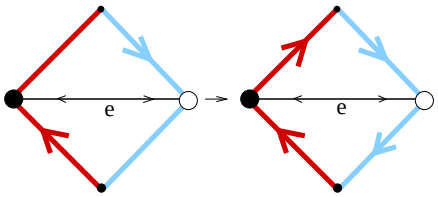


FIGURE 23. The case where the two half-edges of e are oriented inward implies that the contour of the associated face of G' is a clockwise circuit.

Hence, each inner vertex of D has outdegree 3 and each white (resp. black) vertex on the hexagon of D has outdegree 0 (resp. 3), see Figure 20(b).

To prove that the obtained orientation of D is a complete-tri-orientation, it remains to show that the two half-edges of an edge e of D can not both be oriented inward. Assume a contrario that there exists such an edge e. The transposition rules for orientation and the fact that each edge-vertex of G' has outdegree 1 imply that the contour of the face f_e of G' associated to e is a clockwise circuit, see Figure 23. This yields a contradiction with the minimality of the α_3-orientation. □

LEMMA 3.25. *Let D be a bicolored complete irreducible dissection and let G' be its derived map, endowed with its minimal α_3-orientation denoted by X. Let Y be the complete-tri-orientation obtained from X using the inverse of the transposition rules. Then Y has no clockwise circuit.*

Proof. Assume that Y has a clockwise circuit $\mathcal{C}$. For each vertex v on $\mathcal{C}$, we denote by h_v the half-edge of $\mathcal{C}$ incident to v and having the interior of $\mathcal{C}$ on its right, and we denote by e_v the edge of G' that follows h_v in clockwise order around v. As $\mathcal{C}$ is a clockwise circuit for Y, h_v is going out of v. Hence, by definition of the transposition rules, e_v is going out of v. Observe that, in the interior of $\mathcal{C}$, e_v is the most counter-clockwise edge of G' incident to v. From this observation, a clockwise circuit of X can be built in a similar way as in the proof of Lemma 3.12. This yields a contradiction. □

PROPOSITION 3.10. *For each irreducible dissection, there exists a tri-orientation without clockwise circuit.*

Proof. Lemma 3.25 ensures that each bicolored complete irreducible dissection D has a complete-tri-orientation Y without clockwise circuit. Then, Proposition 3.8 ensures that the existence of a complete-tri-orientation without clockwise circuit for any bicolored complete irreducible dissection implies the existence of a tri-orientation without clockwise circuit for any irreducible dissection. □

Finally, Theorem 3.2 follows from Proposition 3.9 and Proposition 3.10.

Conclusion. The concept of minimal α-orientation yields a general method to formulate bijections between families of rooted planar maps and families of rooted trees. Four already known bijections are recovered in this unified framework: with the families of (rooted) tetravalent maps, quadrangulations, triangulated maps, and triangulations.

Concerning the families of 3-connected maps and irreducible triangulations of the 4-gon, the method based on minimal α-orientations does not seem to apply easily. Nevertheless, there also exist bijections related to these two families, with the particular feature that no root is needed (the bijections are between unrooted plane trees and unrooted planar maps). The advantage is that these new bijections yield enumeration fomulas for families of unrooted planar maps. In particular, we obtain the first combinatorial proof for the number of unrooted plane triangulations.

CHAPTER 4

Algorithmic applications

t **Introduction.** As demonstrated in the first three chapters, planar structures (maps, trees) have rich combinatorial properties. These properties are both structural (existence of specific orientations and coloration of edges) and enumerative (existence of bijections). The goal is now to exploit the combinatorial study in order to develop an efficient algorithmics on planar structures. In this chapter, we show that the bijections presented in Chapter 3 give rise to efficient algorithms for random generation and encoding. A striking further application is the development of a random generator for planar graphs with very low polynomial complexity (linear for approximate-size sampling and quadratic for exact-size sampling).

Results obtained in this chapter. The first application we develop is purely enumerative. Recall that we have counted bijectively the family of irreducible dissections of a hexagon and the family of irreducible triangulations of a 4-gon in Chapter 3. In Section 1.1, we show how to count rooted 3-connected maps (Prop. 4.1 p. 124), starting from the enumeration formulas for irreducible dissections. A similar method is applied in Section 1.2 to count rooted 4-connected triangulations (Prop. 4.2 p. 125), starting from irreducible triangulations. In each case, the result is formulated using generating functions; a rational expression in terms of the generating function of a family of trees is obtained. In this way, we recover in a more direct way enumerative results found by Mullin and Schellenberg [**90**] for 3-connected maps and Tutte [**111**] for 4-connected triangulations.

We then describe algorithmic applications. From the bijections we derive encoding procedures that are optimal (in a sense to be made precise) for the families of 3-connected maps (Theo 4.1 p. 127) and 4-connected triangulations (Theo. 4.2 p. 128). The principle consists in encoding the map by the parenthesis (contour) word of the associated tree. Then we focus on random generation, describing how the bijections give rise to efficient generators for rooted 3-connected maps (Prop. 4.3 and 4.4 p. 129) and rooted 4-connected triangulations. Notice that the derivation of map generators from bijections has already been extensively studied by Schaeffer [**99, 100**]. Our most original and important contribution in this chapter is the development of a very efficient uniform random generator for labeled planar *graphs* (in contrast to a map, no explicit embedding is attached to a planar graph). The sampler makes use of a well-known decomposition of planar graphs into 3-connected planar graphs. As 3-connected planar graphs have a unique embedding, the 3-connected components can be drawn using our bijective generator of 3-connected maps. Then these components are assembled into a planar graph using specific branching probabilities, which are computed based on the framework of Boltzmann samplers, as introduced by Duchon, Flajolet, Louchard, and Schaeffer [**43**]. We only provide the description of the algorithms and of its main ingredients, as a

striking application of the combinatorial study of planar maps. The full paper to be submitted [**62**] contains all the proofs.

Motivations. *Random generation.* The bijections presented in Chapter 3 yield uniform random generators for the corresponding families of maps, each time with linear-time complexity. Random generation is an important tool to test conjectures, and to test the average complexity and correctness of algorithms. The uniform distribution is a natural one, as it does not give advantage to any particular object, i.e., it is maximally spread over the set to be sampled. In addition, random generation allows us to observe statistical properties of objects drawn uniformly at random. Typically, asymptotic properties are observed: limit shapes, limit distribution of parameters... As these properties are asymptotic, it is necessary to draw very large objects. For planar maps, a size of order 10^4 is sufficient to observe the asymptotic behaviour in most cases. This is readily achieved by the bijective samplers presented in this chapter, which have linear time complexity.

Let us mention examples where random generation has been a fruitful source of new results on planar maps. In his PhD [**99**], G. Schaeffer observed a distribution in the scale of $n^{1/4}$ for the radius of a random quadrangulation with n faces (the radius of a rooted map is the distance from the root vertex to the farthest vertex); and conjectured a limit distribution, in the scale $n^{2/3}$, for the size of the so-called *core-map* extracted from a random map of size n. These two (independent) observations have been the starting point of fruitful research: in [**30**] it is shown that the radius of a random quadrangulations with n vertices rescaled by $n^{-1/4}$ converges to a so-called Brownian snake; in [**1**] it is shown that the size of the core-map in a random map with n edges fluctuate in the scale of $n^{2/3}$ around its mean according to a so-called Airy distribution. In a similar way, the uniform sampler for irreducible triangulations, obtained in Section 3.2, has made it possible to perform simulations of the compact straight-line drawing algorithm presented in Chapter 5. I have observed that the grid size is always of the form $\alpha n \times \alpha n$, with $\alpha \approx 0.4$ (see Figure 8 page 166). This has encouraged me to perform an analysis (also given in Chapter 5), which ensures that the grid is asymptotically with high probability of size $11n/27 \times 11n/27$ up to fluctuations of order $\sqrt{n}$.

Optimal encoding. A second important application of our bijections is an asymptotically optimal encoding algorithm for the corresponding families of maps. Let us make precise what we mean by *optimal encoder*. A *fixed size* coding algorithm for a set $\mathcal{E}$ is an injective mapping from $\mathcal{E}$ to the set $\{0,1\}^k$ of binary words of length k. The integer k is called the *length* of the code. By injectivity of the mapping, $2^k \geq |\mathcal{E}|$, so that the best compression rate one can expect for the code is with a length $k = \lceil \log_2(|\mathcal{E}|) \rceil$. An encoder with length $\lceil \log_2 |\mathcal{E}| \rceil$ is said to be *optimal*. The concept of encoding is then extended to a combinatorial class $\mathcal{C} = \cup_n \mathcal{C}_n$ graduated by a size parameter n. A *size-uniform* encoder for $\mathcal{C}$ is a procedure that maps bijectively objects of $\mathcal{C}$ to binary words, such that, for $n \geq 0$, the words encoding objects of $\mathcal{C}_n$ have all a same length k_n. The encoding is said to be *linear* if the encoding and decoding procedures have linear-time complexity. Moreover, the encoding procedure is said to be *asymptotically optimal* iff $k_n \sim \lceil \log_2 |\mathcal{C}_n| \rceil$ as $n \to \infty$. Clearly, such encoders give asymptotically the best compression rates for size-uniform encoders. The encoders presented in this chapter, which are derived from explict bijections, are all linear asymptotically optimal encoders. In particular, we obtain the first optimal encoder for the family of 3-connected maps. This family

is of great interest; indeed the incidence structures of meshes with polygonal faces and a spherical topology are exactly 3-connected maps. Hence, our encoder settles a question raised in Computational Geometry, whether there exists an encoder for polygonal meshes achieving the lower bound of 2 bits per edge. This question is important in theory and in practice. Indeed, there exist efficient (not size-uniform) heuristics to encode the combinatorial part of meshes [**70, 79**]. In case of bad compression rate by these heuristics on particular instances, our encoder can be called alternatively, producing invariably a coding word with 2 bits per edge.

1. Counting planar maps

This section explains how 3-connected maps can be counted, starting from the enumeration of irreducible dissections; in the same way 4-connected triangulations are counted starting from the enumeration of irreducible triangulations. The simple expressions we get are in terms of generating functions. The corresponding formulas for coefficients (not given in this thesis) are not closed, involving summations of binomial coefficients. (These formulas are easily extracted using Lagrange inversion formula.)

1.1. Counting rooted 3-connected maps. The idea to count rooted 3-connected maps is the following: 1) rooted irreducible dissections of the hexagon have an explicit counting formula given in Corollary 3.1 page 99, 2) 3-connected maps and irreducible dissections are closely related via a simple decomposition.

1.1.1. *Generating functions of rooted dissections.* Even if the counting formulas obtained in Corollary 3.1 are simple, it proves useful to have an expression of the corresponding generating functions. Indeed, the decomposition-method we develop is suitably handled by generating functions. Recall that $\mathcal{B}^{\bullet}_{ij}$ ($\mathcal{B}^{\circ}_{ij}$) stands for the set of rooted bicolored binary trees such that the root node is black (white, respectively).

Let $r_1(x_\bullet, x_\circ) := \sum |\mathcal{B}^{\bullet}_{ij}| x_\bullet^i x_\circ^j$ and $r_2(x_\bullet, x_\circ) := \sum |\mathcal{B}^{\circ}_{ij}| x_\bullet^i x_\circ^j$ be the series of black-rooted and white-rooted bicolored binary trees. Observe that $\mathcal{B}^{\bullet}_{ij} \simeq \mathcal{B}^{\circ}_{ji}$ (upon switching colors), so that $r_1(x_\bullet, x_\circ) = r_2(x_\circ, x_\bullet)$.

By decomposition at the root, $r_1(x_\bullet, x_\circ)$ and $r_2(x_\bullet, x_\circ)$ are the solutions of the system:

$$\left\{ \begin{array}{rcl} r_1(x_\bullet, x_\circ) & = & x_\bullet \left(1 + r_2(x_\bullet, x_\circ)\right)^2 \\ r_2(x_\bullet, x_\circ) & = & x_\circ \left(1 + r_1(x_\bullet, x_\circ)\right)^2 \end{array} \right. . \tag{13}$$

Define an *edge-marked bicolored binary tree* as a bicolored binary tree with a marked inner edge. Let $\bar{\mathcal{B}}_{ij}$ be the set of edge-marked bicolored binary trees with i black nodes and j white nodes. Cutting the marked edge of such a tree yields a pair made of a black-rooted and a white-rooted binary tree. As a consequence, the generating function counting edge-marked bicolored binary trees is $r_1 \cdot r_2$, i.e., $r_1 \cdot r_2 = \sum_{ij} |\bar{\mathcal{B}}_{ij}| x_\bullet^i x_\circ^j$.

Let us consider bi-rooted objects as in the proof of Theorem 3.4; since any object of $\mathcal{B}_{ij}$ has $(2i-j+1)$ white leaves (connected to a black node) and $(2j-i+1)$ black leaves (connected to a white node),

$$|\mathcal{B}^{\circ}_{ij}| = \frac{2j-i+1}{2i-j+1} |\mathcal{B}^{\bullet}_{ij}|.$$

Similarly, counting in two ways the objects of $\mathcal{B}^{\bullet}_{ij}$ having a marked edge yields

$$|\bar{\mathcal{B}}_{ij}| = \frac{i+j-1}{2i-j+1}|\mathcal{B}^{\bullet}_{ij}|.$$

Recall that the number $|\mathcal{D}'_{ij}|$ of rooted bicolored irreducible dissections with i inner black vertices and j inner white vertices satisfies $|\mathcal{D}'_{ij}| = \frac{3}{2i-j+1}|\mathcal{B}^{\bullet}_{ij}|$, according to (11) page 99. Thus, we have $|\mathcal{B}^{\bullet}_{ij}|+|\mathcal{B}^{\circ}_{ij}|-|\bar{\mathcal{B}}_{ij}| = \frac{3}{2i-j+1}|\mathcal{B}^{\bullet}_{ij}| = |\mathcal{D}'_{ij}|$ (using (11)), so that

$$\sum_{i,j} |\mathcal{D}'_{ij}| x_{\bullet}^i x_{\circ}^j \;=\; r_1(x_{\bullet}, x_{\circ}) + r_2(x_{\bullet}, x_{\circ}) - r_1(x_{\bullet}, x_{\circ}) r_2(x_{\bullet}, x_{\circ}). \tag{14}$$

Substituting $x_{\bullet}$ and $x_{\circ}$ by x, we obtain

$$D(x) := \sum_n |\mathcal{D}'_n| x^n = 2r(x) - r(x)^2, \tag{15}$$

where $|\mathcal{D}'_n|$ is the number of rooted irreducible dissections with n inner vertices and $r(x) = x\,(1 + r(x))^2$ is the generating function of binary trees according to the number of nodes.

1.1.2. *Generating function of rooted 3-connected maps.* We now present a simple decomposition that relates rooted 3-connected maps and rooted irreducible dissections. The idea is to add an outer edge to a rooted irreducible dissection and observe that the separating 4-cycles that might appear are nested. To our knowledge, this decompositon has not been described before. In our case, it makes it possible to derive an expression for the generating function of rooted 3-connected maps from the generating function of rooted irreducible dissections. The expressions we get are equivalent to those obtained by Mullin and Schellenberg using algebraic methods [**90**].

Injection from $\mathcal{Q}'$ into $\mathcal{D}'$. Recall that $\mathcal{Q}'$ stands for the set of rooted irreducible quadrangulations, counted either w.r.t. the number n of vertices or the numbers (i, j) of black and white vertices; and $\mathcal{D}'$ stands for the set of rooted irreducible dissections, counted either w.r.t. the number n of inner vertices or the numbers (i, j) of inner black and inner white vertices. Let us consider the mapping ι defined on rooted quadrangulations by the removal of the root-edge and rerooting on the next edge in counterclockwise order around the root-vertex; ι is clearly injective, and for any quadrangulation Q, $\iota(Q)$ has only quadrangular faces but the outer one, which is hexagonal, and it can not have more separating 4-cycles than Q. Hence the restriction of ι to $\mathcal{Q}'$ is an injection from $\mathcal{Q}'$ to $\mathcal{D}'$, more precisely from $\mathcal{Q}'_n$ to $\mathcal{D}'_{n-4}$ and from $\mathcal{Q}'_{ij}$ to $\mathcal{D}'_{i-3,j-3}$.

It is however not a bijection, since the inverse edge-adding operation π, performed on an irreducible dissection, can create a separating 4-cycle on the obtained quadrangulation. Precisely, given D a rooted irreducible dissection —with root-vertex s and t the vertex of the hexagon opposite to s — a path of length 3 between s and t is called a *decomposition path*. The two paths of edges of the hexagon connecting s to t are called *outer decomposition paths*, and the other ones, if any, are called *inner decomposition paths* of D.

Observe that inner decomposition paths of D are in one-to-one correspondence with separating 4-cycles of the quadrangulation $\pi(D)$.

A rooted irreducible dissection without inner decomposition path is said to be *undecomposable*. The discussion on decomposition paths yields the following result,

where $\mathcal{P}'_n$ is the set of rooted 3-connected maps with n edges and $\mathcal{P}'_{ij}$ is the set of rooted 3-connected maps with i vertices and j faces.

LEMMA 4.1. *Denote by $\mathcal{U}'_n$ the set of rooted undecomposable dissections with n inner vertices and by $\mathcal{U}'_{ij}$ the set of rooted undecomposable dissections with i inner black vertices and j inner white vertices. Then $\mathcal{U}'_{n-4}$ is in bijection with $\mathcal{P}'_n$ and $\mathcal{U}'_{i-3,j-3}$ is in bijection with $\mathcal{P}'_{ij}$.*

Proof. The image of a rooted irreducible quadrangulation by ι is a rooted dissection such that the inverse edge-adding operation π does not create a separating 4-cycle, i.e., an undecomposable dissection. Moreover, Euler's relation ensures that the image of a quadrangulation with n faces has $n-4$ inner vertices. As ι is injective, it is bijective to its image, i.e., it is a bijection between $\mathcal{Q}'_n$ and $\mathcal{U}'_{n-4}$; and a bijection between $\mathcal{Q}'_{ij}$ and $\mathcal{U}'_{i-3,j-3}$. The result follows, as $\mathcal{Q}'_n$ and $\mathcal{Q}'_{ij}$ are respectively in bijection with $\mathcal{P}'_n$ and $\mathcal{P}'_{ij}$ via the angular mapping (Theorem 1.2). □

Thanks to Lemma 4.1, enumerating rooted 3-connected maps reduces to enumerating rooted undecomposable dissections.

Decomposition of rooted irreducible dissections. Since irreducible dissections do not have multiple edges nor cycles of odd length, decomposition paths satisfy the following properties:

LEMMA 4.2. *Let $D \in \mathcal{D}'$, and let $\mathcal{P}_1$ and $\mathcal{P}_2$ be two different decomposition paths of D. Then:*

- *either $\mathcal{P}_1 \cap \mathcal{P}_2 = \{s,t\}$, in which case $\mathcal{P}_1$ and $\mathcal{P}_2$ are said to be* internally disjoint*;*
- *or there exists one inner vertex v adjacent to s or t such that $\mathcal{P}_1 \cap \mathcal{P}_2 = \{s\} \cup \{t\} \cup \{v\}$, in which case $\mathcal{P}_1$ and $\mathcal{P}_2$ are said to be* upper *or* lower joint *depending on whether v is adjacent to s or t.*

Lemma 4.2 implies in particular that two decomposition paths can not cross each other. Hence the decomposition paths of an irreducible dissection D follow a left-to-right order, from the outer decomposition path containing the root (called left outer path) to the other outer decomposition path (called right outer path).

LEMMA 4.3. *Let $D \in \mathcal{D}'$, and let $\mathcal{P}_1$ and $\mathcal{P}_2$ be two upper joint (resp. lower joint) decomposition paths of D. Then the interior of the area delimited by $\mathcal{P}_1$ and $\mathcal{P}_2$ is a face incident to t (resp. to s).*

Proof. This is a trivial consequence of the fact that the interior of each 4-cycle of D is a face. □

Decomposition word of an irreducible dissection. Let $D \in \mathcal{D}'$ and let $\{\mathcal{P}_0, \dots, \mathcal{P}_\ell\}$ be the sequence of decomposition paths of D ordered from left to right. Let us consider the alphabet $\mathcal{A} = \{s\} \cup \{t\} \cup \mathcal{U}'$, where $\mathcal{U}'$ is the set of all rooted undecomposable dissections; the *decomposition word* of D is the word $w = w_1 \dots w_\ell$ of length ℓ on $\mathcal{A}$ such that, for any $1 \leq i \leq \ell$: if $\mathcal{P}_{i-1}$ and $\mathcal{P}_i$ are upper joint, then $w_i = s$; if $\mathcal{P}_{i-1}$ and $\mathcal{P}_i$ are lower joint, then $w_i = t$; if $\mathcal{P}_{i-1}$ and $\mathcal{P}_i$ are internally disjoint, then $w_i = U$, where U is the undecomposable dissection delimited by $\mathcal{P}_{i-1}$ and $\mathcal{P}_i$ (rooted at the first edge of $\mathcal{P}_{i-1}$ and with s as root-vertex), see Figure 1. This encoding is injective, an easy consequence of Lemma 4.3.

Characterization of decomposition words of elements of $\mathcal{D}'$. The fact that D has no separating 4-cycle implies easily that its decomposition word has no factor ss nor tt, and these are the only forbidden factors. Moreover, as a dissection has at

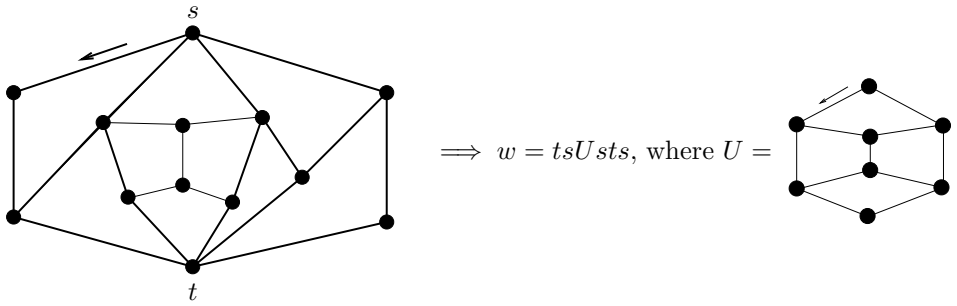


FIGURE 1. Example of the decomposition of a rooted irreducible dissection and of its associated decomposition word.

least one inner vertex, a decomposition word can be neither the empty word, nor the one-letter words s and t, nor the two-letter words st and ts. It is easily seen that all other words encode irreducible dissections of the hexagon.

This leads to the following equation linking the generating functions $D(x)$ and $U(x)$ counting $\mathcal{D}'$ and $\mathcal{U}'$ according to the number of inner vertices,

$$x^2D(x) + 2x^2 + 2x + 1 \;=\; \left(1 + \frac{2x}{1-x}\right) \cdot \frac{1}{1 - x^2U(x)\left(1 + \frac{2x}{1-x}\right)}. \tag{16}$$

Similarly, let $D(x_\bullet, x_\circ) := \sum |\mathcal{D}'_{ij}| x_\bullet^i x_\circ^j$ and $U(x_\bullet, x_\circ) := \sum |\mathcal{U}'_{ij}| x_\bullet^i x_\circ^j$. Then the characterisation of the coding words gives

$$\begin{aligned} &x_\bullet x_\circ D(x_\bullet, x_\circ) + 2x_\bullet x_\circ + x_\bullet + x_\circ + 1 \\ &\quad = (1 + x_\bullet) \cdot \frac{1}{1 - x_\circ x_\bullet} \cdot (1 + x_\circ) \;\cdot\; \frac{1}{1 - x_\bullet x_\circ U(x_\bullet, x_\circ)(1 + x_\bullet)\frac{1}{1 - x_\circ x_\bullet}(1 + x_\circ)}. \end{aligned} \tag{17}$$

PROPOSITION 4.1 (counting rooted 3-connected maps). *Let $\mathcal{P}'_n$ be the number of rooted 3-connected maps with n edges and $\mathcal{P}'_{i,j}$ the number of rooted 3-connected maps with i vertices and j faces. Then*

$$\sum_n |\mathcal{P}'_{n+2}| x^n \;=\; \frac{1-x}{1+x} \;-\; \frac{1}{1 + 2x + 2x^2 + x^2(2r(x) - r(x)^2)},$$

where $r(x) = x\,(1 + r(x))^2$, and

$$\sum_{i,j} |\mathcal{P}'_{i+2,j+2}| x_\bullet^i x_\circ^j \;=\; \frac{1 - x_\bullet x_\circ}{(1 + x_\bullet)(1 + x_\circ)} - \frac{1}{1 + x_\bullet + x_\circ + 2x_\bullet x_\circ + x_\bullet x_\circ(r_1 + r_2 - r_1 r_2)},$$

$$\text{where } \begin{cases} r_1(x_\bullet, x_\circ) &= x_\bullet\,(1 + r_2(x_\bullet, x_\circ))^2 \\ r_2(x_\bullet, x_\circ) &= x_\circ\,(1 + r_1(x_\bullet, x_\circ))^2 \end{cases}.$$

Proof. Lemma 4.1 ensures that $\sum_n |\mathcal{P}'_{n+2}| x^n = x^2U(x)$ and $\sum_{i,j} |\mathcal{P}'_{i+2,j+2}| x_\bullet^i x_\circ^j = x_\bullet x_\circ U(x_\bullet, x_\circ)$. Then, Equation (16) yields an expression of $x^2U(x)$ in terms of $D(x)$ and Equation (17) yields an expression of $x_\bullet x_\circ U(x_\bullet, x_\circ)$ in terms of $D(x_\bullet, x_\circ)$. In these expressions, replace $D(x)$ and $D(x_\bullet, x_\circ)$ by their respective expression in terms of r and in terms of r_1 and r_2, as given by Equations (14) and (15). □

1.2. Counting rooted 4-connected triangulations. Counting method. We make use of the following well known characterisation "a triangulation is 4-connected iff the interior of every 3-cycle is a face, except for the outer 3-cycle, and there are at least 4 vertices". Clearly this is very close to the definition of irreducible triangulations; and, as we show next, an expression for the generating function of 4-connected triangulations can be derived from the coefficients counting irreducible triangulations w.r.t. the number of vertices. In all this section, the set of rooted 4-connected triangulations with n inner vertices is denoted by $\mathcal{C}'_n$ and the set of rooted irreducible triangulations with n inner vertices is denoted by $\mathcal{T}'_n$. Coefficient extraction in the generating function makes it possible to recover the expression for $\mathcal{T}'_n$ found by Tutte [**111**] using algebraic methods.

Decomposing an irreducible triangulation. For $n \geq 5$, the operation of removing the root edge of an object of $\mathcal{C}'_n$ and carrying the root on the ccw-consecutive edge is an injective mapping from $\mathcal{C}'_n$ to $\mathcal{T}'_{n-1}$. However, given $T \in \mathcal{T}'_{n-1}$, the inverse edge-adding operation can create a separating 3-cycle if there exists an internal path of length 2 connecting the origin of the root of T to the vertex diametrically opposed in the outer quadrangle of T. Objects of $\mathcal{T}'_{n-1}$ having no such internal path are said to be *undecomposable*; the corresponding family counted with respect to the number n of inner vertices is denoted by $\mathcal{U}' = \cup_n \mathcal{U}'_n$. The above discussion ensures that $\mathcal{C}'_n$ is in bijection with $\mathcal{U}'_{n-1}$ for $n \geq 5$. In addition, a maximal decomposition of an object of $\mathcal{T}'$ along the above mentioned interior paths of length 2 ensures that such an object is a sequence of objects of $\mathcal{U}'$. Precisely, the graph enclosed between two consecutive paths of length 2 is either an undecomposable triangulation of the 4-gon or a quadrangle with a unique interior edge connecting the middles of the two paths. This leads to the equation

$$T(z) + 1 = \frac{U(z) + 1}{1 - z(U(z) + 1)}, \tag{18}$$

where $T(z) = \sum |\mathcal{T}'_n| z^n$ and $U(z) = \sum |\mathcal{U}'_n| z^n$ are respectively the series counting the set $\mathcal{T}'$ and the set $\mathcal{U}'$ with respect to the number of inner vertices.

PROPOSITION 4.2 (counting 4-connected triangulations). *The series $C(z)$ counting rooted 4-connected triangulations by their number of inner vertices has the following expression,*

$$C(z) = \frac{z(A(z) - A(z)^2 + 1)}{1 + z(A(z) - A(z)^2 + 1)}, \tag{19}$$

where $A(z) = z(1+A(z))^3$ is the series counting rooted ternary trees by their number of nodes.

Proof. As $\mathcal{C}'_n$ is in bijection with $\mathcal{U}'_{n-1}$ for $n \geq 5$ and as the unique 4-connected triangulation with less than 5 vertices is the tetrahedron, we have $C(z) = z(U(z) + 1)$. Hence, Equation (18) yields $C(z) = z(T(z) + 1)/(1 + z(T(z) + 1))$. Thus, it remains to provide an expression of $T(z)$ in terms of the series $A(z) = \sum A_n z^n$ counting rooted ternary trees by their number of nodes. We define respectively the sets $\overline{\mathcal{A}_n}$ and $\widehat{\mathcal{A}_n}$ of ternary trees with n nodes and having the following marks: an inner edge is marked and oriented for objects of $\overline{\mathcal{A}}_n$; an inner edge is marked and oriented and a leaf is marked for objects of $\widehat{\mathcal{A}}_n$. As a ternary tree with n nodes has $n-1$ inner edges and $2n+2$ leaves, we have $A_n \cdot 2(n-1) = |\widehat{\mathcal{A}_n}| = |\overline{\mathcal{A}_n}|(2n+2)$, so that $|\overline{\mathcal{A}_n}| = 2\frac{n-1}{2n+2} A_n$. In addition, given a ternary tree with

a marked oriented edge, the operation of cutting the marked edge produces an ordered pair of rooted ternary trees. Hence, the series counting $\overline{\mathcal{A}} := \cup_n \overline{\mathcal{A}}_n$ with respect to the number of nodes is $A(z)^2$. Finally, we know from Corollary 3.7 that $T_n = \frac{4}{2n+2}A_n$. Hence, we also have $|\mathcal{T}'_n| = \frac{2n+2}{2n+2}A_n - 2\frac{n-1}{2n+2}A_n = A_n - |\overline{\mathcal{A}_n}|$, from which we conclude that $T(z) = A(z) - A(z)^2$. Finally, to obtain the expression of $C(z)$ in terms of $A(z)$, we substitute $T(z)$ by $A(z) - A(z)^2$ in the expression $C(z) = z(T(z)+1)/(1+z(T(z)+1))$. □

From Expression (19), the coefficients of $C(z)$ can be quickly extracted: $C(z) = z + z^3 + 3\,z^4 + 12\,z^5 + 52\,z^6 + 241\,z^7 + \mathcal{O}(z^8)$. The first coefficients match with the values given by the formula $c_n = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n}(-1)^i\binom{3n-i-1}{n-i}\binom{2i}{2}$ for the number of rooted 4-connected triangulations with n vertices. This formula was found by Tutte [**111**] using more complicated algebraic methods.

2. Coding planar maps

Two encoding algorithms are presented in this section, related to the families of 3-connected maps and 4-connected triangulations. The key ingredients are the bijection between binary trees and irreducible dissections of the hexagon to encode 3-connected maps and the bijection between ternary trees and irreducible triangulations to encode 4-connected triangulations. These bijections reduce the task of coding a map to the much easier task of coding a tree.

2.1. Coding 3-connected maps. This section introduces an algorithm to encode a 3-connected map. Precisely, the algorithm encodes an outer-triangular 3-connected map, but it is then easily extended to encode any 3-connected map. Indeed, if the outer face of G is not triangular, fix three consecutive vertices v, v' and v'' incident to the outer face of G and link v and v'' by an edge to obtain an outer-triangular 3-connected planar map $\widetilde{G}$. Then, the coding of G is obtained as the coding of $\widetilde{G}$ plus one bit indicating if an edge-addition has been done.

Let G be an outer-triangular 3-connected map and let G' be its derived map. The encoding of G is done according to the following steps, illustrated in Figure 2.

1) Compute the minimal α_3-orientation of the derived map G'. The orientation is computed using the algorithm presented in Section 4.

2) Compute the irreducible dissection D associated to G. We associate with G the unique bicolored dissection D of the hexagon such that each edge of G connects the two (opposite) black vertices of a unique inner face of D. The fact that G is 3-connected easily implies that the dissection D is irreducible. Notice that D has n inner faces if G has n edges. Hence, according to Euler's relation, D has $n-2$ inner vertices. Similarly, if G has i vertices and j inner faces, then D has i black vertices and $j+3$ white vertices.

3) Compute the tri-orientation of D without clockwise circuit. Inner half-edges of D are oriented in the following way: an inner half-edge h is oriented inward if its origin belongs to the hexagon; otherwise, h receives the orientation of the edge of G' following h in clockwise order around its origin. As shown in Section 3 (precisely, Lemma 3.25 combined with the correspondence of Figure 19), the obtained orientation of the half-edges of D is the unique tri-orientation without clockwise circuit.

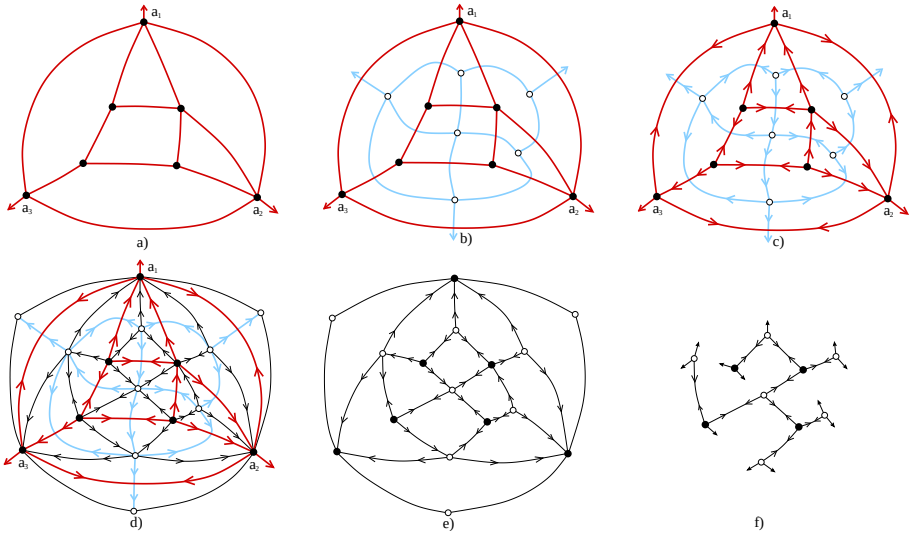


FIGURE 2. Encoding a 3-connected map.

4) Open the dissection D into a binary tree T. Once the tri-orientation without clockwise circuit is computed, D is opened into a binary tree T, by keeping the half-edges oriented outward (see the description of the opening mapping in Section 2.1.2).

5) Encode T. First, choose an arbitrary leaf of T, root T at this leaf, and encode the obtained rooted binary tree using a parenthesis word (also called *Dyck word*). The opening of a 3-connected map with n edges is a binary tree with $n-2$ nodes, yielding an encoding Dyck word of length $2(n-2)$.

Similarly, the opening of a 3-connected map with i vertices and j inner faces is a black-rooted bicolored binary tree with $i-3$ black nodes and j white nodes. A black-rooted bicolored binary tree with a given number of black and white nodes is encoded by a pair of parenthesis words, as explained in Section 2.1.5. Then the two words can be asymptotically optimally encoded in linear time, according to [**17**, Lem.7].

THEOREM 4.1 (coding 3-connected maps). *The coding algorithm has linear-time complexity and is asymptotically optimal: the number of bits per edge of the code of an element of $\mathcal{P}'_n$ ($\mathcal{P}'_{ij}$) is asymptotically equal to the binary entropy per edge, defined as $\frac{1}{n}\log_2(|\mathcal{P}'_n|)$ ($\frac{1}{i+j-2}\log_2(|\mathcal{P}'_{ij}|)$, respectively).*

Proof. It is clear that the encoding algorithm has linear-time complexity, as the algorithm computing the minimal α_3-orientation of the derived map has linear-time complexity.

According to Corollary 3.1, Proposition 4.3 and 4.4, $|\mathcal{B}'_n|/|\mathcal{P}'_n|$ and $|\mathcal{B}^\bullet_{ij}|/|\mathcal{P}'_{ij}|$ are bounded by fixed polynomials. Hence, the entropy per edge of $\mathcal{B}'_n$ and $\mathcal{P}'_n$ are asymptotically equal, and the binary entropy per edge of $\mathcal{B}^\bullet_{ij}$ and $\mathcal{P}'_{ij}$ are asymptotically equal. As the encoding of objects of $\mathcal{B}'_n$ ($\mathcal{B}^\bullet_{ij}$) using parenthesis words (and further compression for $\mathcal{B}^\bullet_{ij}$) is asymptotically optimal, the encoding of objects of $\mathcal{P}'_n$ ($\mathcal{P}'_{ij}$, respectively) is also asymptotically optimal. □

2.2. Coding 4-connected triangulations. We briefly describe here the encoding algorithm for 4-connected triangulations. The idea is to use the bijection between ternary trees and irreducible triangulations. Let T be a rooted 4-connected triangulation with $(n+4)$ vertices. The encoding of T consists of the following steps.

1) Delete the root edge of T. In this way, T becomes an irreducible triangulation with n inner vertices.

2) Open T into a ternary tree τ. We recall that the steps of the opening are 1) the computation of the minimal transversal structure (in linear time) using the algorithm of Section 5, 2) the deletion of the outer half-edges and of the half-edges having the same color as their clockwise-consecutive half-edge.

3) Encode τ. The tree τ, which has n nodes, is first rooted at an arbitrary leaf. Then the obtained rooted tree is classically encoded using a ccw traversal starting at the root, writing a letter 1 when a node is discovered and a letter 0 when a leaf is traversed. The obtained string has n letters 1 and $2n+1$ letters 0. Using [**17**, Lem. 7], this string can be further encoded as a binary string whose length l_n matches asymptotically the binary entropy $\log_2((3n+1)!/(n!(2n+1)!))$, i.e., $l_n \sim n\log_2(27/4)$ (using Stirling's formula). In addition, this mapping has linear time complexity and the ternary tree can be recovered from the string in linear time.

THEOREM 4.2 (coding 4-connected triangulations). *The encoding algorithm for 4-connected triangulations is linear and asymptotically optimal.*

Proof. If the 4-connected triangulation to be encoded has n vertices, then the triangulation obtained by removing the root edge has $n-1$ inner vertices. Hence the ternary tree τ obtained by doing the opening has $n-1$ nodes. As discussed above, the length l_n of the binary string coding a 4-connected triangulation with n vertices satisfies $l_n \sim n\log_2(27/4)$. Moreover, Tutte has shown that $|\mathcal{C}'_n| \sim 3^3/2^9\sqrt{3/\pi}n^{-5/2}(27/4)^{n-3}$, implying directly $\log_2(|\mathcal{C}'_n|) \sim n\log_2(27/4)$. Hence, $l_n \sim \log_2(|\mathcal{C}'_n|)$. □

3. Random sampling of planar maps

As a second important algorithmic application, the bijections presented in this section yield very efficient (linear) random generators for the corresponding families of maps. As map generators derived from bijections are already studied by Schaeffer in much details [**99, 100**], we only discuss these applications for the bijections with binary trees and ternary trees. In the first case, generating a binary tree and applying the closure mapping yields a random generator for rooted 3-connected maps; in the second case, generating a ternary tree and applying the closure mapping yields a random generator for rooted 4-connected triangulations. Compared to direct bijections as those presented in Section 1, a rejection step appears in the derived random generators, so that the complexity analysis requires to study the rejection probability.

3.1. Random sampling of rooted 3-connected maps. The method to perform random sampling of rooted 3-connected maps is the following. Rooted binary trees are in bijection with irreducible dissections of the hexagon, with simple correspondence between the size parameters. Hence, uniform sampling of binary trees combined with the bijection yields a uniform sampler for rooted irreducible

dissections. Moreover, via the angular mapping, rooted 3-connected maps are a subset of rooted irreducible dissections. Hence, running the sampler for rooted dissections until a rooted 3-connected map is output yields a uniform sampler for rooted 3-connected maps. The rejection probability is not large, as the number of rooted 3-connected maps with a given size is of the same order as the number of rooted irreducible dissections with the same size. This yields random generators for rooted 3-connected maps with expected linear time complexity.

3.1.1. *Sampling rooted 3-connected maps with n edges.* Theorem 3.4 ensures that the following algorithm samples rooted 3-connected maps with n edges uniformly at random.

(1) Sample an object $T \in \mathcal{B}'_{n-4}$ uniformly (e.g. using parenthesis words).
(2) Perform the closure of T to obtain an irreducible dissection $D \in \mathcal{D}_{n-4}$. Choose randomly one of the six edges of the hexagon of D to carry the root. If D is not undecomposable, then reject and restart from 1.
(3) Add an edge e in the outer face of D going, with the infinite face on its right, from the root-vertex v of D to the vertex diametrically opposed to v on the hexagon. Take e as root edge, with v as root-vertex. As D is undecomposable, this operation yields a rooted irreducible quadrangulation Q with n faces.
(4) Return the rooted 3-connected map in $\mathcal{P}'_n$ associated to Q by the angular mapping.

PROPOSITION 4.3. *The success probability of the sampler at each trial is equal to $|\mathcal{P}'_n|/|\mathcal{D}'_{n-4}|$, which satisfies*

$$\frac{|\mathcal{P}'_n|}{|\mathcal{D}'_{n-4}|} \underset{n\to\infty}{\to} \frac{2^8}{3^6}.$$

The number of rejections follows a geometric law whose mean is asymptotically $c = \frac{3^6}{2^8}$, so that the random generator has expected linear-time complexity.

Proof. According to Corollary 3.1, $|\mathcal{D}'_n| = \frac{6}{n+2}|\mathcal{B}'_n| = \frac{6(2n)!}{(n+2)!n!}$. Stirling's formula yields $|\mathcal{D}'_{n-4}| \sim \frac{3}{128\sqrt{\pi}}\frac{4^n}{n^{5/2}}$. Moreover, according to [**113**], $|\mathcal{P}'_n| \sim \frac{2}{3^5\sqrt{\pi}}\frac{4^n}{n^{5/2}}$. This yields the limit of $|\mathcal{P}'_n|/|\mathcal{D}'_{n-4}|$. Finally, the linear time complexity of the sampler results from the linear complexity of testing undecomposability of a dissection and the linear time complexity of the closure mapping, which is obtained by putting —during the ccw traversal— the stems and sides of edges in a stack (see [**94**, Sec. 8.3.3] for a discussion on the implementation of such bijections). □

3.1.2. *Sampling rooted 3-connected maps with i vertices and j faces.* Similarly, the following algorithm samples rooted 3-connected maps with i vertices and j faces uniformly at random:

(1) Sample an object $T \in \mathcal{B}^\bullet_{i-3,j-3}$ uniformly. A simple method is described in Section 2.1.5.
(2) Perform the closure of T to obtain an irreducible dissection D with $i-3$ black vertices and $j-3$ white vertices. Choose randomly the root-vertex among the three black vertices of the hexagon. If the dissection is not undecomposable, then reject and restart.
(3) Add an edge e in the outer face of D going, with the infinite face on its right, from the root-vertex v of D to the vertex of the hexagon diametrically opposed to v. Take e as root edge, with v as root-vertex. This gives

a rooted irreducible quadrangulation Q with i black vertices and j white vertices.

(4) Return the rooted 3-connected map in $\mathcal{P}'_{i,j}$ associated to Q by the angular mapping.

PROPOSITION 4.4. *The success probability of the sampler at each trial is equal to* $|\mathcal{P}'_{ij}|/|D_{i-3,j-3}|$. *Let* $\alpha \in]1/2, 2[$*; if* i *and* j *are correlated by* $\frac{i}{j} \to \alpha$ *as* $i \to \infty$, *then*

$$\frac{|\mathcal{P}'_{ij}|}{|\mathcal{D}'_{i-3,j-3}|} \sim \frac{2^8}{3^6}\frac{(2-\alpha)^2(2\alpha-1)^2}{\alpha^2} =: \frac{1}{c_\alpha}.$$

Hence, when $\frac{i}{j} \to \alpha$, *the number of rejections follows a geometric law whose mean is asymptotically* c_α. *Under these conditions, the sampling algorithm has expected linear-time complexity, the linearity factor being asymptotically proportional to* c_α.

Moreover, in the worst case of triangulations, where $j = 2i - 4$, *the mean number of rejections is quadratic, so that the sampling complexity is cubic.*

Proof. These asymptotic results are easy consequences of the expression of $|\mathcal{D}'_{ij}|$ obtained in Corollary 3.1 and of the asymptotic result $|\mathcal{P}'_{ij}| \sim \frac{1}{3^5 2^2 ij}\binom{2i-2}{j+2}\binom{2j-2}{i+2}$ given in [**6**]. □

3.2. Random sampling of rooted 4-connected triangulations. Sampling rooted 4-connected triangulations follows the same lines as for rooted 3-connected maps. The bijection with ternary trees yields a uniform sampler for rooted irreducible triangulations with a given size. From there we obtain a rejection sampler for rooted 4-connected triangulations, which can be seen as a subset of rooted irreducible triangulations.

Sampling irreducible triangulations. The closure mapping (rooted formulation) yields a very efficient uniform sampler for objects of $\mathcal{T}'_n$. As $\mathcal{A}'_n$ is in 4-to-$(n+2)$ correspondence with $\mathcal{T}'_n$, the uniform distribution on $\mathcal{A}'_n$ is transported by the closure mapping into the uniform distribution on $\mathcal{T}'_n$. Moreover, it is well known how to design a linear time uniform sampler SAMPLEA(n) for ternary trees with n nodes, using parenthesis words. Thus we have the following uniform sampler for $\mathcal{T}'_n$:

SAMPLET(n): 1) $\tau \leftarrow$ SAMPLEA(n)
2) $T \leftarrow$ CLOSURE(τ)
3) Choose a border edge e of T at random
4) Root T at e and return T

As the closure mapping can be implemented to run in linear time, the algorithm SAMPLET(n) has also linear time complexity. Then, a uniform sampler for rooted 4-connected triangulations of size $n \geq 5$ is easily obtained by adding to SAMPLET(n) a rejection loop conditioned to output an undecomposable triangulation. We use also a simple procedure ADDROOTEDGE that, given $T \in \mathcal{T}'$, adds an edge e going from the root vertex of T to the diametrically opposed vertex of the outer quadrangle with the outer face on its right, and then roots the obtained triangulation at the edge e.

SAMPLEC(n): 1) repeat $T \leftarrow$ SAMPLET$(n-4)$ until $T \in \mathcal{U}$
2) return ADDROOTEDGE(T)

PROPOSITION 4.5. *For* $n \geq 5$, SAMPLEC*(n) is a uniform sampler for rooted 4-connected triangulations with* n *vertices, with expected linear time complexity. The*

number of trials in the loop (i.e., the number of calls to SAMPLET$(n-4)$*) follows a geometric law of parameter* $|\mathcal{C}'_n|/|\mathcal{T}'_{n-4}| \sim_{n\to\infty} 3^6/2^{10} \approx 0.712$.

Proof. All stated results follow clearly from the above discussion except for the limit $|\mathcal{C}'_n|/|\mathcal{T}'_{n-4}| \sim 3^6/2^{10}$. This limit follows from $|\mathcal{C}'_n| \sim 3^3/2^9\sqrt{3/\pi}n^{-5/2}(27/4)^{n-3}$ and $|\mathcal{T}'_n| \sim 1/2\sqrt{3/\pi}n^{-5/2}(27/4)^n$. The estimate for $|\mathcal{T}'_n|$ is obtained by applying Stirling's formula to the expression given in Proposition 3.7. The estimate for $|\mathcal{C}'_n|$ was found by Tutte [**111**] and can also be derived from the expression of $C(z)$ given in Proposition 4.2 by applying singularity analysis, as detailed in [**51**]. □

4. Random sampling of planar graphs

4.1. Introduction. Let us now focus on the random generation of planar *graphs*, i.e., graphs having at least one planar embedding. The graphs we consider here are labeled (each of the n vertices carries a distinct label in $[1..n]$) and have no loops nor multiple edges. Notice that a uniform generator for labeled planar *maps* yields a generator for labeled connected planar *graphs* such that each graph has a weight proportional to its number of embeddings. Hence, generating planar graphs uniformly is different from generating planar maps uniformly, so that our bijective generators for maps can not be applied directly. A first algorithm for the random generation of planar graphs was proposed by Denise, Vasconcellos, and Welsh [**40**], where a Markov chain on the set $\mathcal{G}_n$ of labeled planar graphs with n vertices is defined. At each step, two different vertices v and v' are chosen at random. If they are adjacent, the edge (v, v') is deleted. If they are not adjacent and if the operation of adding (v, v') does not break planarity, then the edge (v, v') is added. By symmetry of the transition matrix of the Markov chain, the probability distribution converges to the uniform distribution on $\mathcal{G}_n$. This algorithm is very easy to describe but more difficult to implement, as there exists no simple linear-time planarity testing algorithm. Moreover, the rate of convergence to the uniform distribution is unknown.

Our approach is very different. The key ingredient is a well-known decomposition of planar graphs by increasing degree of connectivity. To sum up, a planar graph can be decomposed into a tree-like structure whose nodes are occupied by rooted 3-connected planar graphs. According to a theorem of Whitney [**117**], rooted 3-connected planar graphs have a unique (unoriented) embedding, so that rooted 3-connected planar graphs are isomorphic to rooted 3-connected maps. Essentially, our generator generates the tree-like structure using specific branching probabilities, and then calls the bijective generator for 3-connected maps, described in Section 3.1, for the generation at each node of the tree-like structure. The branching probabilities are values of generating functions of various families of planar graphs (connected, 2-connected,...). In this thesis, we only focus on the complete description of the algorithm. The proofs are in the full paper to be submitted [**62**].

Let us mention that the decomposition of planar graphs by increasing degree of connectivity is also used by Bodirsky *et al* [**13**] to obtain the first uniform sampler for planar graphs with polynomial complexity. As in our case, their approach is to translate the decomposition of planar graphs into a random generator. The main difference relies in the framework used to translate the decomposition; the authors of [**13**] use the recursive method introduced by Nijenhuis and Wilf [**91**] —later further formalized by Flajolet *et al* [**53**]—, whereas we use the recently introduced framework of Boltzmann samplers [**43**]. Both frameworks make it possible to translate a combinatorial decomposition to a random generator in a quite automatic way.

	Aux. mem.	Preproc. time	Time per generation	
Markov chains	$\mathcal{O}(\log n)$	$\mathcal{O}(\log n)$	*unknown*	{exact size}
Recursive method	$\mathcal{O}(n^5 \log n)$	$\mathcal{O}^*(n^7)$	$\mathcal{O}(n^3)$	{exact size}
Boltzmann sampler	$\mathcal{O}((\log n)^k)$	$\mathcal{O}((\log n)^k)$	$\mathcal{O}(n^2)$ $\mathcal{O}(n)$	{exact size} {approx. size}

FIGURE 3. Complexities of the random samplers of planar graphs ($\mathcal{O}^*$ stands for a big $\mathcal{O}$ taken up to logarithmic factors).

In the case of the recursive method, the branching probabilites used to generate the object are expressed in terms of the *coefficients* counting the classes involved in the decomposition. When the objects to be generated are large, this requires to compute large tables of large integers (the coefficients), so that the generation is mostly limited to sizes of a few thousands in the best case, a few hundreds for more involved classes like planar graphs. In contrast, the branching probabilities in Boltzmann samplers involve only a *finite* number of real constants, which are evaluations of generating functions of the classes involved in the decomposition. In practice, these constants are truncated to a fixed precision (e.g. 20 digits), with no detectable bias from uniformity. Boltzmann samplers have mostly linear complexity in the size of the output, making it possible to generate objects of size 10^6 when suitably implemented. The price to pay is that the size of the output is not fixed. Nevertheless, it is often possible to tune the sampler so that the size of the output is around a target size with high probability.

Translating the decomposition of planar graphs into Boltzmann samplers, we obtain very efficient random generators, which produce planar graphs with a fixed number of vertices or with fixed numbers of vertices and edges uniformly at random. Furthermore, our samplers have an approximate-size version where a small relative range, say a few percents, is allowed for the size of the output. For practical purpose, approximate-size random sampling often suffices. The approximate-size samplers we propose are very efficient as they have linear time complexity.

THEOREM 4.3 (Samplers with respect to number of vertices). *For $n \in \mathbb{N}$, there is an* exact-size sampler $\mathfrak{A}_n$ *producing labelled planar graphs with n vertices uniformly at random. For any* tolerance ratio $\epsilon > 0$*, there is an* approximate-size *sampler $\mathfrak{A}_{n,\epsilon}$ producing random planar graphs with number of vertices in $[n(1-\epsilon), n(1+\epsilon)]$ such that the distribution is uniform on each size $k \in [n(1-\epsilon), n(1+\epsilon)]$.*

Under a real-artithmetic complexity model, Algorithm $\mathfrak{A}_n$ is of expected complexity $\mathcal{O}(n^2)$. Algorithm $\mathfrak{A}_{n,\epsilon}$ is of expected complexity $\mathcal{O}(n)$, where the linearity constant depends on ϵ, being of order $1/\epsilon$ as $\epsilon \to 0$.

THEOREM 4.4 (Samplers with respect to the numbers of vertices and edges). *Let $n \in \mathbb{N}$ be a target size and $\mu \in (1,3)$ be a parameter of ratio edges-vertices. There exists an* exact-size *sampler $\mathfrak{A}_{n,\mu}$ producing planar graphs with n vertices and $\lfloor \mu n \rfloor$ edges uniformly at random. For any tolerance-ratio $\epsilon > 0$, there exists an* approximate-size *sampler $\mathfrak{A}_{n,\mu,\epsilon}$ producing random planar graphs with number of vertices in $[n(1-\epsilon), n(1+\epsilon)]$ and ratio edges/vertices in $[\mu(1-\epsilon), \mu(1+\epsilon)]$, such that the distribution is uniform for each fixed pair (number of vertices, number of edges).*

Under a real-artithmetic complexity model, for a fixed $\mu \in (1,3)$, Algorithm $\mathfrak{A}_{n,\mu}$ is of expected complexity $\mathcal{O}(n^{5/2})$, where the constant depends on μ. For fixed

$\mu \in (1,3)$ and $\epsilon > 0$, Algorithm $\mathfrak{A}_{n,\mu,\epsilon}$ is of expected complexity $\mathcal{O}(n)$, the constant of linearity depending both on μ and on ϵ, being of order $1/\epsilon$ as $\epsilon \to 0$.

The samplers are completely stated in Section 4.5.3 and Section 4.5.4.

The real-arithmetic complexity model is that of the number of arithmetic operations (additions, comparisons) over real numbers assumed to be known *exactly*. The complexity of our algorithm is compared to the complexities of the two preceding random samplers in Figure 3.

Let us comment on the practical preprocessing complexity. The implementation of $\mathfrak{A}_{n,\epsilon}$ and $\mathfrak{A}_n$, as well as $\mathfrak{A}_{n,\mu,\epsilon}$ and $\mathfrak{A}_{n,\mu}$, requires the storage of a fixed number of real constants, which are special values of generating functions. The generating functions to be evaluated are those of several families of planar graphs (connected, 2-connected, 3-connected). A crucial result, recently established by O. Giménez and M. Noy [**68**], is that there exist exact analytic equations satisfied by these generating functions. Hence, their numerical evaluation can be performed efficiently, the complexity being of low polynomial degree k in the number of digits that need to be computed.

Fixed-size truncation of real numbers leads to algorithms with a probability of failure (caused by lack of precision) that can be made arbitrarily close to 0. No failure arises with a precision of 20 digits in practice, when we draw objects of size up to the million. In general, to draw objects of size n, the precision needed to make the probability of failure small is of order $\log(n)$ digits. Thus the preprocessing step to evaluate the generating functions with a precision of $\log(n)$ digits has a complexity of order $\log(n)^k$. Notice that it is possible to achieve perfect uniformity by calling adaptative precision routines in case of failure, see Denise and Zimmermann [**41**] for a detailed discussion on similar problems. The following statement summarizes the discussion:

FACT 4.1. *In practice, the auxiliary memory necessary to generate planar graphs of size n is of order $\mathcal{O}(\log(n))$ and the preprocessing time complexity is of order $\mathcal{O}(\log(n)^k)$ for some small integer k.*

The evaluation of the generating functions of planar graphs has been carried out with the mathematical software Maple, based on the analytic expressions given by Giménez and Noy [**68**]. Then, the random generator has been implemented in Java, with a precision of 64 bits for the values of generating functions ("double" type). Using the approximate-size sampler, planar graphs with size of order 100,000 are generated in a few seconds with a machine clocked at 1GHz. In contrast, the recursive method of Bodirsky *et al* is currently limited to sizes of about 100.

4.2. Overview. The algorithm we propose relies on several tools. First, we extend the collection of constructions for Boltzmann samplers, as detailed in [**43**], and develop the more complicated case of substitution constructions, see Section 4.3. We describe in Section 4.4 the recursive decomposition of planar graphs according to successive levels of connectivity (also used in [**13**]) and adapt it to the Boltzmann framework. We start with the development of a Boltzmann sampler for (edge-rooted) 3-connected planar graphs. To do this, we use the bijection between irreducible dissections and binary trees. The realisation of a Boltzmann sampler for binary trees is straightforward and it yields, via the bijection (combined with rejection techniques), a Boltzmann sampler for edge-rooted 3-connected planar graphs. The next step is the realisation of a Boltzmann sampler for 2-connected planar graphs. A well-known decomposition —originally due to Trakhtenbrot and

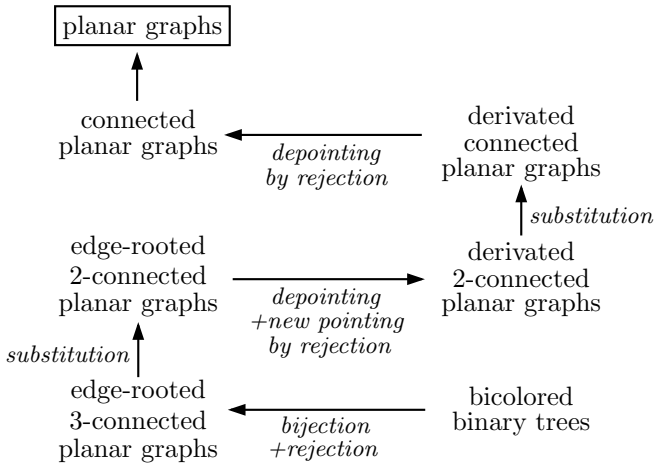


FIGURE 4. The chain of reductions from planar graphs to binary trees.

described in details in [**116**]— ensures that edge-rooted 2-connected planar are assembled in a unique way from edge-rooted 3-connected planar graphs. Translating the decomposition yields a Boltzmann sampler for edge-rooted 2-connected planar graphs. Then we develop a Boltzmann sampler for connected planar graphs, using another decomposition ensuring that vertex-rooted connected planar graphs are assembled in a unique way from vertex-rooted 2-connected planar graphs. Finally, we obtain a Boltzmann sampler for (unconstrained) planar graphs, resulting from the decomposition of planar graphs into connected components. The corresponding Boltzmann sampler is denoted by $\Gamma G(x, y)$, where the variable x marks the number of vertices and the variable y marks the number of edges.

The Boltzmann sampler $\Gamma G(x, y)$ can unfortunately not be used directly to generate large planar graphs with a good time complexity. Indeed, the size distribution of $\Gamma G(x, y)$ is too concentrated on objects of small size. To improve the size distribution, we *point* the objects, in a way inspired by [**43**], which corresponds to a *differentiation* of the associated generating function. The precise singularity analysis of the generating functions of planar graphs, recently performed in [**68**], indicates that we have to perform derivation of planar graphs three times in order to get a usable size distribution. In Section 4.5, we explain how to inject the derivative operator into the decomposition of planar graphs. This gives a Boltzmann sampler $\Gamma G'''(x, y)$ for "triply derived" planar graphs. Our random generators of planar graphs are finally obtained as *targetted samplers*, starting from $\Gamma G'''(x, y)$ and choosing well tuned values $x = x_n$ and $y = y(\mu)$ for each target size n and ratio edges/vertices $\mu \in (1, 3)$. The general scheme of the planar graph generator is shown in Figure 4.

4.3. Boltzmann samplers. In this section, we define Boltzmann samplers and describe the main properties which we will need to develop a Boltzmann sampler for planar graphs in Section 4.4. In particular, we have to extend the framework to the case of *mixed classes*, meaning that the objects have two types of atoms. Indeed the decomposition of planar graphs involves both (labeled) vertices and (unlabeled) edges. The constructions needed to formulate the decomposition

of planar graphs are classical ones in combinatorics: Sum, Product, Set, Substitutions. For each of the constructions, we describe a *sampling rule*, so that Boltzmann samplers can be assembled for any class that admits a decomposition in terms of these constructions. Morevover, the decomposition of planar graphs involves rooting/unrooting operations. Taking these operations into account in the samplers makes it necessary to inject some rejection techniques, as well as derivative operators, in the framework of Boltzmann samplers.

4.3.1. *Definition.* Boltzmann samplers, introduced and developed by Duchon *et al* in [**43**], constitute a general and efficient framework to produce a random generator on a combinatorial class $\mathcal{C}$ that admits an explicit decomposition. Instead of fixing a particular size for the random generation, objects are drawn under a probability distribution spread over the whole class. This distribution assigns to each object of a combinatorial class $\mathcal{C}$ a weight essentially proportional to the exponential of its size n. Precisely, if $\mathcal{C}$ is an unlabelled class, the *ordinary* generating function of $\mathcal{C}$ is

$$C(y) := \sum_{\gamma \in \mathcal{C}} y^{|\gamma|},$$

where $|\gamma|$ stands for the size (e.g. the number of nodes in a tree) of γ, and y is a variable marking the size. It is clear that the sum defining $C(y)$ converges if y is smaller than the radius of convergence ρ_C of $C(y)$, in which case y is said to be *coherent*. Then, the probability distribution assigining to each object γ of $\mathcal{C}$ a probability

$$\mathbf{P}_y(\gamma) = \frac{y^{|\gamma|}}{C(y)}$$

is a well defined distribution, called *ordinary Boltzmann distribution* of parameter y. An *ordinary Boltzmann sampler* of parameter y is a procedure $\Gamma C(y)$ that draws objects of $\mathcal{C}$ at random under the Boltzmann distribution $\mathbf{P}_y$. The authors of [**43**] provide a collection of rules to assemble Boltzmann samplers for combinatorial classes specified using basic combinatorial constructions, like Sum, Product, Sequence. The framework has been recently extended to constructions that are subject to symmetries, like Multiset, Powerset, Cycle [**50**]. An interesting application to random sampling of plane partitions is developed in [**11**].

Boltzmann samplers can similarly be assembled in the framework of labelled objects (e.g. graphs with labelled vertices). The *exponential* generating function of the class $\mathcal{C}$ is defined as

$$C(x) := \sum_{\gamma \in \mathcal{C}} \frac{x^{|\gamma|}}{|\gamma|!},$$

where $|\gamma|$ is the size of an object $\gamma \in \mathcal{C}$ (e.g. the number of vertices of a graph). The exponential Boltzmann distribution assigns to each object of $\mathcal{C}$ a weight

$$\mathbf{P}_x(\gamma) = \frac{x^{|\gamma|}}{|\gamma|! C(x)}\,.$$

Given a coherent value x, i.e., a value smaller than the radius of convergence of $C(x)$, a *Boltzmann sampler* for the labelled class $\mathcal{C}$ is a procedure $\Gamma C(x)$ that draws objects of $\mathcal{C}$ at random under the "labelled" Boltzmann distribution $\mathbf{P}_x$. As in the unlabelled framework, the authors of [**43**] give sampling rules associated to classical combinatorial constructions (Sum, Product, Set).

To assemble a Boltzmann sampler for planar graphs from their combinatorial decomposition, we need to extend the framework of Boltzmann samplers to the case

of a *mixed* combinatorial class. In a mixed class $\mathcal{C} = \cup_{n,m}\mathcal{C}_{n,m}$, an object has n labelled "atoms" and m unlabelled "atoms", e.g., a graph with n labelled vertices and m unlabelled edges. For $\gamma \in \mathcal{C}$, we write $|\gamma|$ for the number of labelled atoms of γ and $||\gamma||$ for the number of unlabelled atoms of γ. The associated generating function $C(x, y)$ is defined as

$$C(x, y) := \sum_{\gamma\in\mathcal{C}} \frac{x^{|\gamma|}}{|\gamma|!} y^{||\gamma||}.$$

For a fixed real value $y > 0$, we denote by $\rho_C(y)$ the radius of convergence of the function $x \to C(x, y)$. A pair (x, y) is said to be *coherent* if $x \in (0, \rho_C(y))$, which means that $\sum_{\gamma\in\mathcal{C}} \frac{x^{|\gamma|}}{|\gamma|!} y^{||\gamma||}$ converges and that $C(x, y)$ is well defined. Given a coherent pair (x, y), the *mixed Boltzmann distribution* is the probability distribution $\mathbf{P}_{x,y}$ assigning to each object $\gamma \in \mathcal{C}$ probability

$$\mathbf{P}_{x,y}(\gamma) = \frac{1}{C(x, y)} \frac{x^{|\gamma|}}{|\gamma|!} y^{||\gamma||}.$$

An important property of this distribution is that two objects with the same parameters $(|\gamma|, ||\gamma||)$ have the same probability. A *mixed Boltzmann sampler* at (x, y) —shortly called Boltzmann sampler hereafter— is a procedure $\Gamma C(x, y)$ that draws objects of $\mathcal{C}$ at random under the Boltzmann distribution $\mathbf{P}_{x,y}$. Observe that the development of the Boltzmann framework for mixed classes is an extension of the labelled case studied in [**43**]. Indeed, $\Gamma C(x, 1)$ is an exponential Boltzmann sampler for $\mathcal{C}$.

4.3.2. *Constructions.* The five constructions that follow serve to describe the decomposition of planar graphs, see [**52**] for a detailed description. We need two substitution constructions, one at labelled atoms called x-substitution, the other at unlabelled atoms called y-substitution.

Sum. The sum $\mathcal{C} = \mathcal{A} + \mathcal{B}$ of two classes is meant as a *disjoint union*, i.e., it is the union of two distinct copies of $\mathcal{A}$ and $\mathcal{B}$. The generating function of $\mathcal{C}$ satisfies

$$C(x, y) = A(x, y) + B(x, y).$$

Product. The product $\mathcal{C} = \mathcal{A} \star \mathcal{B}$ is a classical cartesian product, combined with a relabelling step ensuring that the atoms of an object $\gamma \in \mathcal{A} \star \mathcal{B}$ bear distinct labels in $[1, \ldots, |\gamma|]$. The generating function of $\mathcal{C}$ satisfies

$$C(x, y) = A(x, y) \cdot B(x, y).$$

Set$_{\geq \mathbf{d}}$. For $d \geq 0$ and a class $\mathcal{A}$ having no object of size 0, $\mathcal{C} = \textsc{Set}_{\geq d}(\mathcal{A})$ is the class such that each object $\gamma \in \textsc{Set}_{\geq d}(\mathcal{A})$ is a finite set of at least d objects of $\mathcal{A}$, relabelled so that the atoms of γ bear distinct labels in $[1\,..\,|\gamma|]$. For $d = 0$, this corresponds to the classical construction Set. The generating function of $\mathcal{C}$ satisfies

$$C(x, y) = e_d(A(x, y)), \quad \text{where } e_d(Z) := \sum_{k\geq d} \frac{Z^k}{k!}.$$

x-substitution. Given $\mathcal{A}$ and $\mathcal{B}$ two classes such that $\mathcal{B}$ has no object of size 0, the class $\mathcal{C} = \mathcal{A} \circ_x \mathcal{B}$ is the class of objects that are obtained by taking an object $\rho \in \mathcal{A}$, called the *core-object*, substituting each labelled atom v of ρ by an object

Construction		Boltzmann sampler
empty atom	$\mathbf{1}$	return $\mathbf{1}$
unit atom	$\mathcal{Z}$	return $\mathcal{Z}$
Sum	$\mathcal{C} = \mathcal{A} + \mathcal{B}$	$\Gamma C(x,y)$: if Bern $\left(\frac{A(x,y)}{C(x,y)}\right)$, return $\Gamma A(x,y)$ else return $\Gamma B(x,y)$
Product	$\mathcal{C} = \mathcal{A} \star \mathcal{B}$	$\Gamma C(x,y)$: $\gamma \leftarrow (\Gamma A(x,y), \Gamma B(x,y))$ DistributeLabels(γ); return γ
$\mathrm{Set}_{\geq d}$	$\mathcal{C} = \mathrm{Set}_{\geq d}(\mathcal{A})$	$\Gamma C(x,y)$: $k \leftarrow \mathrm{Pois}_{\geq d}(A(x,y))$ $\gamma \leftarrow (\Gamma A(x,y), \ldots, \Gamma A(x,y))$ {k *ind. calls*} DistributeLabels(γ); return γ
x-subs	$\mathcal{C} = \mathcal{A} \circ_x \mathcal{B}$	$\Gamma C(x,y)$: $\gamma \leftarrow \Gamma A(B(x,y), y)$ for each labeled atom $v \in \gamma$ do replace v by $\gamma_v \leftarrow \Gamma B(x,y)$ od {*ind. calls*} DistributeLabels(γ); return γ
y-subs	$\mathcal{C} = \mathcal{A} \circ_y \mathcal{B}$	$\Gamma C(x,y)$: $\gamma \leftarrow \Gamma A(x, B(x,y))$ for each unlabeled atom $e \in \gamma$ do replace e by $\gamma_e \leftarrow \Gamma B(x,y)$ od {*ind. calls*} DistributeLabels(γ); return γ

FIGURE 5. The sampling rules associated with each of the five constructions.

$\gamma_v \in \mathcal{B}$, and finally relabelling the atoms of $\cup_v \gamma_v$ with distinct labels from 1 to $\sum_v |\gamma_v|$. The generating function of $\mathcal{C}$ satisfies

$$C(x,y) = A(B(x,y), y).$$

y-substitution. Given $\mathcal{A}$ and $\mathcal{B}$ two classes such that $\mathcal{B}$ has no object of size 0, the class $\mathcal{C} = \mathcal{A} \circ_y \mathcal{B}$ is the class of objects that are obtained by taking an object $\rho \in \mathcal{A}$, called the *core-object*, substituting each unlabelled atom e of ρ by an object $\gamma_e \in \mathcal{B}$, and finally relabelling the atoms of $\rho \cup (\cup_e \gamma_e)$ with distinct labels from 1 to $|\rho| + \sum_e |\gamma_e|$. We assume here that the unlabelled atoms of an object of $\mathcal{A}$ are *distinguishable*. This property is satisfied in the case where $\mathcal{A}$ is a family of labelled graphs with no multiple edges, as two different edges are distinguished by the labels of their two incident vertices. The generating function of $\mathcal{C}$ satisfies

$$C(x,y) = A(x, B(x,y)).$$

4.3.3. *Sampling rules.* A nice feature of Boltzmann samplers is that the basic combinatorial constructions (Sum, Product, Set) give rise to simple rules for assembling the associated Boltzmann samplers. To describe these rules, we assume that the exact values of the generating functions at a given coherent pair (x,y) are known. We will also need two well-known probability distributions.

- A *Bernoulli law* of parameter $p \in (0,1)$ is a random variable equal to 1 (or true) with probability p and equal to 0 (or false) with probability $1-p$.
- Given $\lambda > 0$ a real value and d a nonnegative integer, $\mathrm{Pois}_{\geq d}(\lambda)$ is the law of a random variable taking values in $\mathbf{Z}_{\geq d}$ such that

$$\mathbb{P}(k) = \frac{1}{e_d(\lambda)} \frac{\lambda^k}{k!}, \text{ where } e_d(Z) := \sum_{k \geq d} \frac{Z^k}{k!}.$$

For $d = 0$, this corresponds to the classical Poisson law, abbreviated as Pois.

For complexity analysis, a Bernoulli choice is assumed to have unit cost, and drawing from a Poisson law has cost equal to the value of the output. (Indeed, a Poisson law is classically drawn using a loop running k times if the result is k, see [**52**].)

Starting from combinatorial classes $\mathcal{A}$ and $\mathcal{B}$ endowed with Boltzmann samplers $\Gamma A(x, y)$ and $\Gamma B(x, y)$, Figure 5 describes how to assemble a sampler for a class $\mathcal{C}$ obtained from $\mathcal{A}$ and $\mathcal{B}$ (or from $\mathcal{A}$ alone for the construction $\textsc{Set}_{\geq d}$) using the five constructions. The relabelling step, as mentioned in the definition of the constructions, is performed by an auxiliary procedure DistributeLabels. Given an object γ with its labelled atoms ranked from 1 to $|\gamma|$, DistributeLabels(γ) draws a permutation σ of $[1, \ldots, |\gamma|]$ uniformly at random and gives label $\sigma(i)$ to the atom of rank i.

Proposition 4.6. *Let $\mathcal{A}$ and $\mathcal{B}$ be two mixed combinatorial classes endowed with Boltzmann samplers $\Gamma A(x, y)$ and $\Gamma B(x, y)$. Then, for each of the five constructions $\{+, \star, \textsc{Set}_{\geq d}, x-\text{subs}, y-\text{subs}\}$, the sampler $\Gamma C(x, y)$, as defined in Figure 5, is a valid Boltzmann sampler for the combinatorial class $\mathcal{C}$.*

Example 1. Consider the class $\mathcal{C}$ of labelled binary trees where the atoms are the nodes. The class $\mathcal{C}$ has the following decomposition grammar,

$$\mathcal{C} = (\mathcal{C} + \mathbf{1}) \star \mathcal{Z} \star (\mathcal{C} + \mathbf{1}) .$$

Hence, the series $C(x)$ counting binary trees satisfies $C(x) = x\,(1 + C(x))^2$. Thus, $C(x)$ can be easily evaluated for a fixed real parameter $x < \rho_C = \frac{1}{4}$.

Using the sampling rules for Sum and Product, we obtain the following Boltzmann sampler for binary trees,

$\Gamma C(x)$: return $(\Gamma(1 + C)(x), \mathcal{Z}, \Gamma(1 + C)(x))$ {independent calls}

$\Gamma(1 + C)(x)$: if Bern $\left(\frac{1}{1+C(x)}\right)$ return $\varnothing$ (leaf)
else return $\Gamma C(x)$

Remark 1. The procedure DistributeLabels(γ) throws distinct labels uniformly at random on the atoms of γ that support labels. The fact that the relabelling permutation is always chosen uniformly ensures that the call to DistributeLabels can be postponed till the end of the algorithm, i.e., we can apply the labelling to the finally output object (this is also mentioned by Flajolet *et al* [**53**, Sec 3]). Hence, the labels do not really matter and induce no additional complexity to the Boltzmann samplers: for a class $\mathcal{C}$ whose combinatorial decomposition involves the five constructions, we just have to generate the unlabelled *shape* of an object γ produced by $\Gamma C(x, y)$; then we call DistributeLabels(γ).

4.3.4. *Useful techniques related to Boltzmann samplers.* In the following sections, we will make much use of the *derivative* operator. Given a mixed (or labelled) combinatorial class $\mathcal{C} = \cup_{n,m}\mathcal{C}_{n,m}$, an object of the *derived class* $\mathcal{C}'$ is obtained by removing the label n of an object of $\mathcal{C}$ of size n, so that the obtained object has size $n - 1$ (the atom n can be considered as a *pointed* atom that does not count in the size). As a consequence, $\mathcal{C}'_{n-1,m} \simeq \mathcal{C}_{n,m}$, so that the generating function $C'(x, y)$ of $\mathcal{C}'$ satisfies

$$C'(x, y) = \sum_{n,m} |\mathcal{C}_{n,m}| \frac{x^{n-1}}{(n-1)!} y^m = \frac{\partial C}{\partial x}(x, y). \tag{20}$$

The y-derivative of $\mathcal{C}$ is the class $\overline{\mathcal{C}}$ of objects of $\mathcal{C}$ having a marked unlabelled atom that does not count in the size. Thus, the generating function $\overline{C}(x,y)$ of $\overline{\mathcal{C}}$ satisfies

$$\overline{C}(x,y) = \frac{\partial C}{\partial y}(x,y). \tag{21}$$

For the particular case of a class of planar graphs, we will also consider *edge-rooted* objects, i.e., planar graphs where an edge is marked and oriented. In addition, the root edge is not counted as unlabelled atom, and the two extremities of the root do not count as labelled atoms (i.e., are not labelled). The edge-rooted class of $\mathcal{C}$ is denoted by $\overrightarrow{\mathcal{C}}$. The generating function $\overrightarrow{C}(x,y)$ of $\overrightarrow{\mathcal{C}}$ satisfies

$$\overrightarrow{C}(x,y) = \frac{2}{x^2}\frac{\partial C}{\partial y}(x,y). \tag{22}$$

Another useful techique is *rejection*, which offers great flexibility and makes it possible to adjust the distributions of the samplers. This technique will prove particularly useful to cope with re-pointing/depointing operations that intervene in the decomposition of planar graphs.

LEMMA 4.4 (Rejection). *Given a combinatorial class $\mathcal{C}$, let $W : \mathcal{C} \to \mathbb{R}^+$ and $p : \mathcal{C} \to [0,1]$ be two functions, called* weight-function *and* rejection-function*, respectively. Assume that W is summable, i.e., $\sum_{\gamma\in\mathcal{C}} W(\gamma)$ is finite. Let $\mathfrak{A}$ be a random generator on $\mathcal{C}$ that draws each object $\gamma \in \mathcal{C}$ with probability proportional to $W(\gamma)$. Then, the procedure*

$$\mathfrak{A}_{\text{rej}} : \text{repeat } \mathfrak{A} \to \gamma \text{ until } \text{Bern}(p(\gamma));\ \text{return } \gamma$$

is a random generator on $\mathcal{C}$, which draws each object $\gamma \in \mathcal{C}$ with probability proportional to $W(\gamma)p(\gamma)$.

4.4. Decomposition of planar graphs and Boltzmann samplers. The classical method to count planar graphs consists in decomposing a planar graph into planar components that have higher degree of connectivity. The decomposition is stopped at connectivity degree 3, where the graphs have a unique planar embedding. Recall that a graph is k-connected if it has at least k vertices and if the removal of any set of $k-1$ vertices and their incident edges does not disconnect the graph. The generation method we describe *reverses* the decomposition, i.e., planar graphs are assembled in a way that follows the decomposition.

First, by uniqueness of the embedding, generating 3-connected planar graphs is equivalent to generating 3-connected maps, which is done using the sampler presented in Section 3.1 (turned into a Boltzmann sampler). The next step is to generate 2-connected planar graphs from 3-connected ones. We take advantage of a decomposition of 2-connected planar graphs into 3-connected planar components, which has been formalised by Trakhtenbrot [**109**] and later used by Walsh [**116**] to count 2-connected planar graphs and by Bender, Gao, Wormald to obtain asymptotic enumeration [**5**]. Finally, connected planar graphs are generated from 2-connected ones by using a well-known decomposition at separating vertices, and planar graphs are generated from connected ones by choosing the number of connected components and then generating each component. Notice that these steps translate to explicit equations relating the generating functions of 2-connected, connected, and unconstrained planar graphs. Starting from these equations, Giménez

and Noy have solved the asymptotic enumeration of planar graphs, using analytic methods and clever integral manipulations [**68**].

Notations. In the sequel, the number of vertices and the number of edges of a planar graph γ are respectively denoted by $V(\gamma)$ and $E(\gamma)$. Notice that $V(\gamma)$ might not be equal to $|\gamma|$ and $E(\gamma)$ might not be equal to $||\gamma||$, e.g., an edge-rooted planar graph γ satisfies $V(\gamma) = |\gamma| + 2$ and $E(\gamma) = ||\gamma|| + 1$.

4.4.1. *Boltzmann sampler for 3-connected planar graphs. Equivalence with rooted 3-connected maps* A well known result due to Whitney [**117**] states that a 3-connected planar graph has a unique embedding on the sphere up to continuous deformation and reflection (in general a planar graph can have many embeddings). Hence, a labelled edge-rooted 3-connected planar graphs gives rise to two different labelled 3-connected maps, the two maps differing by a reflection. The class of (unlabelled) rooted 3-connected maps is denoted by $\mathcal{M} = \cup_{i,j}\mathcal{M}_{i,j}$ where i is the number of vertices different from the two end points of the rooted edge and j is the number of edges without counting the rooted one. The associated generating function is $M(z,w) = \sum_{i,j} |\mathcal{M}_{i,j}| z^i w^j$ (as both vertices and edges are unlabelled, the series is ordinary in the two variables). Due to the absence of symmetries for rooted maps, the family of *labelled* 3-connected maps with $i+2$ vertices and j faces —the non-root vertices bearing distinct labels in $[1..i]$— is isomorphic to $\mathcal{M}_{i,j} \times i!$. Thus, Whitney's theorem yields

$$\mathcal{M}_{i,j} \times i! \simeq 2\overrightarrow{\mathcal{G}}_{3(i,j)}, \tag{23}$$

which can be written compactly as

$$\mathcal{M} \simeq 2\overrightarrow{\mathcal{G}}_3, \qquad M(z,w) = \frac{4}{z^2}\frac{\partial G_3}{\partial w}(z,w) \tag{24}$$

Definition 1. Let $\mathcal{C} = \cup_{i,j}\mathcal{C}_{i,j}$ be a class with two types of unlabelled atoms, called an *ordinary mixed class*. A *Boltzmann sampler* for $\mathcal{C}$ is a random generator $\Gamma C(x,y)$ drawing each object $\gamma \in \mathcal{C}_{i,j}$ with probability

$$\mathbb{P}(\gamma) = \frac{x^i y^j}{C(x,y)},$$

where $C(x,y) = \sum_{i,j} |\mathcal{C}_{i,j}| x^i y^j$ is the generating function of $\mathcal{C}$, which is ordinary in the two variables. The derived class $\mathcal{C}'$ and y-derived class $\overline{\mathcal{C}}$ are defined in the same way as for mixed classes; $\mathcal{C}'$ ($\overline{\mathcal{C}}$) is the class of objects of $\mathcal{C}$ having a marked atom of the first type (second type, respectively) that does not count in the size.

Equation (24) ensures that rooted 3-connected maps correspond to the unlabelled shape of edge-rooted 3-connected labelled planar graphs. In addition, according to Remark 1, it is sufficient to draw only the unlabelled shape of the objects, so that we have the following result.

LEMMA 4.5. *Finding a Boltzmann sampler $\Gamma\overrightarrow{G_3}(z,w)$ for edge-rooted 3-connected planar graphs is equivalent to finding a Boltzmann sampler $\Gamma M(z,w)$ for rooted 3-connected maps.*

Boltzmann sampler for binary trees. Notice that bicolored binary trees admit a recursive decomposition, so that a Boltzmann sampler is easily derived. Precisely, the class $\mathcal{T}$ of bicolored binary trees is partioned into the class $\mathcal{T}_\bullet$ of black-rooted binary trees and the class $\mathcal{T}_\circ$ of white-rooted binary trees. The associated ordinary

generating functions with respect to the number of black nodes (variable z) and the number of nodes (variable w) are denoted by $T(z,w)$, $T_\bullet(z,w)$, and $T_\circ(z,w)$. The decomposition at the root of a bicolored binary tree yields the following decomposition grammar, where $\mathcal{Z}_\bullet$ and $\mathcal{Z}_\circ$ stand for a black and a white node, respectively.

$$\left\{\begin{array}{rcl}\mathcal{T} &=& \mathcal{T}_\bullet + \mathcal{T}_\circ\\ \mathcal{T}_\bullet &=& (\mathbf{1}+\mathcal{T}_\circ)\star\mathcal{Z}_\bullet\star(\mathbf{1}+\mathcal{T}_\circ)\\ \mathcal{T}_\circ &=& (\mathbf{1}+\mathcal{T}_\bullet)\star\mathcal{Z}_\circ\star(\mathbf{1}+\mathcal{T}_\bullet)\end{array}\right. \qquad (25)$$

$$\left\{\begin{array}{rcl}T(z,w) &=& T_\bullet(z,w)+T_\circ(z,w)\\ T_\bullet(z,w) &=& zw\,(1+T_\circ(z,w))^2\\ T_\circ(z,w) &=& w\,(1+T_\bullet(z,w))^2\end{array}\right. .$$

The decomposition grammar of bicolored binary trees is directly translated into the following Boltzmann sampler $\Gamma T(z,w)$ for bicolored binary trees, based on Remark 2 below:

$\Gamma T(z,w)$: if Bern $\left(\frac{T_\bullet(z,w)}{T(z,w)}\right)$ return $\Gamma T_\bullet(z,w)$
else return $\Gamma T_\circ(z,w)$

$\Gamma T_\bullet(z,w)$: return $(\Gamma(1+T_\circ)(z,w), \mathcal{Z}_\bullet, \Gamma(1+T_\circ)(z,w))$

$\Gamma T_\circ(z,w)$: return $(\Gamma(1+T_\bullet)(z,w), \mathcal{Z}_\circ, \Gamma(1+T_\bullet)(z,w))$

$\Gamma(1+T_\circ)(z,w)$: if Bern $\left(\frac{1}{1+T_\circ(z,w)}\right)$ return $\varnothing$
else return $\Gamma T_\circ(z,w)$

$\Gamma(1+T_\bullet)(z,w)$: if Bern $\left(\frac{1}{1+T_\bullet(z,w)}\right)$ return $\varnothing$
else return $\Gamma T_\bullet(z,w)$

Remark 2. We consider here ordinary mixed classes, i.e., classes with two types of unlabelled atoms, a case not covered by the rules given in Figure 5, where the classes considered have both labeled atoms and unlabeled atoms. However, an easy adaptation of the proof of Proposition 4.6 ensures that the sampling rules for *Sum* and *Product* are also valid in the case of a class with two types of unlabelled atoms, i.e., a Boltzmann sampler for $\mathcal{C} = \mathcal{A} + \mathcal{B}$ is obtained by calling $\Gamma A(x,y)$ with probability $A(x,y)/C(x,y)$ and calling $\Gamma B(x,y)$ otherwise; and a Boltzmann sampler for $\mathcal{C} = \mathcal{A}\star\mathcal{B}$ consists of two independent calls to $\Gamma A(x,y)$ and $\Gamma B(x,y)$.

Boltzmann sampler for rooted irreducible dissections. The bijection between binary trees and irreducible dissections yields the following sampler for rooted irreducible dissections, where rnd(0,1) stands for a real number in $(0,1)$ taken uniformly at random:

$\Gamma I(z,w)$: repeat $u \leftarrow$ rnd$(0,1)$; max_size $\leftarrow \lfloor 1/u \rfloor$;
$\tau \leftarrow \Gamma T(z,w)$;
abort and restart as soon as #nodes$(\tau) + 2 >$ max_size
until (generation finishes)
return closure$(\tau,$ rnd$(1,2,3))$

The sampler is easily seen to be equivalent to the sampler "repeat $\tau \leftarrow \Gamma T(z,w)$ until Bern(1/(#nodes(τ)+2)); return closure(τ,rnd(1,2,3))", which is a Boltzmann sampler for irreducible dissections according to Theorem 3.4 page 99. The early abortion process yields a gain in complexity (the choice to

reject occurs before the entire binary tree is generated), which is crucial to obtain the complexity of generation for planar graphs, as stated in Theorem 4.3 and Theorem 4.4.

Boltzmann sampler for rooted 3-connected maps. As we have seen in Section 1.1, irreducible dissections are closely related to 3-connected maps via the angular mapping. Precisely, rooted 3-connected maps are in bijection with so-called rooted irreducible dissections with no internal path of length 3 connecting the root vertex to the opposite outer vertex; these dissections being called *undecomposable*. Given a rooted irreducible undecomposable dissection δ, we write $\pi(\delta)$ for the rooted quadrangulation obtained from δ by adding an outer (root) edge connecting the root vertex to the opposite outer vertex, and we write Primal(δ) for the primal map of $\pi(\delta)$. As we have seen in Section 1.1, Primal(δ) is 3-connected. From the discussion we derive the following Boltzmann sampler for rooted 3-connected maps:

$\Gamma M(z,w)$: repeat $\delta \leftarrow \Gamma I(z,w)$ until δ is undecomposable
return Primal(δ)

The Boltzmann sampler $\Gamma M(z,w)$ for rooted 3-connected maps is also a mixed Boltzmann sampler $\Gamma\overrightarrow{G_3}(z,w)$ for edge-rooted 3-connected planar graphs, according to the equivalence stated in Lemma 4.5.

4.4.2. *Boltzmann sampler for 2-connected planar graphs.* The next step of our sampler is to realise a Boltzmann sampler for 2-connected planar graphs from the Boltzmann sampler for edge-rooted 3-connected planar graphs obtained in Section 4.4.1. Precisely, we first describe a Boltzmann sampler for edge-rooted 2-connected planar graphs, and subsequently obtain a Boltzmann sampler for the derived class of 2-connected planar graphs, by using rejection techniques.

To generate edge-rooted 2-connected planar graphs, we use a well-known decomposition, due to Trakhtenbrot [**109**] and called *network-decomposition*, which ensures that an edge-rooted 2-connected planar graph can be assembled from edge-rooted 3-connected planar components. Precisely, Trakhtenbrot's decomposition deals with so-called *networks*, where a network is defined as a connected graph N with two distinguished vertices 0 and ∞ called *poles*, such that the graph N^* obtained by adding an edge between 0 and ∞ is a 2-connected planar graph. For the enumeration, the two poles are not counted in the size.

We rely on [**116**] for the description of Trakhtenbrot's decomposition. A *series-network* or s-network is a network made of at least 2 networks connected *in chain* at their poles, the ∞-pole of a network coinciding with the 0-pole of the following network in the chain. A *parallel network* or p-network is a network made of at least 2 networks connected *in parallel*, so that their respective ∞-poles and 0-poles coincide. A network N such that N^* is 3-connected and the poles are not adjacent is called a *pseudo-brick*. A *polyhedral network* or h-network is a network obtained by taking a pseudo-brick and substituting each edge e of the pseudo-brick by a network N_e (polyhedral networks establish a link between 2-connected and 3-connected planar graphs).

Proposition 4.7 (Trakhtenbrot). *Networks with at least 2 edges are partitioned into s-networks, p-networks and h-networks.*

Let us explain how to obtain a recursive decomposition involving the different families of networks. Let $\mathcal{D}$, $\mathcal{S}$, $\mathcal{P}$, and $\mathcal{H}$ be respectively the classes of networks, s-networks, p-networks, and h-networks. Let $D(z,y)$, $S(z,y)$, $P(z,y)$, $H(z,y)$ be

the associated mixed generating functions with respect to the number of non-pole vertices (variable z) and the number of edges (variable y). We recall that $\mathcal{L}$ is the family consisting only of the link-network, i.e., the graph with one edge connecting the two poles. Proposition 4.7 ensures that

$$\mathcal{D} = \mathcal{L} + \mathcal{S} + \mathcal{P} + \mathcal{H}.$$

An s-network can be uniquely decomposed into a non-s-network (the head of the chain) followed by a network (the trail of the chain), which yields

$$\mathcal{S} = (\mathcal{L} + \mathcal{P} + \mathcal{H}) \star \mathcal{Z} \star \mathcal{D},$$

where $\mathcal{Z}$ stands for the articulation-vertex between the head network and the trail network.

A p-network has a unique *maximal* parallel decomposition into a set of components that are not p-networks. Observe that we consider here graphs without multiple edges, so that at most one of these components is an edge. Whether there is one or no such edge-component yields

$$\mathcal{P} = \mathcal{L} \star \textsc{Set}_{\geq 1}(\mathcal{S} + \mathcal{H}) + \textsc{Set}_{\geq 2}(\mathcal{S} + \mathcal{H}).$$

By definition, the class of h-networks corresponds to a y-substitution of networks in pseudo-bricks. We write $\mathcal{G}_3$ for the family of labelled 3-connected planar graphs and denote by $G_3(z, w)$ the associated generating function with respect to vertices and edges. By definition, a pseudo-brick is an edge-rooted 3-connected plane graph. As a consequence,

$$\mathcal{H} = \overrightarrow{\mathcal{G}_3} \circ_y \mathcal{D}.$$

To summarize, Trakhtenbrot's decomposition yields the following decomposition grammar relating networks and edge-rooted 3-connected planar graphs:

$$\text{(N)} \qquad \begin{cases} \mathcal{D} = \mathcal{L} + \mathcal{S} + \mathcal{P} + \mathcal{H} \\ \mathcal{S} = (\mathcal{L} + \mathcal{P} + \mathcal{H}) \star \mathcal{Z} \star \mathcal{D} \\ \mathcal{P} = \mathcal{L} \star \textsc{Set}_{\geq 1}(\mathcal{S} + \mathcal{H}) + \textsc{Set}_{\geq 2}(\mathcal{S} + \mathcal{H}) \\ \mathcal{H} = \overrightarrow{\mathcal{G}_3} \circ_y \mathcal{D} \end{cases}$$

The decomposition grammar (N) is directly translated to a Boltzmann sampler $\Gamma D(z, y)$ for networks, using the sampling rules given in Figure 5. The only terminal nodes of the decomposition grammar are the classes $\mathcal{Z}$, $\mathcal{L}$ (which are explicit), and the class $\overrightarrow{\mathcal{G}_3}$. Thus, the sampler $\Gamma D(z, y)$ and the auxiliary samplers $\Gamma S(z, y)$, $\Gamma P(z, y)$, and $\Gamma H(z, y)$ are recursively specified in terms of $\Gamma\overrightarrow{G_3}(z, w)$, where $w = D(z, y)$.

Observe that each edge-rooted 2-connected planar graph different from the link graph gives rise to two networks, obtained respectively by including or not including the root-edge in the network. This yields the identity

$$(1 + \mathcal{L}) \star \overrightarrow{\mathcal{B}} \simeq (1 + \mathcal{D}), \tag{26}$$

where $\overrightarrow{\mathcal{B}}$ is the class of edge-rooted 2-connected planar graphs, $\mathcal{D}$ is the class of networks and $\mathcal{L}$ is the one-element class made of the link-graph. From that point, a Boltzmann sampler is easily obtained for the family of edge-rooted 2-connected planar graphs. Define a procedure AddRootEdge that adds an edge connecting the two poles 0 and ∞ of a network if they are not already adjacent, and roots the

obtained graph at the edge $(0,\infty)$ oriented from 0 to ∞. Equation (26) translates to the following Boltzmann sampler for $\overrightarrow{\mathcal{B}}$,

$\Gamma\overrightarrow{B}(z,y)$: $\gamma \leftarrow \Gamma(1+D)(z,y)$; AddRootEdge($\gamma$); return γ
$\Gamma(1+D)(z,y)$: if Bern $\left(\frac{1}{1+D(z,y)}\right)$ return link-graph else return $\Gamma D(z,y)$;

The last step is to obtain a Boltzmann sampler for derived 2-connected planar graphs from the Boltzmann sampler for edge-rooted 2-connected planar graphs (indeed, derived 2-connected planar graphs are the building blocks needed to construct connected planar graphs). This requires a simple rejection loop:

$\Gamma B'(z,y)$: repeat $\gamma \leftarrow \Gamma\overrightarrow{B}(z,y)$ until Bern $\left(\frac{V(\gamma)}{2E(\gamma)}\right)$;
DistributeLabels(γ); point a vertex at random; return γ

The probability $V(\gamma)/E(\gamma)$ of keeping the generated graphs turns the Boltzmann distribution for edge-rooted graphs into the Boltzmann distribution for vertex-rooted (i.e., derived) graphs.

4.4.3. *Boltzmann sampler for connected planar graphs.* To obtain a Boltzmann sampler for connected planar graphs, we translate a decomposition linking derived connected and derived 2-connected planar graphs to a Boltzmann sampler for derived connected planar graphs. Then, a further rejection step yields a Boltzmann sampler for connected planar graphs. The *block-decomposition* (see [**72**, p.10] for a detailed description) is specified as follows. Each derived connected planar graph can be uniquely constructed by composition in the following way: take a set of derived 2-connected planar graphs and attach them, by merging their marked vertices into a unique marked vertex. Then, for each unmarked vertex v of each 2-connected component, take a derived connected planar graph γ_v and merge the marked vertex of γ_v with v (this operation corresponds to an x-substitution). Writing $\mathcal{B}$ for the class of 2-connected planar graphs and $B(z,y)$ for its mixed generating function with respect to vertices and edges, the block-decomposition implies

$$\mathcal{C}' = \textsc{Set}\left(\mathcal{B}' \circ_x (\mathcal{Z} \star \mathcal{C}')\right), \qquad C'(x,y) = \exp(B'(xC'(x,y),y)). \tag{27}$$

The block-decomposition translates to the following Boltzmann sampler for derived connected planar graphs (based on the rules of Figure 5):

$\Gamma C'(x,y)$: $k \leftarrow \text{Pois}(B'(z,y))$ [with $z = xC'(x,y)$]
$\gamma \leftarrow (\Gamma B'(z,y), \ldots, \Gamma B'(z,y))$ {k independent calls}
merge the k components of γ at their marked vertices
for each unmarked vertex v of γ
  $\gamma_v \leftarrow \Gamma C'(x,y)$
  merge the marked vertex of γ_v with v
return γ.

A Boltzmann sampler for connected planar graphs is then simply obtained from $\Gamma C'(x,y)$ by using a rejection step so as to adjust the probability distribution:

$\Gamma C(x,y)$: repeat $\gamma \leftarrow \Gamma C'(x,y)$ until Bern $\left(\frac{1}{V(\gamma)}\right)$; return γ.

4.4.4. *Boltzmann sampler for planar graphs.* Let $\mathcal{G}$ be the class of planar graphs and $\mathcal{C}$ the class of connected planar graphs. Let $G(x,y)$ and $C(x,y)$ be the mixed

generating functions of $\mathcal{G}$ and $\mathcal{C}$ with respect to the number of vertices and edges. A planar graph is decomposed into the set of its connected components, yielding

$$\mathcal{G} = \textsc{Set}(\mathcal{C}), \qquad G(x,y) = \exp(C(x,y)), \tag{28}$$

which translates to the following Boltzmann sampler for planar graphs

$\Gamma G(x,y)$:	$k \leftarrow \text{Pois}(C(x,y))$
	return $(\Gamma C(x,y), \dots, \Gamma C(x,y))$ {k independent calls}

4.5. Deriving an efficient sampler. The preceding section has provided the complete description of a Boltzmann sampler for planar graphs. However more is needed to achieve the complexity stated in Theorems 4.3 and 4.4, as shown here.

4.5.1. *Size distribution.* In the last section, we have described a method to produce a mixed Boltzmann sampler $\Gamma G(x,y)$ for labelled planar graphs. In particular, $\Gamma G(x,1)$ is a Boltzmann sampler for labelled planar graphs, drawing two planar graphs with the same number of vertices with equal probability. For practical purpose, a *target size* n is chosen by the user (e.g. $n = 100,000$), and the sampler is required to return a random planar graph whose size is around n up to a few percents, or even exactly n. As a consequence, the size distribution of planar graphs output by $\Gamma G(x,y)$ has to be studied. Typically, we need to *tune* the real parameter x in order to ensure that the size distribution is concentrated around the target value n. However, this tuning operation does not always apply depending on the *singularity type* of $G(x)$.

Definition 2. Given $\alpha \in \mathbf{R}\backslash\mathbf{Z}_{\geq 0}$, a generating function $f(x)$ is said to be α-singular if the following expansion holds in a Δ-neighbourhood of its dominant singularity ρ (see [**43**] for technical conditions of such neighbourhoods),

$$f(x) \underset{x\to\rho}{=} P(x) + c_\alpha \left(1 - \frac{x}{\rho}\right)^\alpha + o\left(1 - \frac{x}{\rho}\right)^\alpha,$$

where $P(x)$ is a polynomial and c_α is a non-zero real value.

The following lemma, Theorem 6.3 of [**43**], ensures that the tuning operation mentioned above applies well when $f(x)$ is α-singular with $\alpha < 0$. We state it in a slightly more general version, extended to mixed Boltzmann samplers.

LEMMA 4.6. (Duchon *et al* [**43**]) *Let $\mathcal{F}$ be a mixed combinatorial class endowed with a Boltzmann sampler $\Gamma F(x,y)$. Let $F(x,y)$ be the mixed generating function of $\mathcal{F}$. Given $y > 0$, assume that the function $x \to F(x,y)$ is α-singular with $\alpha < 0$. For each integer n, define $x_n = \rho_G(y)\left(1 + \frac{\alpha}{n}\right)$, $\rho_G(y)$ being the radius of convergence of $x \to F(x,y)$. Let X_n be the random variable defined as the labelled size of an object output by $\Gamma F(x_n, y)$.*

Then, for each fixed tolerance-ratio $\epsilon > 0$,

$$\mathbf{P}\left(X_n \in [n(1-\epsilon), n(1+\epsilon)]\right) \to p_\epsilon \quad \text{as } n \to \infty,$$

where p_ϵ is a positive constant depending on ϵ, being of order ϵ as $\epsilon \to 0$: $p_\epsilon \sim_{\epsilon\to 0} \sigma \cdot \epsilon$ for some constant σ.

Moreover,

$$\mathbf{P}\left(X_n = n\right) \sim \frac{\sigma}{n} \quad \text{as } n \to \infty, \text{ for the same constant } \sigma > 0.$$

The following lemma indicates that we have to "derive 3 times" the Boltzmann sampler $\Gamma G(x,y)$ for planar graphs, so that the size distribution of the output gets the good behaviour stated in Lemma 4.6.

LEMMA 4.7 (Gimenez and Noy [**68**]). *Let $G(x,y)$ be the mixed generating function of labelled planar graphs. Then, for each $y>0$, the function $x \to G'''(x,y)$ is $(-1/2)$-singular.*

4.5.2. *Injecting the derivative operator in the decomposition of planar graphs.* The derivative operator is easily injected in the 5 constructions used to describe the decomposition of planar graphs,

$$(29) \qquad \left\{ \begin{array}{rcl} (\mathcal{A}+\mathcal{B})' & = & \mathcal{A}'+\mathcal{B}' \\ (\mathcal{A}\star\mathcal{B})' & = & \mathcal{A}'\star\mathcal{B}+\mathcal{A}\star\mathcal{B}' \\ \textsc{Set}_{\geq d}(\mathcal{A})' & = & \mathcal{A}'\star\textsc{Set}_{\geq d-1}(\mathcal{A}) \\ (\mathcal{A}\circ_x\mathcal{B})' & = & \mathcal{B}'\star\mathcal{A}'\circ_x\mathcal{B} \\ (\mathcal{A}\circ_y\mathcal{B})' & = & \mathcal{A}'\circ_y\mathcal{B}+\mathcal{B}'\star\overline{\mathcal{A}}\circ_y\mathcal{B}, \end{array} \right.$$

where we recall that $\overline{\mathcal{A}}$ stands for the y-derived class of $\mathcal{A}$. As a consequence, the derivative operator can be injected in the chain of decompositions, in order to assemble a Boltzmann sampler for triply derived planar graphs, as explained next.

Boltzmann samplers for derived 3-connected planar graphs. Given a bicolored binary tree τ, we denote by $|\tau|_\bullet$ the number of black nodes of τ and by $|\tau|$ the number of nodes of τ. Let $\Gamma T(z,w)$ be a Boltzmann sampler for bicolored binary trees and $\Gamma T'(z,w)$ be a Boltzmann sampler for the class $\mathcal{T}'$ of bicolored binary trees with a pointed black node that does not count in the size. In other words, $\Gamma T'(z,w)$ draws a bicolored binary tree τ with probability proportional to $|\tau|_\bullet z^{|\tau|_\bullet} w^{|\tau|}$. The class $\mathcal{T}'$ has a complete recursive decomposition, obtained by deriving the decomposition grammar of $\mathcal{T}$ with respect to z,

$$(30) \qquad \left\{ \begin{array}{rcl} \mathcal{T}' & = & \mathcal{T}'_\bullet+\mathcal{T}'_\circ \\ \mathcal{T}'_\bullet & = & \mathcal{T}'_\circ\star\mathcal{Z}_\bullet\star(1+\mathcal{T}_\circ)+(1+\mathcal{T}_\circ)\star(1+\mathcal{T}_\circ)+(1+\mathcal{T}_\circ)\star\mathcal{Z}_\bullet\star\mathcal{T}'_\circ \\ \mathcal{T}'_\circ & = & \mathcal{T}'_\bullet\star\mathcal{Z}_\circ\star(1+\mathcal{T}_\bullet)+(1+\mathcal{T}_\bullet)\star\mathcal{Z}_\circ\star\mathcal{T}'_\bullet, \end{array} \right.$$

which is translated to a Boltzmann sampler $\Gamma T'(z,w)$ using the sampling rules for Sum and Product.

The rooted version of the bijection between binary-trees and irreducible dissections (Theo. 3.4 p. 99) and Lemma 4.4 (rejection lemma) ensure that the following algorithms are Boltzmann samplers for the derived classes of rooted 3-connected maps up to order 2.

$\Gamma M'(z,w)$: repeat $\tau \leftarrow \Gamma T(z,w)$
until ($M \leftarrow \text{closure}(\tau, \text{rnd}(1,2,3))$ is undecomposable and Bern $\left(\frac{|\tau|_\bullet+1}{|\tau|+2}\right)$);
return M

$\Gamma\overline{M}(z,w)$: repeat $\tau \leftarrow \Gamma T(z,w)$
until ($M \leftarrow \text{closure}(\tau, \text{rnd}(1,2,3))$ is undecomposable and Bern $\left(\frac{3}{4}\frac{|\tau|+3}{|\tau|+2}\right)$);
return M

$\Gamma M''(z,w)$:repeat $\tau \leftarrow \Gamma T'(z,w)$
until ($M \leftarrow \text{closure}(\tau, \text{rnd}(1,2,3))$ is undecomposable and Bern $\left(\frac{|\tau|_\bullet+1}{|\tau|+2}\right)$);
return M

$\Gamma\overline{M}'(z,w)$: repeat $\tau \leftarrow \Gamma T'(z,w)$
until ($M \leftarrow \text{closure}(\tau, \text{rnd}(1,2,3))$ is undecomposable and Bern $\left(\frac{3}{8}\frac{(|\tau|_\bullet+1)(|\tau|+3)}{|\tau|_\bullet(|\tau|+2)}\right)$);
return M

$\Gamma\overline{\overline{M}}(z,w)$: repeat $\tau \leftarrow \Gamma T'(z,w)$
until ($M \leftarrow \text{closure}(\tau, \text{rnd}(1,2,3))$ is undecomposable and Bern $\left(\frac{1}{7}\frac{|\tau|+3}{|\tau|_\bullet}\right)$);
return M

Boltzmann samplers for derived 2-connected planar graphs. Starting from the 4-lines decomposition grammar (N) of networks and deriving two times, we obtain successively

$$\text{(N)} \quad \begin{cases} \mathcal{D} = \mathcal{L} + \mathcal{S} + \mathcal{P} + \mathcal{H} \\ \mathcal{S} = (\mathcal{L} + \mathcal{P} + \mathcal{H}) \star \mathcal{Z} \star \mathcal{D} \\ \mathcal{P} = \mathcal{L} \star \textsc{Set}_{\geq 1}(\mathcal{S} + \mathcal{H}) + \textsc{Set}_{\geq 2}(\mathcal{S} + \mathcal{H}) \\ \mathcal{H} = \overrightarrow{\mathcal{G}_3} \circ_y \mathcal{D} \end{cases}$$

$$\text{(N')} \quad \begin{cases} \mathcal{D}' = \mathcal{S}' + \mathcal{P}' + \mathcal{H}' \\ \mathcal{S}' = (\mathcal{P}' + \mathcal{H}') \star \mathcal{Z} \star \mathcal{D} + (\mathcal{L} + \mathcal{P} + \mathcal{H}) \star (\mathcal{D} + \mathcal{Z} \star \mathcal{D}') \\ \mathcal{P}' = \mathcal{L} \star (\mathcal{S}' + \mathcal{H}') \star \textsc{Set}(\mathcal{S} + \mathcal{H}) + (\mathcal{S}' + \mathcal{H}') \star \textsc{Set}_{\geq 1}(\mathcal{S} + \mathcal{H}) \\ \mathcal{H}' = \overrightarrow{\mathcal{G}_3}' \circ_y \mathcal{D} + \mathcal{D}' \star \overrightarrow{\mathcal{G}_3} \circ_y \mathcal{D} \end{cases}$$

$$\text{(N")} \quad \begin{cases} \mathcal{D}'' = \mathcal{S}'' + \mathcal{P}'' + \mathcal{H}'' \\ \mathcal{S}'' = (\mathcal{P}''+\mathcal{H}'') \star \mathcal{Z} \star \mathcal{D} + 2(\mathcal{P}'+\mathcal{H}') \star (\mathcal{D}+\mathcal{Z}\star\mathcal{D}') + (\mathcal{L}+\mathcal{P}+\mathcal{H}) \star (2\mathcal{D}'+\mathcal{Z}\star\mathcal{D}'') \\ \mathcal{P}'' = (\mathcal{L}\star(\mathcal{S}''+\mathcal{H}'') + (1+\mathcal{L}) \star (\mathcal{S}'+\mathcal{H}')^2) \star \textsc{Set}(\mathcal{S}+\mathcal{H}) + (\mathcal{S}''+\mathcal{H}'') \star \textsc{Set}_{\geq 1}(\mathcal{S}+\mathcal{H}) \\ \mathcal{H}'' = \overrightarrow{\mathcal{G}_3}'' \circ_y \mathcal{D} + 2\mathcal{D}' \star \overrightarrow{\mathcal{G}_3}' \circ_y \mathcal{D} + \mathcal{D}'^2 \star \overrightarrow{\overrightarrow{\mathcal{G}_3}} \circ_y \mathcal{D} + \mathcal{D}'' \star \overrightarrow{\mathcal{G}_3} \circ_y \mathcal{D} \end{cases}$$

In these three systems taken together, the only terminal nodes are the class $\overrightarrow{\mathcal{G}_3}$ and its derived classes up to order 2, which are isomorphic to the class $\mathcal{M}$ of rooted 3-connected maps and its derived classes up to order 2, via the identity $\mathcal{M} \simeq 2\overrightarrow{\mathcal{G}_3}$. In addition, we have obtained in Section 4.5.2 Boltzmann samplers for the derived classes of rooted 3-connected planar maps up to order 2. Hence, using the sampling rules of Figure 5 , the three systems for networks, derived networks, and doubly derived networks are translated respectively to Boltzmann samplers $\Gamma D(z,y)$, $\Gamma D'(z,y)$, and $\Gamma D''(z,y)$, which are recursively specified in terms of the Boltzmann samplers for $\mathcal{M}$ and its derived classes up to order 2, taken at $(z, w = D(z,y))$.

Then, Boltzmann samplers for derived edge-rooted 2-connected planar graphs are easily obtained. Indeed, Equation (26) yields successively

$$(1 + \mathcal{L}) \star \overrightarrow{\mathcal{B}} \simeq (1 + \mathcal{D}), \quad (1 + \mathcal{L}) \star \overrightarrow{\mathcal{B}}' \simeq \mathcal{D}', \quad (1 + \mathcal{L}) \star \overrightarrow{\mathcal{B}}'' \simeq \mathcal{D}'',$$

which translates to

$\Gamma\overrightarrow{B}(z,y)$: $\gamma \leftarrow \Gamma(1+D)(z,y)$; AddRootEdge($\gamma$); return γ
$\Gamma\overrightarrow{B}'(z,y)$: $\gamma \leftarrow \Gamma D'(z,y)$; AddRootEdge($\gamma$); return γ
$\Gamma\overrightarrow{B}''(z,y)$: $\gamma \leftarrow \Gamma D''(z,y)$; AddRootEdge($\gamma$); return γ

Finally, the rejection technique allows us to obtain Boltzmann samplers for derived 2-connected planar graphs from Boltzmann samplers for (derived) edge-rooted 2-connected planar graphs; the following samplers $\Gamma B'(z,y)$, $\Gamma B''(z,y)$, and $\Gamma B'''(z,y)$ are valid Boltzmann samplers for $\mathcal{B}'$, $\mathcal{B}''$, and $\mathcal{B}'''$ (after a call to DistributeLabels(γ)).

(31)
$\Gamma B'(z,y)$: repeat $\gamma \leftarrow \Gamma\overrightarrow{B}(z,y)$ until Bern $\left(\frac{V(\gamma)}{2E(\gamma)}\right)$; return γ
$\Gamma B''(z,y)$: repeat $\gamma \leftarrow \Gamma(\overrightarrow{B}+z\overrightarrow{B}')(z,y)$ until Bern $\left(\frac{V(\gamma)}{2E(\gamma)}\right)$; return γ
$\Gamma(\overrightarrow{B}+z\overrightarrow{B}')(z,y)$: if Bern $\left(\frac{\overrightarrow{B}(z,y)}{\overrightarrow{B}(z,y)+z\overrightarrow{B}'(z,y)}\right)$ return $\Gamma\overrightarrow{B}(z,y)$ else return $\Gamma\overrightarrow{B}'(z,y)$
$\Gamma B'''(z,y)$: repeat $\gamma \leftarrow \Gamma(2\overrightarrow{B}'+z\overrightarrow{B}'')(z,y)$ until Bern $\left(\frac{V(\gamma)}{E(\gamma)}\right)$; return γ
$\Gamma(2\overrightarrow{B}'+z\overrightarrow{B}'')(z,y)$: if Bern $\left(\frac{2\overrightarrow{B}'(z,y)}{2\overrightarrow{B}'(z,y)+z\overrightarrow{B}''(z,y)}\right)$ return $\Gamma\overrightarrow{B}'(z,y)$ else return $\Gamma\overrightarrow{B}''(z,y)$

Boltzmann samplers for derived connected planar graphs. Starting from Equation (27), the derivative rules (29) yield successively

(32)
$$\begin{cases} \mathcal{C}' = \textsc{Set}\left(\mathcal{B}' \circ_x (\mathcal{Z} \star \mathcal{C}')\right), \\ \mathcal{C}'' = (\mathcal{C}' + \mathcal{Z} \star \mathcal{C}'') \star \mathcal{B}'' \circ_x (\mathcal{Z} \star \mathcal{C}') \star \mathcal{C}', \\ \mathcal{C}''' = (2\mathcal{C}'' + \mathcal{Z} \star \mathcal{C}''') \star \mathcal{B}'' \circ_x (\mathcal{Z} \star \mathcal{C}') \star \mathcal{C}' + (\mathcal{C}' + \mathcal{Z} \star \mathcal{C}'')^2 \star \mathcal{B}''' \circ_x (\mathcal{Z} \star \mathcal{C}') \star \mathcal{C}' + (\mathcal{C}' + \mathcal{Z} \star \mathcal{C}'') \star \mathcal{B}'' \circ_x (\mathcal{Z} \star \mathcal{C}') \star \mathcal{C}''. \end{cases}$$

Using the sampling rules of Figure 5, these decompositions translate to Boltzmann samplers $\Gamma C'(x,y)$, $\Gamma C''(x,y)$, and $\Gamma C'''(x,y)$, which are recursively specified in terms of the Boltzmann samplers $\Gamma B'''(z,y)$, $\Gamma B''(z,y)$, and $\Gamma B'(z,y)$, where $z = xC'(x,y)$.

Boltzmann samplers for derived planar graphs. Starting from $\mathcal{G} = \textsc{Set}(\mathcal{C})$, the derivative rules (29) yield successively

(33)
$$\begin{cases} \mathcal{G} &= \textsc{Set}(\mathcal{C}), \\ \mathcal{G}' &= \mathcal{C}' \star \mathcal{G}, \\ \mathcal{G}'' &= \mathcal{C}'' \star \mathcal{G} + \mathcal{C}' \star \mathcal{G}', \\ \mathcal{G}''' &= \mathcal{C}''' \star \mathcal{G} + 2\mathcal{C}'' \star \mathcal{G}' + \mathcal{C}' \star \mathcal{G}''. \end{cases}$$

Again, using the sampling rules of Figure 5, these decompositions translate to Boltzmann samplers $\Gamma G'(x,y)$, $\Gamma G''(x,y)$, and $\Gamma G'''(x,y)$, which are specified in terms of the Boltzmann samplers $\Gamma C'''(x,y)$, $\Gamma C''(x,y)$, $\Gamma C'(x,y)$, and $\Gamma C(x,y)$. (The Boltzmann sampler $\Gamma C(x,y)$ has already been obtained from $\Gamma C'(x,y)$ using rejection).

The complete algorithmic scheme, from binary trees to triply derived planar graphs, is summarized in Figure 6 and Figure 7.

4.5.3. *Samplers according to the number of vertices.* The random sampler of planar graphs we use is the "triply derived" Boltzmann sampler $\Gamma G'''(x_n,1)$ with the value $x_n = \rho_G\left(1-\frac{1}{2n}\right)$ tuned as indicated in Lemma 4.6, ρ_G being the radius of convergence of $G(x,1)$. The exact-size sampler is

$\mathfrak{A}_n$: repeat $\gamma \leftarrow \Gamma G'''(x_n,1)$ until $V(\gamma) = n$; return γ.

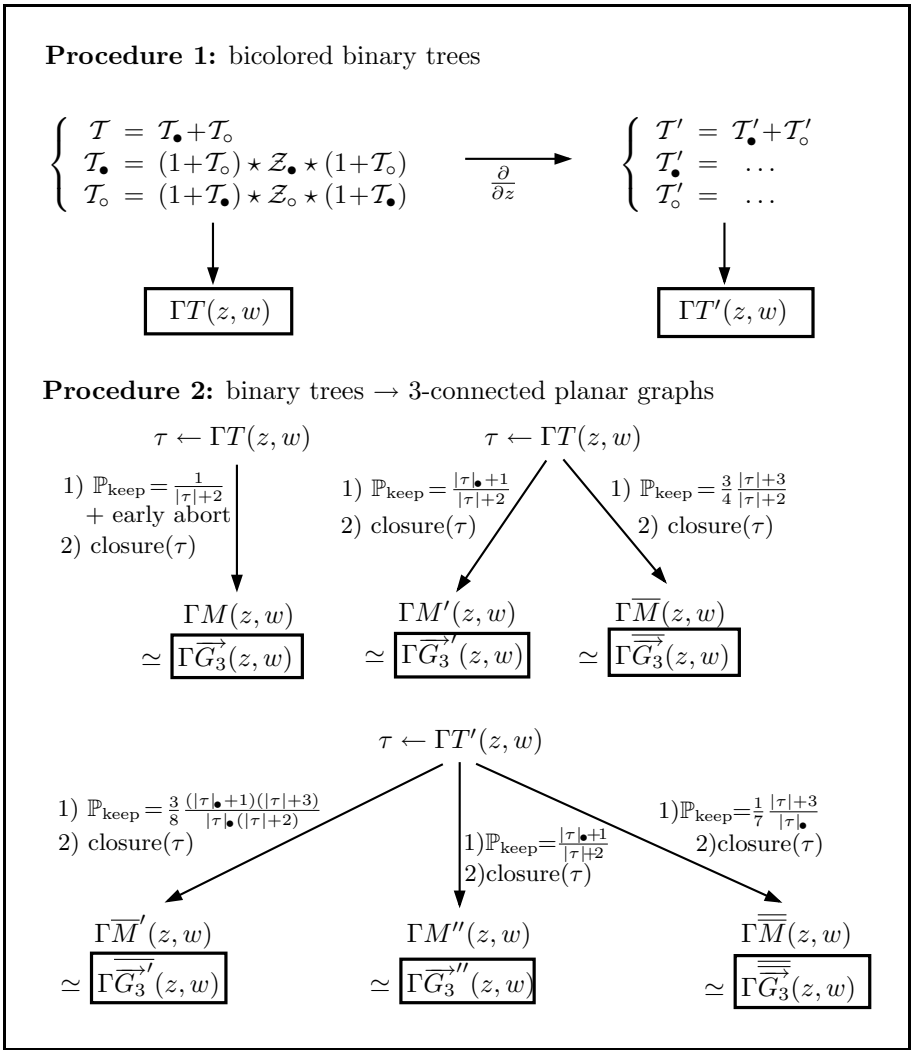


FIGURE 6. The algorithmic scheme producing Boltzmann samplers for 3-connected planar graphs from Boltzmann samplers for bicolored binary trees.

For any $\epsilon > 0$, the approximate-size sampler is

$\mathfrak{A}_{n,\epsilon}$: repeat $\gamma \leftarrow \Gamma G'''(x_n, 1)$ until $V(\gamma) \in [n(1-\epsilon), n(1+\epsilon)]$; return γ.

4.5.4. *Samplers according to the numbers of vertices and edges.* For any $y > 0$, we denote by $\rho_G(y)$ the radius of convergence of $x \to G(x,y)$. Let $\mu(y)$ be the function defined as

$$\mu(y) := -y\frac{\mathrm{d}\rho_G}{\mathrm{d}y}(y)/\rho_G(y).$$

It has been shown by Giménez and Noy [**68**] that the function $\mu(y)$ is strictly increasing on $(0, +\infty)$, with $\lim \mu(y) = 1$ as $y \to 0$ and $\lim \mu(y) = 3$ as $y \to +\infty$. As a consequence, $\mu(y)$ has an inverse function $y(\mu)$ defined on $(1,3)$. We define $x_n(\mu) := \rho_G(y(\mu))(1 - \frac{1}{2n})$. The exact size sampler we propose is

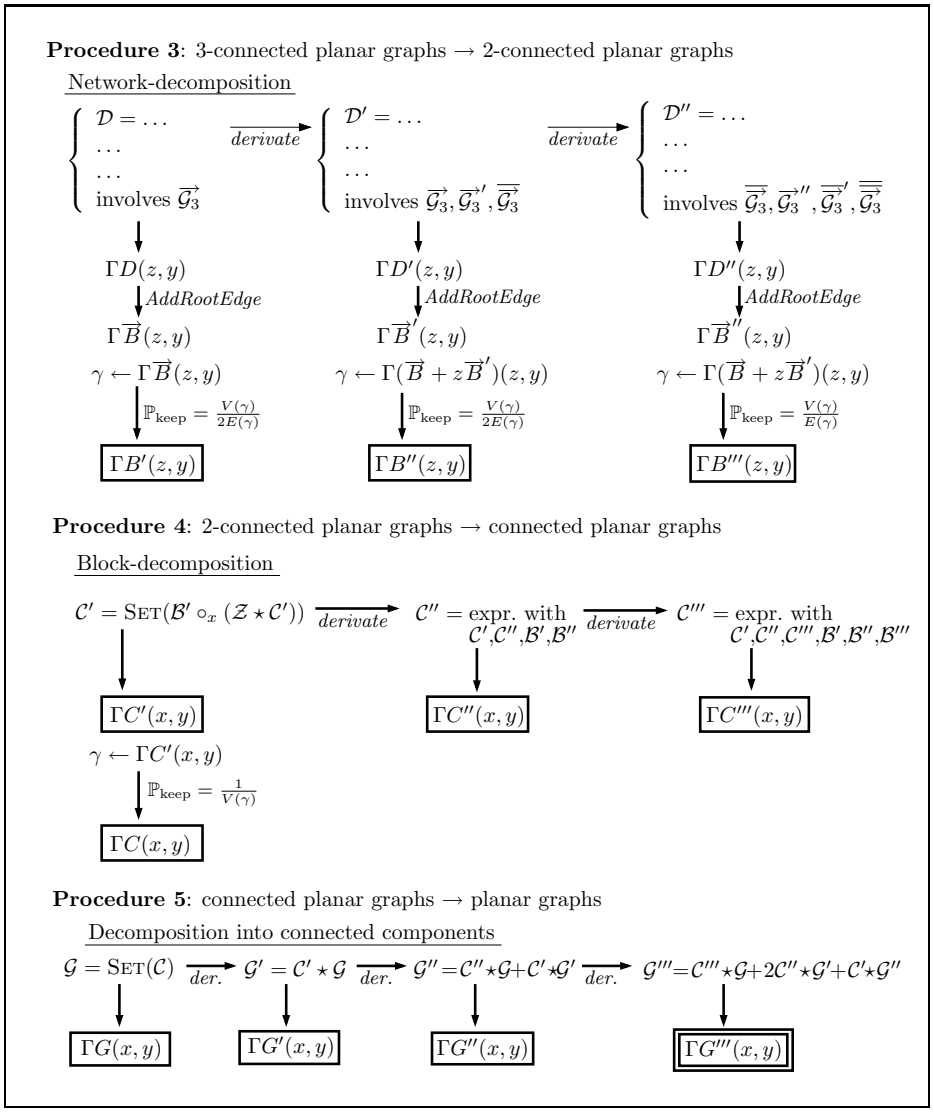


FIGURE 7. The algorithmic scheme producing a Boltzmann sampler for triply derived planar graphs from Boltzmann samplers for 3-connected planar graphs.

$\mathfrak{A}_{n,\mu}$: repeat $\gamma \leftarrow \Gamma G'''(x_n(\mu), y(\mu))$ until $(V(\gamma) = n$ and $E(\gamma) = \lfloor \mu n \rfloor)$; return γ.

For any $\epsilon > 0$, the approximate-size sampler is

$\mathfrak{A}_{n,\mu,\epsilon}$: repeat $\gamma \leftarrow \Gamma G'''(x_n(\mu), y(\mu))$
until $(V(\gamma) \in [n(1-\epsilon), n(1+\epsilon)]$ and $\frac{E(\gamma)}{V(\gamma)} \in [\mu(1-\epsilon), \mu(1+\epsilon)])$;
return γ.

4.6. Implementation and experimental results. We have realised a full implementation of the random samplers for planar graphs. First we evaluated

with good precision—typically 20 digits are enough for our purpose—the generating functions of the families of planar graphs that intervene in the decomposition: general, connected, 2-connected, 3-connected, pointed up to 3 times. The calculations have been carried out in Maple using the analytic expressions of Giménez and Noy for the generating functions. We performed the evaluations for values of the parameter x associated with a bunch of reference target sizes in logarithmic scale, $n = \{10^2, 10^3, 10^4, 10^5, 10^6\}$. From the values of the generating functions we computed the vectors of real values that are associated to the random choices to be performed during the generation, e.g., a Poisson law vector with parameter $C(x)$ (the GF of connected planar graphs) is used for drawing the number of connected components of the graph.

The second step has been the implementation of the random sampler in Java. To build the graph all along the generation process, it proves more convenient to manipulate planar maps rather than planar graphs. The advantage is that the graph to be generated will be equipped with an explicit (arbitrary) planar embedding. Thus if the graph generated is to be drawn in the plane, we do not need to call the rather involved algorithms for embedding a planar graph. Moreover, planar maps have the advantage that they are suitably manipulated using the so-called *half-edge structure*, where each half-edge occupies a memory block containing a pointer to the opposite half-edge along the same edge and to the next half-edge in ccw order around the incident vertex. Using the half-edge structure, it proves very easy to implement in cost $O(1)$ all primitives used for building the graph—typically, merging two components at a common vertex or edge. Doing this, the actual complexity of implementation corresponds to the complexity of the random samplers as stated in Theorem 4.3 and Theorem 4.4: linear for approximate-size sampling and quadratic for exact-size sampling. In practice, generating a graph of size of order 10^5 takes a few seconds on a standard PC.

Experimentations. The good complexity of our random samplers allows us to observe statistical properties of parameters on very large random planar graphs—in the range of sizes 10^5—where the asymptotic regime is already visible. We focus here on the experimental study of parameters that are known or expected to be concentrated around a limit value.

Number of edges. First we have checked that the random variable X_n that counts the number of edges in a random connected planar graph with n vertices is concentrated. Precisely, Giménez and Noy have proved that $Y_n := X_n/n$ converges in law to a constant $\mu \approx 2.213$, with gaussian fluctuations of magnitude $1/\sqrt{n}$. Figure 8 shows in ordinate the ratio $\#(edges)/\#(vertices)$ for a collection of 80 random connected planar graphs, obtained by calling repeatedly the triply pointed Boltzmann sampler $\Gamma C^{\bullet\bullet\bullet}(x)$ for a parameter x close to the singularity, and keeping the generated graphs of size at least 10^4. The observed ratios are concentrated around the horizontal line $y = \mu$, showing good agreement with the convergence results of Giménez and Noy.

Degrees of vertices. Another parameter of interest is the distribution of the degrees of vertices in a random planar graph. For a planar graph γ with n vertices, we denote by $N_n^{(k)}(\gamma)$ the number of vertices of γ that have k neighbours. Accordingly, $p_k := n_k/n$ is the proportion of vertices of degree k. It is known from Giménez and Noy that, for $k = 1, 2$, p_k converges in law to an explicit constant. Figure 9 shows in abscissa the parameter k and in ordinate the value of p_k for a collection of 80

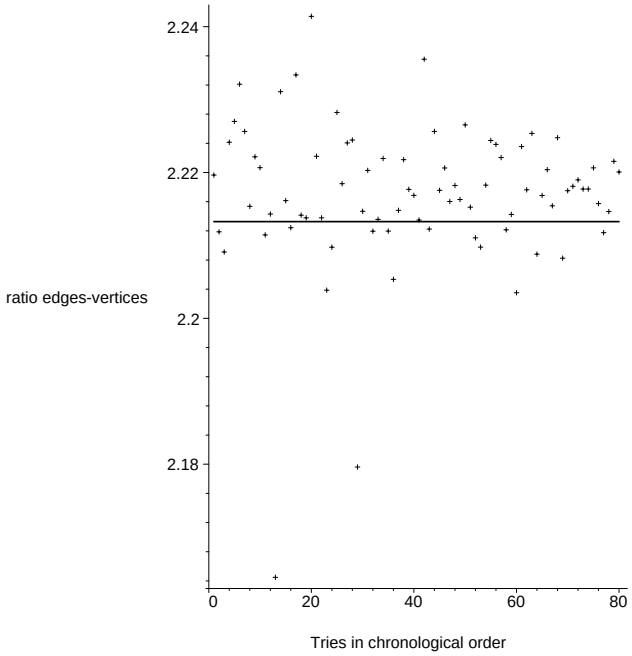


FIGURE 8. Ratio edges/vertices observed on a collection $G_1, \ldots, G_{80}$ of 80 random connected planar graphs of size at least 10^4; each graph G_i yields a point at coordinates $(i, \mathrm{Rat}(G_i)$, where $\mathrm{Rat}(G) := \#edges(G)/\#vertices(G)$.

connected planar graphs that have at least 10^4 vertices. Hence, each abscissa-line $x = k$ is occupied by 80 points whose ordinates correspond to the values taken by p_k for each of the graphs. As we can see, for k small—typically $k << \log n$—the values of p_k are concentrated around a constant. This leads us to the following conjecture.

CONJECTURE 4.1. *For $k \geq 1$, let $N_n^{(k)}$ be the random variable for the number of vertices of degree k in a random planar graph with n vertices taken uniformly at random. Then $N_n^{(k)}/n$ converges to an explicit constant π_k. In addition $\sum_k \pi_k = 1$.*

Notice that the conjecture is on planar graphs whereas experimentations are on connected planar graphs. However, from the works of Giménez and Noy, a random planar graph consists of a huge connected component, the total size of the other components having expectation of order $O(1)$. Thus, statistical properties like that stated in Conjecture 4.1 should be the same for random planar graphs as for random connected planar graphs.

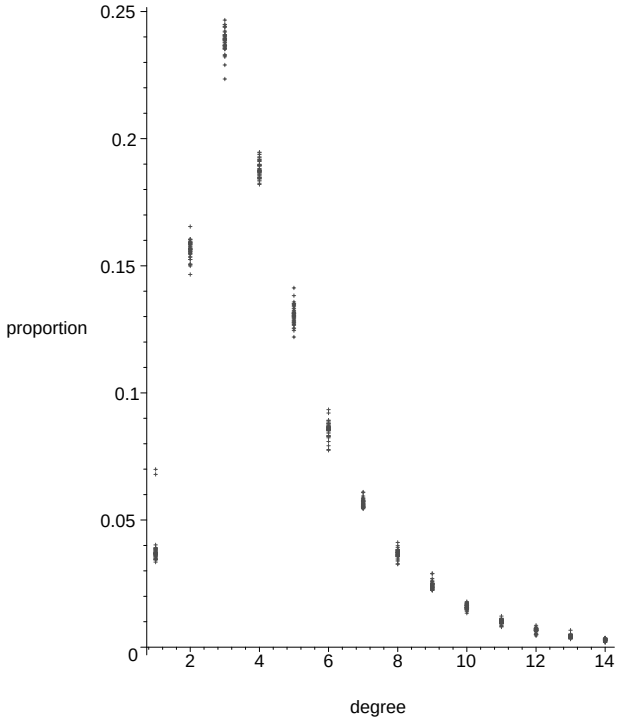


FIGURE 9. The degrees of vertices observed on a collection $G_1, \ldots, G_{80}$ of 80 random connected planar graphs of size at least 10^4. Each graph G yields points at coordinates $(1, \pi_1(G)), (2, \pi_2(G)), \ldots, (m, \pi_m(G))$, where m is the maximal degree of G and, for $1 \leq k \leq m$, $\pi_k(G)$ is the proportion of vertices of G that have degree k.

Conclusion. The bijections presented in Chapter 3 yield efficient algorithms to perform encoding and random generation of planar maps. Encoding is optimal in the sense that the length of the code matches asymptotically the binary entropy (defined as $n^{-1} \log_2(|\mathcal{C}_n|)$) of the class of maps considered. Random generation has linear time complexity, so that maps of very large size (in the range of 10^6) are readily generated.

More surprisingly, our bijective generator for 3-connected maps —combined with the framework of Boltzmann samplers— makes it possible to develop a uniform random generator for planar *graphs* (different from maps, as no explicit planar embedding is attached) with very low polynomial complexity. Fixed-size uniform sampling has expected quadratic complexity. More importantly perhaps, approximate-size sampling has expected linear time complexity, making it possible to generate planar graphs of very large size uniformly at random. In practice, planar graphs of size in the range of 10^5 are readily generated, whereas the best previously known uniform planar graph sampler was limited to sizes in the range of 10^2.

CHAPTER 5

Straight-line drawing

Introduction. Efficient tools of visualisation are very helpful to study the structure of complex systems (e.g., metabolism, web graph, social networks, neural cortex). The common feature is that these systems are made of dependencies (e.g. hyperlinks), so that they are naturally represented as *graphs*. This has motivated the emergence of a branch of computer science, called *graph drawing*, which addresses the issue of displaying graphs so as to satisfy various constraints and aesthetic criteria. Graph drawing is an active research topic, which finds many applications such as circuit layout, floor planning, or diagram display [**2**]. In general, a graph drawing algorithm takes as input a combinatorial description of the graph and outputs a drawing (in general in the plane). Classical examples are planar drawings (drawings such that no two edges cross), orthogonal drawings (the edges are either vertical or horizontal), and upward drawings, where a hierarchy is geometrically displayed in a down-to-top direction.

We focus in this chapter on planar drawings. Clearly planar drawings are preferable to have a clear visualisation. Moreover, deploying complex systems such as circuit layouts or road maps often asks for a representation avoiding edge-crossings, as these represent collisions. This requires a planarization step (e.g. adding vertices at crossings), or the decomposition of the graph into several planar layers. Once the planarization is done, the task comes down to drawing planar graphs with no edge-crossing. A very natural type of planar embedding are *straight-line drawings*, i.e., crossing-free drawings with vertices represented by points and edges represented by segments. Notice that the definition of planar graphs states the existence of a crossing-free embedding with edges represented as smooth curves. Hence the question of feasibility of a straight-line drawing is non-trivial. The answer is positive, the first proof being due to Fary [**45**], Wagner [**115**], and Stein[**106**] (this is also a consequence of Steinitz's theorem on the realizability of 3-connected planar graphs as polytopes). This chapter introduces and analyses several straight-line drawing algorithms for different families of planar maps, like triangulations and quadrangulations.

Results obtained in this chapter. The main contribution of this chapter is a new straight-line drawing algorithm based on transversal structures. Like the algorithm for 3-connected maps based on Schnyder woods (recalled in Section 1), the principle is to embed the vertices using simple face-counting operations. We first describe the algorithm in the general case. In a similar way as for Schnyder woods, the grid size can be reduced by deleting specific edges, yielding a more compact drawing. Then the algorithm is adapted to the family of irreducible triangulations and to the family of quadrangulations. The semi-perimeter is $n-1$ at most (i.e., in the worst case) if the graph has n vertices, equalling the best previously known algorithms: [**74, 89**] for irreducible triangulations and [**10**] for quadrangulations. Then, in Section 3, we

focus on the probabilistic analysis of the grid size, i.e., we study the distribution of the grid size of a planar map drawn uniformly at random from a given family and with a fixed size. This approach, which has been initiated by Bonichon *et al* [**18**], is original in the context of graph drawing, which usually focuses on a worst-case bounds. The analysis we develop is also interesting on its own, as it combines structural properties studied in Chapter 1, the bijections described in Chapter 3, and modern tools of Analytic Combinatorics such as the quasi-power theorem. With all these ingredients, we come up with a precise analysis of the face-counting drawing algorithms based on Schnyder woods and transversal structures, showing that the grid is always asymptotically squared and is significantly smaller than in the worst case. For instance, the grid size of a random irreducible triangulation with n vertices converges in law to $11n/27 \times 11n/27$, while the worst case grid size is $n/2 \times n/2$.

Motivations. Notice that the classical display devices (screen, squared paper) are finite regular grids, the size of the grid corresponding to the display resolution. Due to the limited resolution, it is thus crucial for straight-line drawing algorithms to produce a small grid size with respect to the numbers of vertices of the map. Notice that the first straight-line drawing algorithms, like the spring embedding of Tutte [**114**], had no guarantee on the grid area; the implementation requires delicate real-arithmetic computations to calculate the coordinates, with no guarantee on the minimal distance between two vertices. De Fraysseix *et al* [**39**] and Schnyder [**102**] have independently introduced the first algorithms embedding planar graphs with n vertices on a grid of area $\mathcal{O}(n^2)$. The algorithmic approaches are quite different: a) purely iterative for De Fraysseix *et al*, where the vertices are treated in a specific order and the drawing is globally updated at each step [**39, 77, 89**]; b) one-shot type for Schnyder, where the planar map is first endowed with a combinatorial structure (e.g. Schnyder woods for triangulations), which is used to compute the coordinates of vertices using some face-counting operations [**16, 46, 102**].

Our algorithm based on transversal structures is of the one-shot type and is also based on face-counting operations. The algorithms of this type have the advantage that the coordinates of vertices are computed independently, so that they are easier to implement and to perform on a piece of paper. In addition, the grid size can be expressed in terms of combinatorial parameters of the graph, making it possible to perform a precise grid-size analysis for a typical random instance.

1. Drawing using Schnyder woods

The combinatorial structure of Schnyder woods gives rise to an elegant straight-line drawing algorithm, introduced by Schnyder [**102**] for triangulations and subsequently extended independently by Felsner [**46**] and Di Battista *et al* [**3**] to 3-connected maps. The idea is to use the acyclicity property of Schnyder woods to embed the vertices in a barycentric way, using face-counting operations.

Then, as explained in Section 1.2, the grid size can be further reduced using specific edge-deletion operations that decrease the number of inner faces. This improved algorithm has been developed by Bonichon *et al* [**16**], extending on edge-deletion principles appearing in an algorithm of Zhang and He that performs compact straight-line drawing of triangulations [**118**]. The grid size obtained in this way is $(n-1-\max(1,\Delta)\times(n-1-\max(1,\Delta))$, where $\Delta \geq 0$ is an explicit combinatorial parameter of the graph. In addition, the algorithm has the nice feature that the faces are (weakly) convex. An algorithm by Chrobak and Kant [**32**] (not described

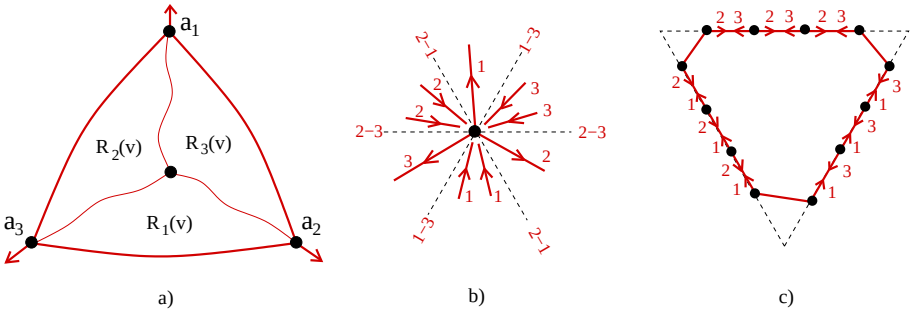


FIGURE 1. The three areas associated with a vertex (Fig.a), the configuration of embedded edges incident to a vertex (Fig.b) and to a face (Fig.c).

here) also has these features, i.e., convexity and worst-case grid $(n-2)\times(n-2)$, but the difference between $n-2$ and the actual grid width is not specified in terms of an explicit combinatorial parameter.

1.1. Generic algorithm. Consider a suspended internally 3-connected map $M=(V,E)$ endowed with a Schnyder wood. Recall that, for $i\in\{1,2,3\}$, the graph T_i induced by the edges carrying label i is a spanning tree rooted at the outer vertex a_i, see Figure 12 page 37. Moreover, for each inner vertex v of M, the paths $P_1(v), P_2(v), P_3(v)$ going from v to the root in each tree T_1, T_2, T_3 are disjoint after leaving v. Thus, the three paths delimit three areas $R_1(v), R_2(v), R_3(v)$, where $R_i(v)$ is delimited by $P_{i-1}(v)$ and $P_{i+1}(v)$, see Figure 1(a).

LEMMA 5.1 ([**3, 46**]). *Let $e=\{v,v'\}$ be an edge of M. Then the following inclusion relations are satisfied.*

- *If e is simply oriented from v to v' with label i, then*
$$R_{i+1}(v')\subsetneq R_{i+1}(v),\quad R_{i-1}(v')\subsetneq R_{i-1}(v),\quad R_i(v)\subsetneq R_i(v').$$
- *If e is bi-oriented with label $i+1$ at v and label $i-1$ at v', then*
$$R_i(v)=R_i(v'),\quad R_{i+1}(v)\subsetneq R_{i+1}(v'),\quad R_{i-1}(v')\subsetneq R_{i-1}(v').$$

Let $f_1(v)$, $f_2(v)$, $f_3(v)$ be the numbers of inner faces in $R_1(v)$, $R_2(v)$, $R_3(v)$, respectively. Observe that $f_1(v)+f_2(v)+f_3(v)$ is equal to f, the number of inner faces of M. We associate to v the barycentric triple
$$x_1(v):=\frac{f_1(v)}{f},\quad x_2(v):=\frac{f_2(v)}{f},\quad x_3(v):=\frac{f_3(v)}{f}.$$
The coordinates $x_i(v)$ satisfy $x_1(v)+x_2(v)+x_3(v)=1$, so that they give rise in a natural way to the following barycentric embedding of the map M,

(1) place the outer vertices a_1, a_2, a_3 so as to form a clockwise triangle,
(2) place each inner vertex v of M at the barycentre $x_1(v)\cdot a_1+x_2(v)\cdot a_2+x_3(v)\cdot a_3$,
(3) embed all edges of M as segments.

The obtained embedding is easily shown to be planar, based on Lemma 5.1. Indeed, the inclusion relations stated in the lemma ensure that, around each vertex v, bi-oriented edges are on a line $\{x_i=x_i(v)\}$ and simply oriented edges are in one particular sextant delimited by the three lines parallel to (a_1,a_2), (a_2,a_3),

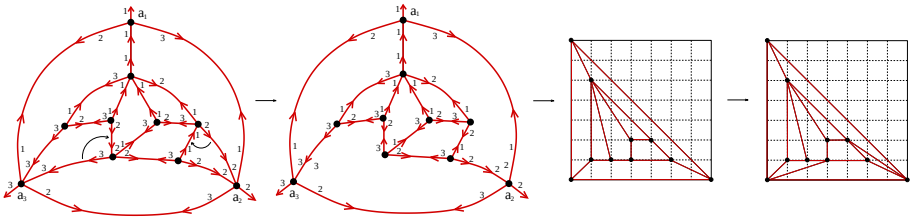


FIGURE 2. Embedding a suspended internally 3-connected map using Schnyder woods and face-counting operations.

and (a_3, a_1), respectively (see Figure 1(b)). Precisely, 1) if an edge $e = \{v, v'\}$ is simply oriented with label i from v to v', then $\{x_i(v') > x_i(v),\ x_{i-1}(v') < x_{i-1}(v),\ x_{i+1}(v') < x_{i+1}(v)\}$, so that e is in the open sextant delimited by $x_{i-1} = x_{i-1}(v)$ and $x_{i+1} = x_{i+1}(v)$; 2) if an edge $e = \{v, v'\}$ is bi-oriented with label i at v and label j at v', then e is on the half-line $\{x_k = x_k(v), x_i \geq x_i(v)\}$, where $\{k\} = \{1, 2, 3\} \backslash \{i, j\}$.

In particular, a judicious placement of a_1, a_2, a_3 yields a straight-line drawing of M on a regular square grid of width f, with f the number of inner faces of M:

SCHNYDERDRAW(M):
(1) Take a regular grid $f \times f$.
(2) Place the outer vertices a_1 at $(0, f)$, a_2 at $(f, 0)$, and a_3 at $(0, 0)$.
(3) Place each inner vertex of M at the point $(f_2(v), f_1(v))$.

PROPOSITION 5.1. [**3, 46, 102**] *Let M be a suspended map endowed with a Schnyder wood. Let f be the number of inner faces of M. Then the procedure* SCHNYDERDRAW(M) *outputs a straight-line of M on the square grid $f \times f$, with the property that all faces are weakly convex.*

1.2. Improvement based on edge-deletions. Recall that an important issue in graph drawing is to keep the size of the grid as small as possible. By construction, the algorithms based on Schnyder woods produce embeddings on a square grid of width equal to the number of inner faces of the map. A nice idea, introduced by Bonichon *et al* [**16**], is to delete edges of the map while maintaining a Schnyder wood. In this way, the number of inner faces (hence the width of the grid) is decreased. Then, the reduced map is embedded by calling the algorithm SCHNYDERDRAW. The difficulty is to specify edge-deletion operations so that the embedding remains planar after re-inserting the deleted edges.

The edge-deletions are formalised as *merges*. Define a *clockwise merge* as the operation represented on Figure 3(a), i.e., find two simply oriented clockwise-consecutive edges e_1 and e_2 arond a vertex v, with e_1 outgoing and e_2 ingoing; then delete e_1 and carry the orientation and label of e_1 on the second half-edge of e_2. A clockwise-merge gives rise to a 3-connected planar map with one inner face less. In addition, as shown in [**16**], the induced orientation and labelling of the half-edges is a Schnyder wood, which is more easily verified by working on the derived map and checking that a merge preserves an α_3-orientation, see Figure 3(b).

COMPACTSCHNYDERDRAW(M):
(1) Compute the minimal Schnyder wood of M using the algorithm of Section 4.

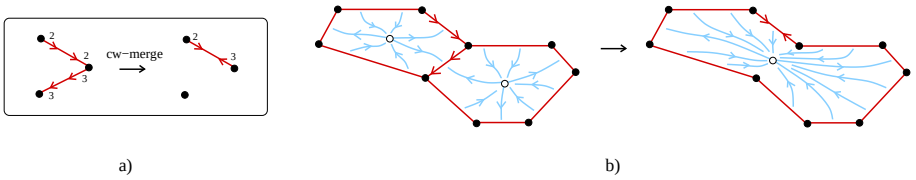


FIGURE 3. A clockwise merge (top), and the effect of a clockwise merge on the derived map (down).

(2) Starting from the graph M, perform a maximal sequence of clockwise-merges, yielding a "reduced" graph $\widehat{M}$ endowed with the Schnyder wood induced by the merges.
(3) Call SCHNYDERDRAW($\widehat{M}$).
(4) Reinsert the edges of $M \setminus \widehat{M}$ as segments.

Counter-clockwise faces. As explained in the remark page 39 (see also Figure 1(c)), the contour of an inner face f of M contains three particular edges $e_1(f)$, $e_2(f)$ and $e_3(f)$ such that the boundary path between $e_i(f)$ and $e_{i+1}(f)$ has only bi-oriented edges with labels i and $(i+1)$. A face f is called *counter-clockwise* if there is a ccw half-edge in each label on the boundary. If the Schnyder wood is minimal, it is easily shown that a face f is counterclockwise iff the edges $e_1(f)$, $e_2(f)$ and $e_3(f)$ are either simply oriented ccw or bi-oriented.

THEOREM 5.1 (Bonichon *et al* [**16**]). *For each suspended internally 3-connected map M,* COMPACTSCHNYDERDRAW(M) *outputs a convex straight-line drawing of M on a grid $(n-1-\Delta) \times (n-1-\Delta)$, where Δ is the number of counter-clockwise faces of M.*

Proof. The embedding $\widehat{E}$ given by SCHNYDERDRAW($\widehat{M}$) is planar and convex, according to Proposition 5.1. Hence the edges re-inserted at the end are chords of weakly convex faces. In addition, the set of re-inserted chords in each face are combinatorially organised in a non-crossing way (because M is a planar map). Thus the only point is to show that the extremities of a re-inserted edge are not connected by a straight path along an inner face f of $\widehat{M}$. This is proved in [**16**, Theo 5], based mainly on the generic configuration of an embedded face (Figure 1(c)) and on a study of the configurations of re-inserted edges.

The formula for the grid size relies on the following property, which easily follows from the proof of [**16**, Theo 6]: if a suspended graph is endowed with its minimal Schnyder wood, then the length of any maximal sequence of clockwise merges on M is equal to $f - n + \Delta + 1$, where Δ is the number of counterclockwise faces of M. Hence, the width of the square grid of the embedding output by COMPACTSCHNYDERDRAW(M) is equal to $n - 1 - \Delta$. □

Let us mention that the algorithm stated above outputs a grid size $(n-1) \times (n-1)$ in the worst case. The authors of [**16**] explain how to slightly twist the drawing when $\Delta = 0$ so as to obtain a $(n-2) \times (n-2)$ in the worst case. The worst-case grid size $(n-2) \times (n-2)$ is the best currently known, and has already been attained by other algorithms, first by Schnyder using vertex counting (instead of face counting) arguments [**102**], and by Chrobak and Kant [**32**], improving on the principles of the shift-based procedure by De Fraysseix *et al* [**39**]. It is in fact

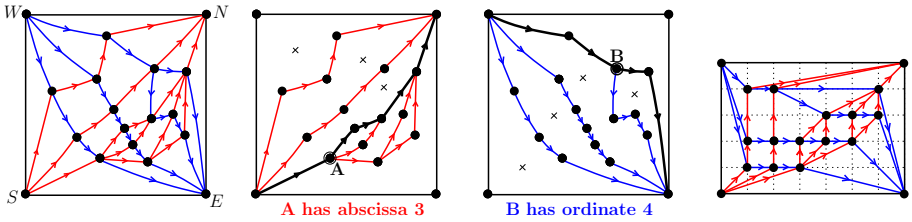


FIGURE 4. Example of execution of TRANSVERSALDRAW(M).

conjectured that any planar map with n vertices and no loop nor multiple edge has a straight-line drawing on a grid $\lfloor 2n/3 - 1 \rfloor \times \lfloor 2n/3 - 1 \rfloor$.

2. Drawing using transversal structures

In a similar way as Schnyder woods, transversal structures give rise to a simple straight-line drawing algorithm based on face-counting operations, which is our main contribution in this chapter. The idea is to use the red edges to associate an abscissa and the blue edges to associate an ordinate to each vertex. The algorithm applies to embed irreducible triangulations (Section 2). More surprisingly, it also applies to embed quadrangulations (Section 2.4), even if quadrangulations do not admit transversal structures.

2.1. Generic algorithm. Let us first describe the generic straight-line drawing algorithm TRANSVERSALDRAW. Consider a planar map M with quadrangular outer face and endowed with a transversal structure. The *red map* (blue map) of M is the map M_r (M_b) obtained by deleting all blue edges (red edges, respectively) of M, see Figure 15(c). Given an inner vertex v of M, the *rightmost ingoing red path* of v is the path $P_S(v) = (v_0 = v, v_1, \ldots, v_k = S)$ from v to S where, for each $i \in [0..k-1]$, (v_{i+1}, v_i) is the rightmost ingoing red edge of v_i, i.e., the unique ingoing edge of v_i whose ccw consecutive edge around v_i is outgoing. The *leftmost outgoing red path* of v is the path $P_N(v) = (v_0 = v, v_1, \ldots, v_l = N)$ where, for $i \in [0..l-1]$, (v_i, v_{i+1}) is the leftmost outgoing red edge of v_i. The *separating red path* $\mathcal{P}_\mathrm{r}(v)$ of v is the concatenation of $P_N(v)$ and $P_S(v)$. Hence, $\mathcal{P}_r(v)$ is a path from S to N. We define similarly the *separating blue path* $\mathcal{P}_b(v)$ of v, which goes from W to E.

TRANSVERSALDRAW(M):

(1) Take a regular grid of size $f_r \times f_b$, where f_r (f_b) is the number of inner faces of the red map M_r (of the blue map M_b, respectively).
(2) Place the outer vertices S, W, N, E at the grid corners $(0,0)$, $(0, f_b)$, (f_r, f_b) and $(f_r, 0)$, respectively.
(3) For each inner vertex v of M, let x be the number of inner faces of M_r on the left of $\mathcal{P}_\mathrm{r}(v)$ and let y be the number of inner faces of M_b on the right of $\mathcal{P}_\mathrm{b}(v)$. Place v at the grid point of coordinates (x, y).
(4) Link each pair of adjacent vertices by a segment.

See Figure 4 for an example of execution of TRANSVERSALDRAW(M).

THEOREM 5.2 (face-counting algorithm based on transversal structures). *Given a planar map M with quadrangular outer face and endowed with a transversal structure,* TRANSVERSALDRAW(M) *outputs a straight-line drawing of M in linear time.*

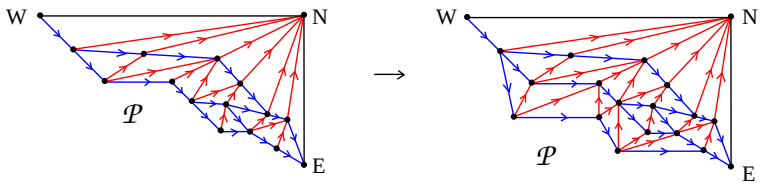


FIGURE 5. The direction properties of the edges ensure that the drawing is planar, step after step.

Red edges are strictly directed from bottom to top and weakly directed from left to right; blue edges are strictly directed from left to right and weakly directed from top to bottom. The semi-perimeter of the grid is equal to $n - 1 - \Delta$, where n is the number of vertices and Δ is the number of quadrangular faces of M. All faces of the drawing are strictly convex.

Proof. 1) Proof of the direction property of edges. Let e be a red edge of M and let v_1 be its origin and v_2 be its end-vertex. To prove that e is oriented from bottom to top in the embedding computed by TRANSVERSALDRAW(M), we have to show that the ordinate of v_2 is greater than the ordinate of v_1. This follows from the fact that $\mathcal{P}_b(v_1)$ is on the right of $\mathcal{P}_b(v_2)$. Indeed, $\mathcal{P}_b(v_1)$ and $\mathcal{P}_b(v_2)$ can not cross (see Fact 5.3 for a justification), and Condition C1' ensures that v_1 is on the right of v_2 in the orientation induced by the blue edges of M. Then, to prove that e is weakly oriented from left to right, we have to prove that the abscissa of v_1 is not greater than the abscissa of v_2, which comes down to proving that $\mathcal{P}_r(v_1)$ is not on the right of $\mathcal{P}_r(v_2)$. This last fact follows from two easy observations: the leftmost outgoing red path of v_1 is (weakly) on the left of the leftmost outgoing red path of v_2; and the rightmost ingoing red path of v_2 is (weakly) on the right of the rightmost ingoing red path of v_1. Similary the blue edges are oriented from left to right and weakly oriented from top to bottom.

2) Proof of planarity of the embedding. The proof that the embedding is planar relies on the direction properties of the red and blue edges and on the fact that they are combinatorially transversal. To carry out the proof, we use a sweeping process akin to the iterative algorithm presented in Section 2.4. The idea consists in maintaining an oriented blue path $\mathcal{P}$ from W to E called the *sweeping path*, such that the following invariants are maintained: the embedding of the edges of T that are not on the right of $\mathcal{P}$ is a straight-line drawing delimited to the top by the embedding of (W, N), to the right by the embedding of (N, E), and to the bottom-left by the embedding of $\mathcal{P}$. The sweeping path is initially (W, N, E) and then "moves" toward S. At each step, one chooses an inner face f of the blue-map of T such that its left lateral path is included in $\mathcal{P}$. Then, $\mathcal{P}$ is updated by replacing the left lateral path of f by the right lateral path of f. Thus $\mathcal{P}$ remains an oriented path from W to E and is moved toward the bottom left corner of the embedding, i.e., the vertex S. The fact that the invariants are maintained is easily checked from the geometric direction properties of the red and blue edges and from the fact that all red edges inside f connect transversally the two lateral blue paths, see Figure 5. At the end, the sweeping path is equal to (W, S, E), so that there are no edges anymore on the right of $\mathcal{P}$. Thus, the fact that the invariants are true at the end implies that the embedding of T is a straight-line drawing.

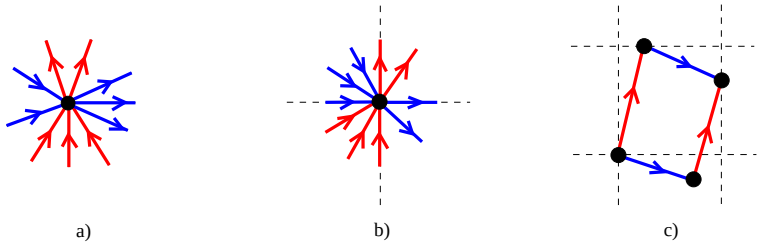


FIGURE 6. The configuration of an inner vertex in a transversal structure (Fig. a), the associated configuration in the embedding given by TRANSVERSALDRAW (Fig.b), and the configuration of an embedded quadrangular face (Fig.c).

3) Proof of the formula for the semi-perimeter. We show now that the semi-perimeter of the grid of TRANSVERSALDRAW(M) is equal to $n-1-\Delta$ if M has n vertices and Δ quadrangular faces. We write respectively f_r and f_b for the number of inner faces of the red map M_r and of the blue map M_b of M. By construction of TRANSVERSALDRAW, f_r is the width and f_b is the height of the grid. Hence we have to prove that $f_r + f_b = n - 1 - \Delta$. We write respectively e_r and e_b for the number of edges of M_r and M_b. Euler's relation ensures that the total number e of edges of M is $3n - 7 - \Delta$. Hence, $e_b + e_r = e + 4 = 3n - 3 - \Delta$. In addition, Euler's relation, applied respectively to M_r and M_b, ensures that $n + f_r = e_r + 1$ and $n + f_b = e_b + 1$. Hence, $f_r + f_b = e_r + e_b - 2n + 2 = n - 1 - \Delta$.

4) Proof of the linear time complexity. Finally, the algorithm has linear-time complexity because performing the face-counting operations takes linear time. The implementation is the following. For each inner vertex v of M, consider the rightmost outgoing red path P_1, the leftmost outgoing red path P_2, the leftmost ingoing red path P_3, and the rightmost ingoing red path P_4 starting at v (in each case, the path starts at v and leaves each vertex with the same type of edge, e.g., the rightmost outgoing path leaves each vertex using the rightmost outgoing red edge). These 4 paths partition the set of inner faces of the red-map T_r into 4 areas $\mathcal{U}(v)$, $\mathcal{L}(v)$, $\mathcal{D}(v)$ and $\mathcal{R}(v)$, respectively delimited by (P_1, P_2), (P_2, P_3), (P_3, P_4), and (P_4, P_1). Let $U(v)$, $L(v)$, $D(v)$, and $R(v)$ be the number of inner faces of T_r in $\mathcal{U}(v)$, $\mathcal{L}(v)$, $\mathcal{D}(v)$ and $\mathcal{R}(v)$. The quantity $U(v)$ is easily computed in one pass, by doing a traversal of the vertices of M from S to N. Similarly, $D(v)$ can be computed for all inner vertices of M in one pass. Then, $L(v)$ is also computed in one pass (using $D(v)$ and $U(v)$) by doing a traversal from W to E. Finally, the abscissas of all vertices can be computed using $\text{Abs}(v) = D(v) + L(v)$.

5) Proof of strict convexity of the faces. According to Proposition 1.12, all faces are either triangular or quadrangular. Clearly triangular faces are always strictly convex in any straight-line drawing. Consider now a quadrangular face f. Such a face has always the configuration of Figure 18(a). The strict convexity of f is an easy consequence of the direction property of edges, which are in specific quadrants depending on their color and orientation, see Figure 6. □

2.2. Improvement based on edge-deletions. As the drawing algorithm based on Schnyder woods, the procedure TRANSVERSALDRAW relies on face-counting operations. The idea of deleting edges to reduce the size of the grid is also fruitful

here. The principle is similar to the one used for Schnyder woods: 1) delete edges while maintaining a transversal structure, 2) embed the reduced map, 3) re-insert the deleted edges. However, the description is simpler, as the edge-deletions are done without updating the orientation and coloration of neighbouring edges.

Edge-deletion. Given M a planar map endowed with a transversal structure, an edge e of M is called *deletable* if the induced structure on $M\backslash\{e\}$ is still a transversal structure. Clearly, given the condition C1 for transversal structure, a red edge $e = (v, v')$ is deletable iff e is neither the unique outgoing red edge at v nor the unique ingoing red edge at v', and the same property holds for blue edges. Similarly, we define a *deletable set of edges* as a set $\mathcal{E}$ of edges such that the induced structure on $M\backslash\mathcal{E}$ is a transversal structure. (Notice that a set of deletable edges might not be a deletable set of edges.)

PROPOSITION 5.2 (drawing after edge-deletions). *Let M be a planar map with quadrangular outer face and endowed with a transversal structure. Given any deletable set $\mathcal{E}$ of M, the following procedure,*

(1) *call* TRANSVERSALDRAW*$(M\backslash\mathcal{E})$,*
(2) *insert the edges of $\mathcal{E}$ in the obtained embedding,*

outputs a strictly convex straight-line drawing of M on a grid of semi-perimeter $n - 1 - |\mathcal{E}| - \Delta$, where Δ is the number of quadrangular faces of M.

Proof. According to Theorem 5.2, TRANSVERSALDRAW$(M\backslash\mathcal{E})$ is a straight-line drawing of $M\backslash\mathcal{E}$, and all faces of $M\backslash\mathcal{E}$ are strictly convex. Hence, the reinsertion of edges clearly maintains planarity and the strict convexity of the faces. The grid size is $\widehat{f_r} \times \widehat{f_b}$, where $\widehat{f_r}$ ($\widehat{f_b}$) is the number of inner faces of the red map (blue map, respectively) of $M\backslash\mathcal{E}$. Let f_r (f_b) be the number of inner faces of the red map (blue map, respectively) of M. Clearly, $\widehat{f_r} + \widehat{f_b} = f_r + f_b - |\mathcal{E}|$. According to Theorem 5.2, $f_r + f_b = n - 1 - \Delta$, so that the semi-perimeter of the embedding is equal to $\widehat{f_r} + \widehat{f_b} = n - 1 - \Delta - |\mathcal{E}|$. □

Thus the smallest grid to be expected with this procedure is obtained by using a deletable set with maximal cardinality, which is described next.

Computing deletable sets of maximal cardinality. Clearly, a red (blue) edge is deletable iff the induced orientation of the red (blue) edges is still bipolar. Moreover, an important property used here is that the two bipolar orientations have no transitive edge, as mentioned after Lemma 2.5. Hence, we have to characterise the edges whose deletion does not break bipolarity in a bipolar orientation with no transitive edge. Define an N-pattern of X as a path of length 3 $(v_0, e_0, v_1, e_1, v_2, e_2, v_3)$ such that e_0 and e_1 are ingoing at v_1, e_1 and e_2 are outgoing at v_2, e_1 follows e_0 in ccw order around v_1, and e_1 follows e_2 in ccw order around v_2, see Figure 7(a). Similarly, define an S-pattern of X as a path of length 3 $(v_0, e_0, v_1, e_1, v_2, e_2, v_3)$ such that e_0 and e_1 are outgoing at v_1, e_1 and e_2 are ingoing at v_2, e_1 follows e_0 in cw order around v_1, and e_1 follows e_2 in cw order around v_2, see Figure 7(b). The edge e_1 is called the *central edge* of the pattern. The conditions of plane bipolar orientation (see Figure 6) and the fact that X has no transitive edge easily imply that an edge is deletable iff it is the central edge of an N-pattern or the central edge of an S-pattern . We define a *zigzag-path* as a path $P = (v_0, e_0, \ldots, e_k, v_{k+1})$ of length at least 3 such that any three consecutive edges of P form alternatively an N-pattern or an S-pattern, see Figure 7(c). The two edges e_0 and e_k are called

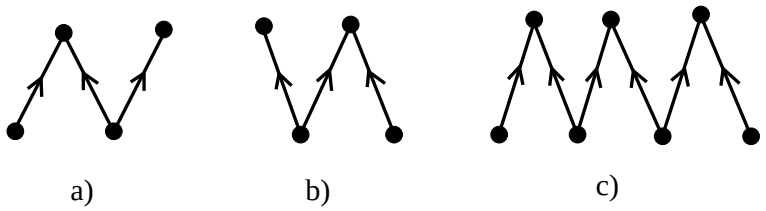


FIGURE 7. An N-pattern (Fig.a), an S-pattern (Fig.b), and a zigzag path (Fig.c).

the *extremal edges* of P and the other edges of P are called *internal edges* of P. A zigzag-path is called *maximal* if it is not strictly contained in another zigzag-path.

LEMMA 5.2. *Let X be a plane bipolar orientation with no transitive edge. Then each deletable edge is an internal edge in a unique maximal zigzag-path.*

Proof. As already mentioned, an edge is deletable iff it is the central edge of an N-pattern or the central edge of an S-pattern. Clearly, there is a unique way to extend such a pattern to a maximal zigzag-path. The result follows. □

Notice that an internal edge of a zigzag path can be deleted iff its two neighbouring edges in the zigzag-path are not deleted. Based on this, it is easy to describe a procedure that computes a deletable set $\mathcal{E}$ of maximal cardinality:

COMPUTEMAXIMALDELETABLESET(X)
(1) Compute the maximal zigzag-paths of X.
(2) For each maximal zigzag-path $P = (v_0, e_0, \ldots, e_k, v_{k+1})$, add to $\mathcal{E}$ the edges e_i of P with odd index.

To compute a deletable set of a transversal structure with maximal cardinality, it thus suffices to run this procedure on each of the two transversal bipolar orientations. We can now describe an algorithm that exploits edge-deletions optimally.

COMPACTTRANSVERSALDRAW(M)
#M is endowed with a transversal structure
#X_r (X_b) is the red (blue, respectively) bipolar orientation
(1) Call $\mathcal{E}_r \leftarrow$ COMPUTEMAXIMALDELETABLESET(X_r).
$\mathcal{E}_b \leftarrow$ COMPUTEMAXIMALDELETABLESET(X_b)
(2) Call TRANSVERSALDRAW($M \backslash (\mathcal{E}_r \cup \mathcal{E}_b)$).
(3) Insert the edges of $\mathcal{E}_r \cup \mathcal{E}_b$ in the obtained embedding.

The above discussion yields directly the following result on COMPACTTRANSVERSALDRAW(M).

THEOREM 5.3 (optimal drawing after edge-deletions). *Given a planar map M with quadrangular outer face and endowed with a transversal structure, the algorithm* COMPACTTRANSVERSALDRAW(M) *outputs a straight-line drawing of M in linear time. The semi-perimeter of the grid is equal to $n - 1 - \widehat{\Delta}$, where n is the number of vertices and $\widehat{\Delta}$ is the sum of the number of quadrangular faces in M and of the maximal number of edges that can be deleted from M while keeping a transversal structure. All faces of the drawing are strictly convex.*

Reduction of the grid based on coordinate-deletions. A simpler technique to reduce the size of the grid is to work directly on the embedding. The reduction

is done by deleting unoccupied coordinates, see Figure 9 for an example. However, the following lemma ensures that the technique of edge-deletion performs at least as good as the technique of coordinate-deletion. In particular, it ensures that the algorithm CompactTransversalDraw outputs a straight-line drawing where all coordinate-lines are occupied.

Lemma 5.3. *Let M be a planar map endowed with a transversal structure X. Then the unoccupied abscissas (ordinates) of* TransversalDraw*(M) are in one-to-one correspondence with the central edges of red (blue, respectively) S-patterns of X. Moreover, deleting a set of unoccupied coordinates is equivalent to calling the algorithm of Theorem 5.3 with the deletable set $\mathcal{E}$ containing the corresponding central edges of N-patterns.*

The proof of this lemma is rather technical and is delayed to the appendix (Section 4).

2.3. Drawing irreducible triangulations. The first family of maps for which the straight-line drawing applies is the family of irreducible triangulations, as these can be endowed with transversal structures. To carry out a probabilistic analysis of the grid size (see Section 3.2), we consider the minimal transversal structure.

Theorem 5.4 (drawing irreducible triangulations). *For any irreducible triangulation T, the procedure*

(1) *Compute the minimal transversal structure of T,*
(2) $E \leftarrow$ TransversalDraw(T)
    $E_c \leftarrow$ CompactTransversalDraw(T)

has linear time complexity and outputs two straight-line drawings of T (E and E_c). In both drawings, the red edges are directed from bottom to top and weakly directed from left to right; and the blue edges are directed from left to right and weakly directed from top to bottom. If T has n vertices, then the semi-perimeter of E is equal to $n-1$ and the semi-perimeter of E_c is $n-1-\Delta$, where Δ is the number of unused coordinates of E.

Proof. Given a transversal structure on T, the presence of an N-pattern clearly yields a right alternating 4-cycle. As we consider the minimal transversal structure, there is no N-pattern, so that the maximal zigzag paths are of length 3 and are S-patterns. Hence the algorithm CompactTransversalDraw deletes the internal edges of each N-pattern, and then calls the face-counting algorithm TransversalDraw. According to Lemma 5.3, this is equivalent to deleting the unused coordinates, see Figure 9. Clearly, deleting unoccupied coordinates clearly preserves the geometric direction property of red and blue edges satisfied by TransversalDraw(M). □

The choice of the minimal transversal structure makes it possible to perform a probabilistic analysis of the grid size of TransversalDraw and CompactTransversalDraw. As we will prove in Section 3.2, both algorithms output asymptotically with high probability square grids that are respectively $n/2 \times n/2$ for TransversalDraw and $11n/27 \times 11n/27$ for CompactTransversalDraw (up to relative fluctuations of order $1/\sqrt{n}$). An example is given in Figure 8, where the irreducible triangulation is uniformly taken in $\mathcal{T}'_{200}$ using the random sampler of Section 3.2 . The gain both in length and in width by a factor $\approx 27/22$ is already visible.

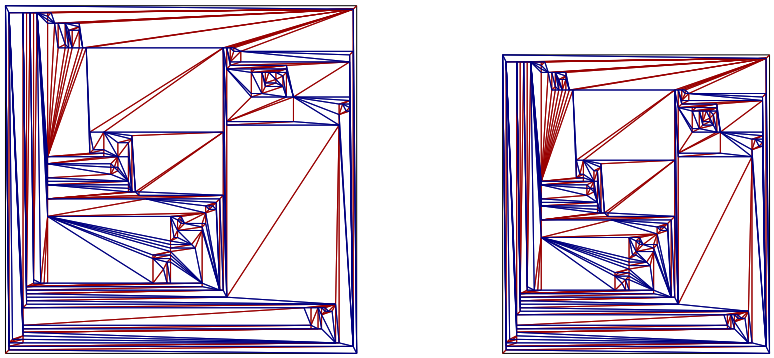


FIGURE 8. A random triangulation with 200 vertices endowed with its minimal transversal structure and embedded with TRANSVERSALDRAW and COMPACTTRANSVERSALDRAW.

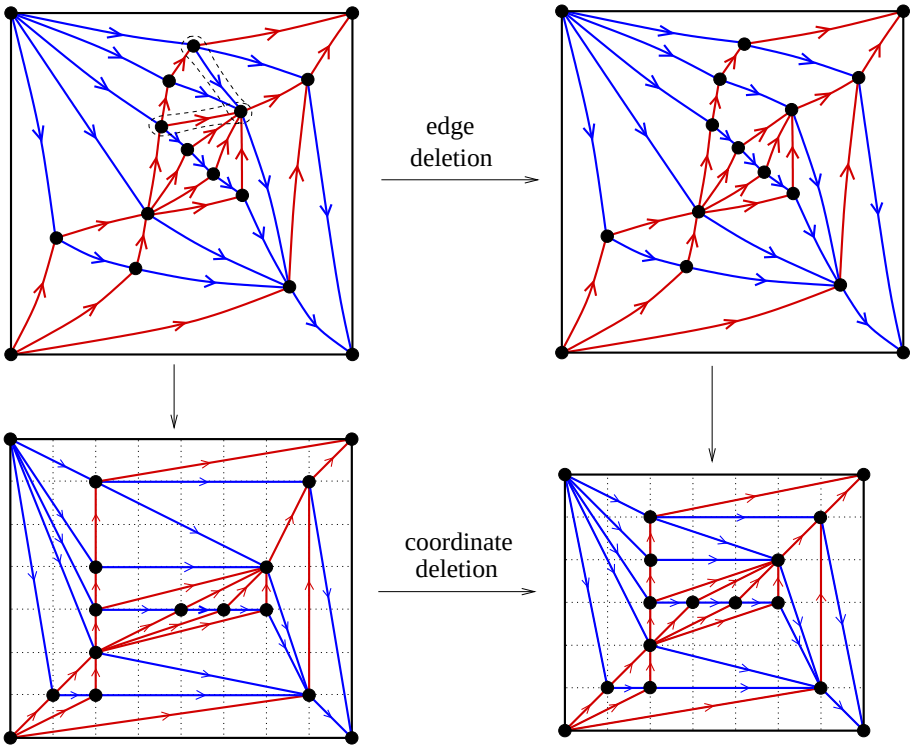


FIGURE 9. Given an irreducible triangulation endowed with its minimal transversal structure, the result of COMPACTTRANSVERSALDRAW is the same as a call to TRANSVERSALDRAW followed by a deletion of the unused coordinates.

2.4. Drawing quadrangulations. Surprisingly, the straight-line drawing algorithm based on transversal structures can be applied to draw quadrangulations, even if quadrangulations do not admit a transversal structure. The key obervation

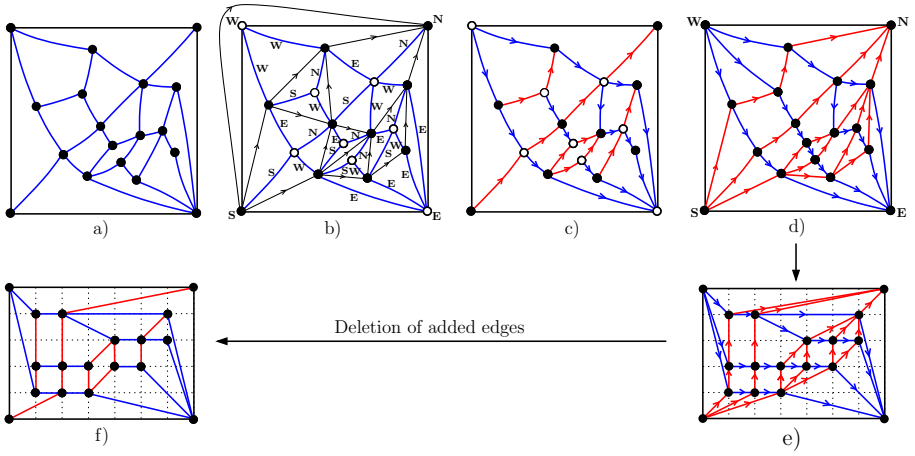


FIGURE 10. A quadrangulation Q (Fig.a), endowed with an angular partition of the inner edges (Fig.b), the associated uncomplete transversal structure (Fig.c), and the straight-line drawing of Q (Fig.d).

is that it is possible to augment a quadrangulation by adding edges, so as to obtain a map endowed with a transversal structure. The edge-addition process leads us to refine the correspondence (described in Section 4.2) between 2-orientations of a quadrangulation Q and bipolar orientations of the primal map M. This time, we show that a bipolar orientation of M gives rise to a specific quadri-partition of the inner edges of Q. Such a quadri-partition is also investigated in [**76**] using another terminology (with labels). We use the quadri-partition to triangulate partially Q into a map M endowed with a transversal structure, which is then embedded using TRANSVERSALDRAW(M).

Quadri-partition of inner edges of a quadrangulation. Let Q be a quadrangulation, the four outer vertices in clockwise order denoted by N, E, S, and W. We endow Q with its unique bicoloration of vertices such that S and N are black, and consider the primal map M of Q, see Figure 10(b).

Given a bipolar orientation X of M with source S and sink N, an angle (e, e') of two edges of M around a black vertex v —with e' following e in cw order around v— is called an angle *of type* N, S, W, E if (e, e') are (ingoing, ingoing), (outgoing, outgoing), (ingoing, outgoing), (outgoing, ingoing), respectively. Accordingly, the inner edges of Q are partitioned into N-edges, S-edges, W-edges and E-edges, depending on the type of their associated angle. This partition is called the *angular edge-partition* of Q associated with X, see Figure 10(b). Recall the property of plane bipolar orientations illustrated in Figure 6: the edges incident to an inner vertex are partitioned into a non-empty interval of ingoing edges and a non-empty interval of outgoing edges; and, dually, each face f of M has two particular vertices S_f and N_f such that the contour of f consists of two non-empty oriented paths both going from S_f to N_f, called *left lateral path* and *right lateral path* of f, respectively. Hence, each inner black vertex v of Q is incident to one W-edge e_W and one E-edge e_E, which are separated by a possibly empty interval of N-edges in the cw sector between e_E and e_W and a possibly empty interval of S-edges in the cw sector

between e_W and e_E. Dually, each white vertex of Q, corresponding to a face f of M, is incident to one N-edge e_N (connected to N_f) and one S-edge e_S (connected to S_f), which are separated by a possibly empty interval of W-edges in the cw sector between e_N and e_S and a possibly empty interval of E-edges in the cw sector between e_S and e_N. Observe also that all inner edges of Q incident to N, E, S, and W are N-edges, E-edges, S-edges, and W-edges respectively, see Figure 10(b).

The algorithm. Let Q be a quadrangulation endowed with an angular partition of the inner edges, associated with a bipolar orientation X of the primal map M. The *uncomplete transversal structure* is the orientation and bicoloration of the inner edges of Q where the N-edges and S-edges are colored red, the W-edges and E-edges are colored blue, the S-edges and E-edges are oriented from their black to their white vertex, and the N-edges and W-edges are oriented from their white to their black vertex, see Figure 10(c). Clearly, the conditions of a transversal structure are satisfied, except that some of the four intervals of edges around an inner vertex may be empty. Precisely, if a black vertex v of M has $i \geq 1$ ingoing edges and $j \geq 1$ outgoing edges, then v is incident in cw order to an outgoing blue edge (the E-edge e_E), an interval of $(i-1)$ ingoing red edges, an ingoing blue edge (the W-edge e_W), and an interval of $(j-1)$ outgoing red edges. Dually, if a face of M has a left lateral path of length $i \geq 1$ and a right lateral path of length $j \geq 1$, then the associated white vertex of Q is incident in cw order to an outgoing red edge (the N-edge e_N), an interval of $(j-1)$ outgoing blue edges, an ingoing red edge (the S-edge e_S), and an interval of $(i-1)$ ingoing blue edges.

We now describe an algorithm PartTriang(Q), which adds colored oriented edges to Q so as to obtain a partially triangulated planar map G endowed with a transversal structure. An edge e of M is said to be *undeletable* if e is the unique outgoing edge of its origin or the unique ingoing edge of its end-vertex (or both). An edge of M different from the edge (S, N) is said to be *transitive* if it connects the two poles N_f and S_f of a face of M. Notice that an edge of M can not be undeletable and transitive. The planar map graph $G =$ PartTriang(Q) is obtained by adding to Q the undeletable edges of M, colored red and oriented as in X; and by adding the edges dual to the transitive edges, colored blue and oriented from left to right, i.e., for each face of Q associated with a transitive edge e of M, a blue edge is added connecting the two white vertices of the face and oriented from the left of e to the right of e. It is easily checked that the edge bicoloration and orientation of G is a transversal structure. An example of execution of PartTriang is shown in Figure 10(c)-(d).

Theorem 5.5 (drawing quadrangulations). *Let Q be a quadrangulation endowed with an angular partition of its inner edges. The algorithm that first computes* G=PartTriang(Q)*, then calls* TransversalDraw(G)*, and finally deletes the edges added from Q to G, is a straight-line drawing algorithm for quadrangulations with linear time complexity. No edge of G is deletable, so that the result of* TransversalDraw*(G) is the same as* CompactTransversalDraw*(G). In particular, all coordinate-lines are occupied by at least one vertex. The semi-perimeter $W + H$ of the grid satisfies*

$$W + H = n - 1 - \Delta,$$

where n is the number of vertices and Δ is the number of alternating faces of Q, i.e., faces whose contour consists of two red edges and two blue edges that alternate. Alternating faces are strictly convex in the drawing.

Proof. The result follows from Theorem 5.2 and from the fact that the faces of Q that are not split into two triangles are the alternating faces. These faces are strictly convex because they are faces of G and TransversalDraw(G) is strictly convex. Finally, TransversalDraw(G) is the same as CompactTransversalDraw(G) because G has no deletable edge. Indeed, the obtained transversal structure of G satisfies the following stronger property (P): each edge $e = (v, v')$ of the red (blue) bipolar orientation of G is either the unique outgoing red (blue) edge at v or is the unique ingoing red (blue, respectively) edge at v'. The proof of (P) is easy; we check that (P) is true on the uncomplete transversal structure of Q and remains true for each edge-addition. □

Definition. Given a quadrangulation Q, the angular edge-partition associated with the minimal bipolar orientation of the primal map of Q is called the *minimal* angular edge-partition of Q.

Choosing the minimal angular edge-partition in the generic embedding algorithm described in Theorem 5.5 yields a canonical embedding that can be computed in linear time.

DrawQuad(Q):
(1) Compute the minimal angular edge-partition of Q.
(2) Compute $G = \text{PartTriang}(Q)$.
(3) Call TransversalDraw(G).
(4) Delete the edges added from Q and G.

As we will see in Section 3.3, the choice of the minimal angular edge-partition makes it possible to perform a probabilistic analysis of the grid size of the algorithm DrawQuad(Q). The net result is that a rooted n-vertex quadrangulation drawn uniformly at random is embedded on a grid whose size is with high probability $13n/27 \times 13n/27$, up to fluctuations of order $\sqrt{n}$.

3. Analysis of the drawing algorithms

Due to their close connection with minimal α-orientations, the bijections presented in Chapter 3 lead to elegant methods to analyze straight-line drawing algorithms. The method applies when the embedding algorithms make use of a combinatorial structure that can be formulated as a minimal α-orientation. This property is satisfied by the algorithms CompactSchnyderDraw (embedding 3-connected maps), CompactTransversalDraw (embedding irreducible triangulations), and DrawQuad (embedding quadrangulations). The scheme of the analysis is that the grid size is formulated in terms of combinatorial parameters of the map, which turn out to correspond to specific parameters of the associated tree. The analysis of the distribution of these parameters in a random tree of size n is then performed using the generating function methodology. This method has been introduced by Bonichon *et al* [**18**] to analyse the grid size of an algorithm embedding triangulations [**118**], which improves on Schnyder's [**102**] by discarding some edges in the face-counting process.

3.1. Grid size of 3-connected maps. As we have seen in Section 1.2, the algorithm CompactSchnyderDraw(M) outputs a straight-line drawing of M on a grid $(n-1-\Delta) \times (n-1-\Delta)$, where n is the number of vertices of M and Δ is an explicit parameter of M endowed with its minimal Schnyder wood, namely Δ is the number of *counter-clockwise faces* of M. Let us mention that, using the

bijection of Poulalhon and Schaeffer [**95**], the parameter Δ is shown in [**18**] to be of order $n/8$ for random triangulations with n vertices. The authors of [**16**] also perform tests indicating that Δ is of order $m/8$ for random 3-connected planar map with m edges.

The present section provides a thorough analysis of the parameter Δ for random 3-connected planar maps, recovering in particular the result for random triangulations and validating the tests for random m-edges 3-connected planar maps. The analysis relies on the bijection of Section 2.1, formulated as a correspondence between 3-connected planar maps and binary trees.

THEOREM 5.6 (Grid size of 3-connected planar maps). *Let $n \geq 4$ and $\beta \in [1/2+1/n, 2-5/n]$* [1]*. Let G be taken uniformly at random among 3-connected planar map with n vertices and $\lfloor \beta n \rfloor$ inner faces, and embedded using* CompactSchnyderDraw*. Then the width $X_{n,\beta}$ of the square grid is asymptotically with high probability equal to $(1-(4\beta)^{-1})n$ in the following "exponential" sense: for each fixed $\epsilon > 0$, there exist a constant $c_\epsilon > 0$ and an integer N_ϵ, depending only on ϵ, such that*

$$\mathbf{P}\left(X_{n,\beta} \notin [(1-(4\beta)^{-1}-\epsilon)n, (1-(4\beta)^{-1}+\epsilon)n]\right) \leq \exp(-c_\epsilon n)$$

for $n \geq N_\epsilon$ and $\beta \in [1/2+1/n, 2-5/n]$. In addition, c_ϵ is of order ϵ^2 and N_ϵ is of order $\epsilon^{-2}\ln(\epsilon^{-1})$ when ϵ gets small.

COROLLARY 5.1. *In the same "exponential" sense, the following asymptotic results hold:*

- *The width of the square grid for a random suspended internally 3-connected map with n vertices is asymptotically with high probability $c\,n$ where $c = \frac{\sqrt{7}-1}{2} \approx 0.82$.*
- *The width of the square grid for a random suspended internally 3-connected map with m edges is asymptotically with high probability $\frac{3}{4}n$, where n is its number of vertices.*
- *The width of the square grid for a random triangulation with n vertices is asymptotically with high probability $7n/8$ (already stated in* [**16**]*).*

3.1.1. *Opening a suspended internally 3-connected map into a binary tree.* A crucial tool to analyse the compaction parameter Δ of CompactSchnyderDraw is to reformulate the bijection of Section 2.1 as an injective mapping from suspended internally 3-connected maps to binary trees, in a way close to the presentation of the encoding algorithm for 3-connected maps (Section 2.1).

The *opening* of a suspended internally 3-connected planar map M consists of the following four steps, illustrated in Figure 11; 1) endow the derived map M with its minimal α_3-orientation; 2) consider the irreducible dissection D whose edge-set corresponds to vertex-face incidences of M; 3) give to each half-edge h of D the direction of the edge of the derivated map M' that follows in cw order around the origin of h; 4) open the oriented dissection into a binary tree τ by deleting the ingoing half-edges and the three outer white vertices, and root τ at the half-edge going out of a_1 and having the outer face on its right.

PROPOSITION 5.3. *For $n \geq 4$ and $r \in [n/2+1, 2n-5]$, the opening is an injective map from the set of suspended 3-connected planar maps with n vertices*

[1] The lower bound $1/2 + 1/n$ and upper bound $2 - 5/n$ correspond respectively to cubic 3-connected planar graphs (all vertices have degree 3) and to triangulations.

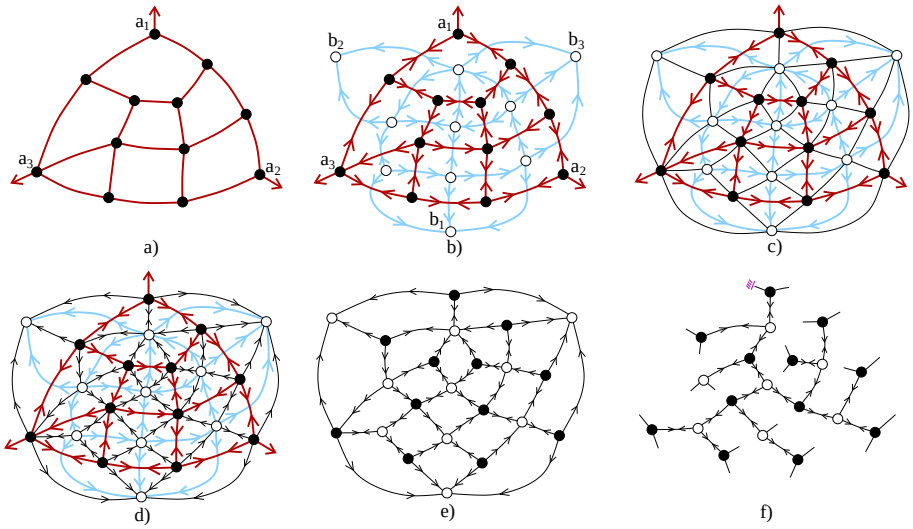


FIGURE 11. Opening of a suspended 3-connected planar map into a black-rooted bicolored binary tree.

and r inner faces to the set of black-rooted bicolored binary trees with n black nodes and r white nodes.

Proof. The proof relies on easy modifications of the arguments used to show the bijection between binary trees and irreducible dissections in Section 2.1. Let us indicate the main points. The fact that the dissection is irreducible results from the fact that M is internally 3-connected. The orientation of half-edges of D is such that each vertex has outdegree 3 except the 3 outer white vertices, which have outdegree 0. Moreover, no edge is made of two ingoing half-edges, so that the edges of D are either bi-directed or simply directed. In addition, the orientation has no clockwise cycle, i.e., all cycle have at least one simply directed edge with the exterior on its right. Simple counting arguments and the absence of cycle of bi-directed edges ensure that the subgraph made of bi-directed edges is a spanning tree of the dissection. Thus the figure finally obtained is a binary tree. Finally, injectivity follows from the fact that the dissection (and then the suspended map) can be recovered from the tree using closure operations, as described in Section 2.1. □

3.1.2. *Analysis of the grid size using the opening mapping.* According to Theorem 5.1, analysing the grid size of the algorithm CompactSchnyderDraw reduces to analysing the number of counterclockwise faces in a random suspended 3-connected planar map. This latter parameter turns out to correspond, via the opening map, to a simple parameter of the associated binary tree.

Definition. An *internal node* in a binary tree is a node whose three neighbours are themselves nodes, i.e., an internal node is a node not adjacent to any leaf.

LEMMA 5.4. *Let M be a suspended internally 3-connected planar map, endowed with its minimal Schnyder wood, and let τ be the bicolored binary tree obtained by doing the opening of M. Then the number of counterclockwise faces of G is equal to the number of internal white nodes of τ.*

Proof. Let us characterise the internal nodes of the tree in terms of the orientation of half-edges of the dissection D of the hexagon. When the ingoing half-edges of D are deleted to obtain a binary tree, a vertex v of D does not become connected to a leaf iff its three outgoing edges are bi-directed The end-vertex of a simply oriented edge going out of v is called a *deletion-neighbour* of v. Thus, internal white nodes of the binary tree correspond to white vertices of D without deletion-neighbours.

Given a white vertex v of the dissection D, let f be the inner face of M associated with v, and let $e_1(f)$, $e_2(f)$, $e_3(f)$ be the three particular edges of f, as illustrated in Figure 1(c) page 157. Observe that the neighbours of v in D are the vertices of M on the contour of f. Given a vertex v' on the contour of f, we write $e(v')$ for the half-edge of M on the contour of f that starts from v' with the interior of f on its left. Notice that $e(v')$ is also an edge of the derived map M'. As each half-edge of D inherits the direction of the cw-consecutive edge of the derived map M', there exists a deletion-neighbour v' of v in D iff there exists a vertex v' of M on the contour of f such that $e(v')$ is ingoing. As all edges on f and different from $e_1(f), e_2(f), e_3(f)$ are bi-directed, this happens iff one of the three edges $e_1(f), e_2(f), e_3(f)$ is simply directed clockwise. Thich concludes the proof, given the characterisation of counter-clockwise faces when the Schnyder wood is minimal, see the definition page 159. □

LEMMA 5.5. *Let $B_{n,r}$ be the number of black-rooted bicolored binary trees with n black nodes and r white nodes; let $B_{n,r,k}$ be the number of black-rooted bicolored binary trees with n black nodes, r white nodes and k internal white nodes. Then*

$$B_{n,r} = \frac{1}{n}\binom{2n}{r}\binom{2r}{n-1} \tag{34}$$

$$B_{n,r,k} = \frac{(2n-1)!2^{n-2k}}{(2n-r)!(r-n+k+1)!(n-1-2k)!k!} \tag{35}$$

Proof. Let $B_{n,r}$ (resp. $W_{n,r}$) be the number of black-rooted (resp. white-rooted) binary trees with n black nodes and r white nodes. Let $B(x,y) := \sum_{n,r} B_{n,r}x^n y^r$ and $W(x,y) := \sum_{n,r} W_{n,r}x^n y^r$ be their associated generating functions, where the formal variables x and y mark respectively the number of black nodes and the number of white nodes. The decomposition of a rooted bicolored binary tree at the root yields

$$B(x,y) = x(1+W(x,y))^2 \qquad W(x,y) = y(1+B(x,y))^2.$$

Given a bivariate generating function $f(x,y) = \sum_{n,r} f_{n,r}x^n y^r$, write $[x^n y^r]f(x,y)$ for the coefficient $f_{n,r}$. Similary, given a univariate generating function $f(x) = \sum_n f_n x^n$, write $[x^n]f(x)$ for the coefficient f_n. The bivariate version of Lagrange inversion formula [**67**] states that, if two bivariate generating functions $B(x,y)$ and $W(x,y)$ are related by a system of the form

$$B(x,y) = x\phi(W(x,y)) \qquad W(x,y) = y\psi(B(x,y)),$$

with $W \to \phi(W)$ and $B \to \psi(B)$ two univariate series, then

$$[x^n y^r]B(x,y) = \frac{1}{n}[W^r](\phi(W)^n)[B^{n-1}](\psi(B)^r).$$

As a consequence, $B_{n,r} = \frac{1}{n}[W^r](1+W)^{2n}[B^{n-1}](1+B)^{2r} = \frac{1}{n}\binom{2n}{r}\binom{2r}{n-1}$. The coefficient $B_{n,r,k}$ can be derived in a similar way. First, observe that a white node of a black-rooted binary tree is internal iff its two sons are not leaves. Let

$B_{n,r,k}$ (resp. $W_{n,r,k}$) be the number of black-rooted (resp. white-rooted) binary trees with n black nodes, r white nodes and k white nodes whose two sons are not leaves. Let $B = B(x,y,u) := \sum_{n,r,k} B_{n,r,k} x^n y^r u^k$ and $W = W(x,y,u) := \sum_{n,r,k} W_{n,r,k} x^n y^r u^k$ be their associated generating functions, where the variables x, y and u mark respectively the number of black nodes, the number of white nodes and the number of white nodes whose two sons are not leaves. The decomposition at the root node yields

$$B = x(1+W)^2 \qquad W = y(1+2B+uB^2),$$

where the variable u appears at the root if the root node is white and has two non-empty subtrees. Observe that $[x^n y^r]B(x,y,u)$ is the univariate polynomial $B_{n,r}(u) := \sum_k B_{n,r,k} u^k$. Thus, bivariate Lagrange inversion formula yields

$$\begin{aligned} B_{n,r}(u) &= \frac{1}{n}[W^r](1+W)^{2n}[B^{n-1}](1+2B+uB^2)^r \\ &= \frac{1}{n}\binom{2n}{r} \sum_{\substack{r_1+r_2+r_3=r \\ r_2+2r_3=n-1}} \frac{r!}{r_1!r_2!r_3!} 2^{r_2} u^{r_3} \\ &= \frac{1}{n}\binom{2n}{r} \sum_{k\geq 0} \frac{r!}{(r-n+1+k)!(n-1-2k)!k!} 2^{n-1-2k} u^k. \end{aligned}$$

As a consequence $B_{n,r,k} = [u^k]B_{n,r}(u) = \frac{(2n-1)!2^{n-2k}}{(2n-r)!(r-n+1+k)!(n-1-2k)!k!}$. In particular, $B_{n,r,k} = 0$ if $k \geq n/2$ or $k < n-r-1$. □

LEMMA 5.6. *Let $n \geq 4$ and $\beta \in [1/2, 2]$. Let τ be a black-rooted bicolored binary tree with n black nodes and $\lfloor \beta n \rfloor$ white nodes taken uniformly at random. Then the number $X_{n,\beta}$ of internal white nodes of τ is asymptotically with high probability close to $\frac{n}{4\beta}$ in the following "exponential" sense: for each fixed $\epsilon > 0$, there exist a constant $c_\epsilon > 0$ and an integer N_ϵ, depending only on ϵ, such that*

$$\mathbf{P}\left(X_{n,\beta} \notin [((4\beta)^{-1} - \epsilon)n, ((4\beta)^{-1} + \epsilon)n]\right) \leq \exp(-c_\epsilon n)$$

for $n \geq N_\epsilon$ and $\beta \in [1/2, 2]$. In addition, c_ϵ is of order ϵ^2 and N_ϵ is of order $\epsilon^{-2}\ln(\epsilon^{-1})$ for ϵ small.

Proof. Let $\mathcal{E}$ be the set $\{\beta \in [1/2,2], \lambda \in [\max(0, 1-\beta), 1/2]\}$. Based on Stirling Formula, it is easily proved that the expressions for $B_{n,r}$ and $B_{n,r,k}$ given in Lemma 5.5 satisfy for $\beta \in [1/2, 2]$,

$$\lim_{n\to\infty} \frac{1}{n} \ln\left(B_{n,\lfloor \beta n \rfloor}\right) = 2\ln(2) - \beta\ln(\beta) - (2-\beta)\ln(2-\beta) + 2\beta\ln(2\beta) - (2\beta-1)\ln(2\beta-1),$$

and for $(\beta, \lambda) \in \mathcal{E}$,

$$\begin{aligned} \lim_{n\to\infty} \frac{1}{n} \ln\left(B_{n,\lfloor \beta n \rfloor, \lfloor \lambda n \rfloor}\right) &= 2\ln(2) - (\lambda+\beta-1)\ln(\lambda+\beta-1) - (1-2\lambda)\ln(1-2\lambda) \\ &\quad -\lambda\ln(\lambda) + (1-2\lambda)\ln(2) - (2-\beta)\ln(2-\beta). \end{aligned}$$

Write $F(\beta)$ for the first limit and $f(\beta,\lambda)$ for the second limit. A simple calculation ensures that $\frac{\partial^2 f}{\partial \lambda^2}(\beta,\lambda)$ is strictly negative on $\mathcal{E}$ (with $-\infty$ limits on the boundary). Hence, for $\beta \in [1/2, 2]$ $\lambda \to f(\beta,\lambda)$ has its unique maximum at the solution of $\frac{\partial F}{\partial \lambda}(\beta,\lambda) = 0$, which is found to be $\lambda_0 = (4\beta)^{-1}$. Moreover, $f(\beta, (4\beta)^{-1}) = F(\beta)$. Let $\epsilon > 0$. As $(\beta,\lambda) \to f(\beta,\lambda) - F(\beta)$ is continuous and strictly negative on the compact set $\mathcal{E}_\epsilon := \mathcal{E} \backslash \{\lambda \in [(4\beta)^{-1} - \epsilon, (4\beta)^{-1} + \epsilon]\}$, there exists a constant η such that $f(\beta,\lambda) \leq F(\beta) - \eta$ on $\mathcal{E}_\epsilon$. Precisely, Taylor formula ensures that one can take

$\eta = M\epsilon^2$, where $M = \frac{1}{2}|\max_{\mathcal{E}}(\frac{\partial^2 f}{\partial \lambda^2})|$ (M is computable, $M \approx 6.24$). Moreover, the Stirling bounds $(s/e)^s \leq s! \leq e^{1/12s}\sqrt{2\pi s}(s/e)^s$ (given for instance in [**14**]) imply that, in the first limit above, $\frac{1}{n}\ln\left(B_{n,\lfloor\beta n\rfloor}\right)$ differs from $F(\beta)$ by a term of order $\ln(n)/n$; and, in the second limit, $\frac{1}{n}\ln\left(B_{n,\lfloor\beta n\rfloor,\lfloor\lambda n\rfloor}\right)$ differs from $f(\beta,\lambda)$ by a term of order $\ln(n)/n$. Hence, there exists a number $\tilde{N}_\epsilon$ —which is easily proved to be of order $\eta^{-1}\ln(\eta^{-1})$ (i.e., of order $\epsilon^{-2}\ln(\epsilon^{-1})$)— such that

$$\frac{1}{n}\ln\left(B_{n,\lfloor\beta n\rfloor,k}\right) \leq \frac{1}{n}\ln\left(B_{n,\lfloor\beta n\rfloor}\right) - \frac{\eta}{2}$$

for $n \geq \tilde{N}_\epsilon$, $\beta \in [1/2, 2]$ and $k/n \in \mathcal{E}_\epsilon$, so that $\mathbf{P}(X_{n,\beta} = k) \leq \exp(-\eta n/2)$. As $\mathbf{P}(X_{n,\beta} = k) = 0$ if $k \geq n/2$, there are at most $n/2$ possible values for k. Hence, for $n \geq \tilde{N}_\epsilon$ and $\beta \in [1/2, 2]$,

$$\mathbf{P}\left(X_{n,\beta} \notin [((4\beta)^{-1} - \epsilon)n, ((4\beta)^{-1} + \epsilon)n]\right) \leq \frac{n}{2}\exp(-\eta n/2).$$

As $n\exp(-n\eta/2) \to 0$, there exists a number $\hat{N}_\epsilon$ —which is easily proved to be of order $\eta^{-1}\ln(\eta^{-1})$ (i.e., of order $\epsilon^{-2}\ln(\epsilon^{-1})$)— such that $\frac{n}{2}\exp(-\eta n/2) \leq \exp(-\eta n/4)$ for $n \geq \hat{N}_\epsilon$. The lemma is proved; take $c_\epsilon := \frac{1}{4}M\epsilon^2$ and $N_\epsilon := \max(\tilde{N}_\epsilon, \hat{N}_\epsilon)$. □

Proof of Theorem 5.6. For $\epsilon > 0$, $n \geq 4$ and $\beta \in [1/2 + 1/n, 2 - 5/n]$, we define:

- $\mathcal{G}_{n,\beta}$ the set of suspended internally 3-connected maps with n vertices and $\lfloor\beta n\rfloor$ faces, counting the outer face with weight 3.
- $\mathcal{G}_{n,\beta,\epsilon}$ the set of elements G of $\mathcal{G}_{n,\beta}$ whose number of counterclockwise faces is outside of $[((4\beta)^{-1} - \epsilon)n - 1, ((4\beta)^{-1} + \epsilon)n - 1]$.
- $\mathcal{B}_{n,\beta}$ the set of black-rooted bicolored binary trees with n black nodes and $\lfloor\beta n\rfloor$ white nodes.
- $\mathcal{B}_{n,\beta,\epsilon}$ the set of elements of $\mathcal{B}_{n,\beta}$ whose number of internal white nodes is outside of $[((4\beta)^{-1} - \epsilon)n - 1, ((4\beta)^{-1} + \epsilon)n - 1]$.

Proving Theorem 5.6 reduces to finding a constant $c_\epsilon > 0$ of order ϵ^2 and a number N_ϵ of order $\epsilon^{-2}\ln(\epsilon^{-1})$ such that $|\mathcal{G}_{n,\beta,\epsilon}| \leq e^{-c_\epsilon n}|\mathcal{G}_{n,\beta}|$ for each $n \geq N_\epsilon$ and $\beta \in [1/2 + 1/n, 2 - 5/n]$.

Injectivity of the opening map (Proposition 5.3) and Lemma 5.4 ensure that $|\mathcal{G}_{n,\beta,\epsilon}| \leq |\mathcal{B}_{n,\beta,\epsilon}|$. Moreover, Lemma 5.6 ensures that there a exist a constant $\tilde{c}_\epsilon$ of order ϵ^2 and an integer $\tilde{N}_\epsilon$ of order $\epsilon^{-2}\ln(\epsilon^{-1})$ such that $|\mathcal{B}_{n,\beta,\epsilon}| \leq e^{-\tilde{c}_\epsilon n}|\mathcal{B}_{n,\beta}|$ for each $n \geq \tilde{N}_\epsilon$ and $1/2 \leq \beta \leq 2$. The angular mapping easily implies that $\mathcal{G}_{n,\beta}$ is in bijection with the set of irreducible dissections with n black vertices, $\lfloor\beta n\rfloor$ white vertices, and no outer black vertex of degree 2. The set of such dissections with n black vertices and r white vertices is easily shown to be $\binom{2n}{r}\binom{2r}{n}$ up to rational factors in n and r. Recall that this is also the asymptotic form form of the coefficients $B_{n,r}$ counting black-rooted bicolored binary trees with n black nodes and r white nodes (Equation (34). Hence, there exists a constant $C > 0$ such that $\ln(|\mathcal{B}_{n,\beta}|) \leq \ln(|\mathcal{G}_{n,\beta}|) + C\ln(n)$ for each $n \geq 4$ and $\beta \in [1/2 + 1/n, 2 - 5/n]$. Finally we obtain $|\mathcal{G}_{n,\beta,\epsilon}| \leq e^{-\tilde{c}_\epsilon n}|\mathcal{B}_{n,\beta}| \leq n^C e^{-\tilde{c}_\epsilon n}|\mathcal{G}_{n,\beta}|$ for $n \geq \tilde{N}_\epsilon$ and $\beta \in [1/2+1/n, 2-5/n]$. Then it is easily proved that there exists an integer $\hat{N}_\epsilon$ of order $\tilde{c}_\epsilon^{-1}\ln(\tilde{c}_\epsilon^{-1})$ (i.e., of order $\epsilon^{-2}\ln(\epsilon^{-1})$) such that $n^C e^{-\tilde{c}_\epsilon n} \leq e^{-\tilde{c}_\epsilon n/2}$ for $n \geq \hat{N}_\epsilon$. This concludes the proof, by taking $c_\epsilon := \tilde{c}_\epsilon/2$ and $N_\epsilon := \max(\tilde{N}_\epsilon, \hat{N}_\epsilon)$. □

Proof of Corollary 5.1. Given a suspended internally 3-connected map G with n vertices taken uniformly at random, let β be the random variable giving the ratio faces-vertices of G. Define $\beta_0 = (3 + \sqrt{7})/4$. Using asymptotic results on rooted

3-connected maps [**6**] and Stirling formula, a similar reasoning as in the proof of Lemma 5.6 ensures that, for each fixed $\epsilon > 0$, the ratio vertices-faces β is equal to β_0 in the same exponential sense as in the statement of Theorem 5.6. As a consequence (upon composing the ϵ's), Theorem 5.6 can be applied as if a random n-vertex 3-connected map would always have $\lfloor \beta_0 n \rfloor$ faces. Similarly, for $\epsilon > 0$, a random 3-connected map with m edges has exponentially small probability that the numbers of vertices or faces are outside of $[m(1/2-\epsilon), m(1/2+\epsilon)]$. This implies that Theorem 5.6 can be applied as if the graph would be a random 3-connected map with $\lfloor m/2 \rfloor$ vertices and with a ratio faces-vertices $\beta = 1$. Finally, the result for triangulations easily follows from Theorem 5.6, as triangulations are exactly 3-connected maps for which $\beta = 2 - 5/n$. □

3.2. Grid size of irreducible triangulations. In this section, we explain how the bijection with ternary trees, presented in Chapter 3, gives rise to a precise probabilistic analysis of the grid size of TransversalDraw and CompactTransversalDraw. The whole section is dedicated to the proof of the following result. In all this section, $\mathcal{T}'_n$ denotes the set of rooted irreducible triangulations with n inner vertices.

Theorem 5.7 (Grid size of irreducible triangulations). *Let T be taken uniformly at random in $\mathcal{T}'_n$, T being endowed with its minimal transversal structure. Let $W \times H$ and $W_c \times H_c$ be respectively the size of the grid of* TransversalDraw*(T) and of* CompactTransversalDraw*(T). Then W and H are asymptotically equal to $n/2$ up to fluctuations of order $\sqrt{n}$; and W_c and H_c are asymptotically equal to $11n/27$ up to fluctuations of order $\sqrt{n}$. The same result holds with an ϵ-formulation: for any fixed $\epsilon > 0$, the probability that W or H are outside of $[\frac{n}{2}(1-\epsilon), \frac{n}{2}(1+\epsilon)]$ and the probability that W_c or H_c are outside of $[\frac{11n}{27}(1-\epsilon), \frac{11n}{27}(1+\epsilon)]$ are asymptotically exponentially small.*

We first concentrate on the analysis of W and H, i.e., the width and the height of TransversalDraw(T). This task is rather easy and allows us to introduce some tools —generating functions and the so-called quasi power theorem— that will also be used to analyze the grid size of CompactTransversalDraw(T).

3.2.1. *Analysis of the grid size of* TransversalDraw*(T).* Given $T \in \mathcal{T}'_n$ endowed with its minimal transversal structure, the width of the grid of TransversalDraw(T) is, by definition, the number of inner faces of the red-map T_r. Euler's relation applied to T_r ensures that $e_r = n + f_r + 1$. By definition of the opening (see Section 2.2.2), e_r is equal to the number of red edges (including the stems) of the edge-bicolored ternary tree obtained by doing the opening of T. An easy consequence of the proof of Corollary 3.7 is that the uniform distribution on $\mathcal{T}'_n$ is transported into the uniform distribution on rooted edge-bicolored ternary tree with n nodes (the edge-bicoloration being such that each angle is bicolored). These observations lead to the following statement:

Fact 5.1. *The distribution of the width of* TransversalDraw(T) *for T uniformly sampled in $\mathcal{T}'_n$ is equal to the distribution of $e_r - n + 1$, where e_r is the number of red edges of a uniformly sampled rooted edge-bicolored ternary tree with n nodes.*

We denote by $\mathcal{R}$ and $\mathcal{B}$ the sets of rooted edge-bicolored ternary trees whose root leaf is respectively incident to a red stem and to a blue stem. For a rooted

edge-bicolored ternary tree τ, we denote by $|\tau|$ the number of nodes of τ and by $\xi(\tau)$ the number of red edges of τ. We define the generating functions $R(z,u) = \sum_{\tau\in\mathcal{R}} z^{|\tau|}u^{\xi(\tau)}$ and $B(z,u) = \sum_{\tau\in\mathcal{B}} z^{|\tau|}u^{\xi(\tau)}$ that count the set $\mathcal{R}$ and the set $\mathcal{B}$ with respect to the number of nodes and the number of red edges. The generating function $E(z,u)$ counting rooted edge-bicolored ternary trees with respect to the number of nodes and the number of red edges is thus equal to $R(z,u)+B(z,u)$. The classical decomposition of a rooted ternary tree at the root node into three subtrees is directly translated into the following equation system:

$$\left\{\begin{array}{rcl} R(z,u) &=& u\,(1+B(z,u))\,(u+R(z,u))\,(1+B(z,u)) \\ B(z,u) &=& (u+R(z,u))\,(1+B(z,u))\,(u+R(z,u))\,. \end{array}\right.$$

As this system is polynomial in R, B, z and u, it is easy to derive from it, by algebraic elimination, a trivariate polynomial $P(E,z,u)$ such that the formal power series $P(E(z,u),z,u)$ is equal to 0, which means that $E(z,u)$ is an algebraic series.

A useful tool at this point is an adaptation of the so-called quasi power theorem [**52**, Theo. IX.7, Cor. IX.1], [**42**] for algebraic series. From this theorem, the distribution of the number of red edges in a rooted edge-bicolored ternary tree can be analyzed.

THEOREM 5.8 (Algebraic quasi power theorem). *Let $E(z,u)$ be the generating function of a combinatorial class $\mathcal{E}$, where the variable z marks the size $|\gamma|$ of an object $\gamma\in\mathcal{E}$ and the variable u marks a parameter ξ, i.e. $E(z,u)=\sum_{\gamma\in\mathcal{E}} z^{|\gamma|}u^{\xi(\gamma)}$. Assume that $E(z,u)$ is an algebraic series, i.e., there exists a trivariate polynomial $P(E,z,u)$ with rational coefficients such that $P(E(z,u),z,u)=0$. Consider the polynomial system:*

$$(\mathrm{S}) := \left\{ P(E,z,u)=0, \quad \frac{\partial P}{\partial E}(E,z,u)=0 \right\}.$$

Assume that (S) has a solution $\{\tau,\rho\}$ at $u=1$ such that τ and ρ are positive real values and there exists no other (complex) solution (E,z) of (S) at $u=1$ such that $|z|\le\rho$. Assume further that the derivative-condition *$\{\frac{\partial^2 P}{\partial E^2}(\tau,\rho,1)\neq 0, \frac{\partial P}{\partial z}(\tau,\rho,1)\neq 0\}$ is satisfied. Assume also that the Jacobian of (S) with respect to E and z does not vanish at $(\tau,\rho,1)$, i.e.,*

$$\det\begin{pmatrix} \frac{\partial P}{\partial E}(\tau,\rho,1) & \frac{\partial^2 P}{\partial E^2}(\tau,\rho,1) \\ \frac{\partial P}{\partial z}(\tau,\rho,1) & \frac{\partial^2 P}{\partial z\partial E}(\tau,\rho,1) \end{pmatrix} \neq 0.$$

Then there exists a unique pair of algebraic series $(\tau(u),\ \rho(u))$ such that

$$\left\{ P(\tau(u),\rho(u),u)=0, \frac{\partial P}{\partial E}(\tau(u),\rho(u),u)=0, \tau(1)=\tau, \rho(1)=\rho \right\}.$$

Finally, assume that $\rho(u)$ satisfies the mean condition $\rho'(1)\neq 0$ and variance condition $\rho''(1)\rho(1)+\rho'(1)\rho(1)-\rho'(1)^2\neq 0$. Then, for γ taken uniformly at random among the objects of $\mathcal{E}$ of size n, the random variable $X_n=\xi(\gamma)$ is asymptotically equal to μn, up to Gaussian fluctuations of order $\sigma\sqrt{n}$, where $\mu=-\frac{\rho'(1)}{\rho(1)}$ and $\sigma^2=-\frac{\rho''(1)}{\rho(1)}-\frac{\rho'(1)}{\rho(1)}+\left(\frac{\rho'(1)}{\rho(1)}\right)^2$. In other words,

$$\frac{X_n-\mu n}{\sigma\sqrt{n}} \underset{n\to\infty}{\to} \mathcal{N}(0,1), \quad \text{convergence in law.}$$

where $\mathcal{N}(0,1)$ is a standard Gaussian law. The result also holds with an ϵ-formulation: for any $\epsilon > 0$, the probability that X_n is outside of $[(1-\epsilon)\mu n, (1+\epsilon)\mu n]$ is asymptotically exponentially small.

This theorem, despite a rather long statement and several conditions to check, is easy to apply in practice. In our case, $E(z,1)$ is clearly equal to $2A(z)$ with $A(z) = z(1+A(z))^3$ the generating function of rooted ternary trees. Hence, at $u=1$, we have $P(E,z) = E/2 - z(1+E/2)^3$. Using a computer algebra software, the solution of (S) at $u=1$ is found to be $\{\tau = 1, \rho = 4/27\}$. The derivative condition and the Jacobian condition are then easily checked. The algebraic function $\rho(u)$ is obtained by taking the resultant of the two equations of (S) with respect to E; then the factor $Q(z,u)$ of the resultant such that $Q(\rho,1) = 0$ gives an algebraic equation for $\rho(u)$, i.e., $Q(\rho(u),u) = 0$. From the algebraic equation $Q(\rho(u),u) = 0$, the derivative and second derivative of $\rho(u)$ at $u=1$ are readily calculated. To calculate $\rho'(1)$, we derivate $Q(\rho(u),u) = 0$ and find $\frac{\partial Q}{\partial z}(\rho(u),u)\rho'(u) + \frac{\partial Q}{\partial u}(\rho(u),u) = 0$. Hence $\rho'(1) = -\frac{\partial Q}{\partial u}(\rho,\tau)/\frac{\partial Q}{\partial z}(\rho,\tau)$. From that, the mean condition and variance condition are readily checked and we find $\mu = 3/2$. Hence, the number e_r of red edges in a random rooted edge-bicolored ternary tree is asymptotically equal to $3n/2$, up to fluctuations of order $\sqrt{n}$, and an ϵ-formulation also holds.

This result, together with Fact 5.1, ensure that the width of TransversalDraw(T), for T uniformly sampled in $\mathcal{T}'_n$, is asymptotically equal to $n/2$ up to fluctuations of order $\sqrt{n}$, and an ϵ-formulation also holds. The result is the same for the height H of TransversalDraw(T). Indeed, the distribution of H is the same as the distribution of W, by stability of $\mathcal{T}'_n$ and of TransversalDraw under the $\pi/2$-clockwise rotation. Thus, we have proved the statement of Proposition 5.7 on the distribution of the grid size of TransversalDraw(T).

3.2.2. *Analysis of the grid size of* CompactTransversalDraw*(T).* We focus here on the distribution of the width of CompactTransversalDraw(T), for T taken uniformly at random in $\mathcal{T}'_n$. By definition, the width of CompactTransversalDraw(T) is $W - \Delta_r$, where W is the width of TransversalDraw(T) and Δ_r is the number of red S-patterns of T. We have already proved that W is asymptotically equal to $n/2$ up to fluctuations of order $\sqrt{n}$ and have also given an ϵ-formulation of the result. Hence, to obtain the statement of Proposition 5.7 about W_c, it is sufficient to prove that Δ_r is asymptotically equal to $5n/54$ up to fluctuations of order $\sqrt{n}$ and to prove also an ϵ-formulation of this statement. The steps of the proof are the following. First, the red S-patterns of T are shown to correspond to particular red edges of the ternary tree obtained by doing the opening of T. Then, using generating functions and the algebraic quasi power theorem, it is proved that the number of such edges in a random rooted edge-bicolored ternary tree with n nodes is asymptotically equal to $5n/54$, up to fluctuations of order $\sqrt{n}$, and an ϵ-formulation also holds.

Definition. Given a ternary tree A, an *internal edge* is an inner edge $e = \{v, v'\}$ of A such that the the clockwise-following edge at each extremity of e is an inner edge (not a stem).

Lemma 5.7 (reduction to a parameter on ternary trees). *Let T be an irreducible triangulation and let A be the ternary tree obtained by doing the opening of T, T being endowed with its minimal transversal edge-partition and A being endowed with its edge-bicoloration. Then each central edge of a red (resp. blue) S pattern of T corresponds, via the opening of T, to an internal red (resp. blue) edge of A.*

Proof. Let $e = (v, v')$ be the central edge of a red N-pattern of T. By definition, the ccw-consecutive edge of e at v' is red. Hence the cw-consecutive edge of e at v is blue, otherwise there would be a unicolored face in T, contradicting Lemma 1.1. Similarly, the cw-consecutive edge of e at v' is blue. Hence, by definition of the opening mapping, e is an inner edge of the ternary tree A obtained by doing the opening of T. To prove that e is an internal edge of A, we have to prove that the cw-consecutive edge of A at each extremity of e is an inner edge of A. Let $e_1, \dots, e_k$ be the interval of blue edges of T (in clockwise order) following e in clockwise order around v (the word interval refeering to the terminology of Condition C1); and let $v_1, \dots, v_k$ be the corresponding sequence of neighbours of v. By definition of the opening mapping, the edge of A following e in clockwise order around v is e_k. To prove that e_k is an inner edge of A it remains to show that the edge e' following e_k in clockwise order around v_k is red. If $k = 1$, e' is also the edge following e in counterclockwise order around v'. Hence, the fact that e is internal red ensures that e' is red. If $k \geq 2$, then $e' = (v_k, v_{k-1})$. As (v, v_k) and (v, v_{k-1}) are blue, Lemma 1.1 ensures that (v_k, v_{k-1}) is red. Finally, e_k is an inner edge of A. Similarly, the edge following e in clockwise order around v' is an inner edge of A.

Conversely, let $e = \{v, v'\}$ be an internal red edge of A. Recall that the edge-bicoloration of A is transported into a transversal edge-partition of T (see Section 2.2.1), in such a way that the edge of T following e in ccw order around v (resp. v') has the same color as e iff the extremity of a stem is merged with v (resp. v') over e during the partial closure. We claim that this condition is satisfied by e at v' (and similarly at v). Indeed, the sides of inner edges incident to the outer face at the end of the partial closure are such that the cw-consecutive edge at their right extremity is a stem. Hence, there must be a step at which the right side of e, traversed from v to v', is enclosed into a triangular face due to a local closure. At that step, the stem going over e can not be incident to v, because the edge of A following e in cw order around v is not a stem. Hence $e = \{v, v'\}$ is the second inner edge following the stem, i.e. the extremity of the stem going over e is merged with v'. □

Lemma 5.7 yields the following result that, as Fact 5.2, is an easy consequence of Corollary 3.7.

FACT 5.2. *The distribution of the parameter $\Delta_r(T)$, for T taken uniformly at random in $\mathcal{T}'_n$, is equal to the distribution of the number of internal red edges in a rooted edge-bicolored ternary tree with n nodes taken uniformly at random.*

We denote by X_n the random variable denoting the number of internal red edges in a rooted edge-bicolored ternary tree with n nodes taken uniformly at random. Fact 5.2 and the discussion in the overview ensure that, to prove the statement of Proposition 5.7 about W_c, we have to analyze the distribution of X_n and show that it is concentrated around $5n/54$.

Analysis of the random variable X_n. We introduce the generating functions $F(z, u)$ and $G(z, u)$ counting respectively red-rooted ternary trees and blue-rooted ternary trees with respect to the number of nodes and the number of internal red edges. As for the series $R(z, u)$ and $B(z, u)$ involved in the analysis of TRANSVERSALDRAW, an equation system linking the series $F(z, u)$ and $G(z, u)$ can be obtained by decomposing a rooted ternary tree at its root node into three subtrees.

To keep track of the parameter counting internal red edges, we introduce two auxiliary generating functions; $\widehat{F}(z, u)$ is the series counting red-rooted ternary

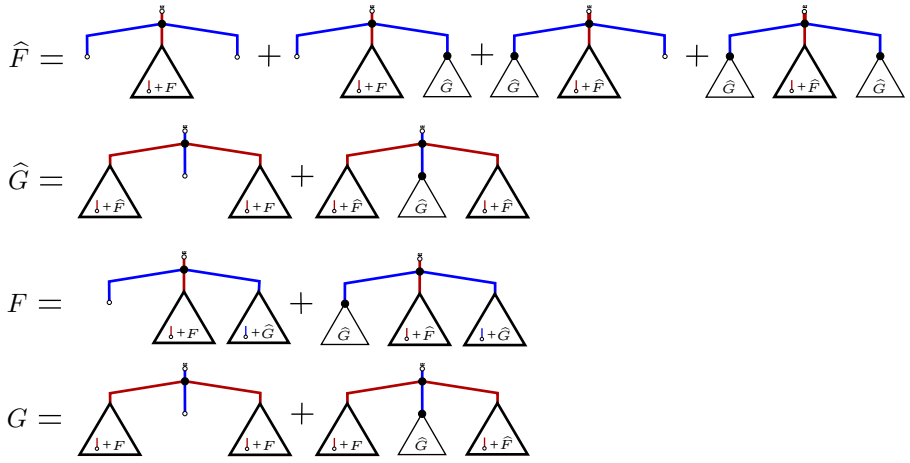


FIGURE 12. Decomposition at the root node keeping track of the number of internal red edges.

trees with respect to the number of nodes and the number of internal red edges, with the difference that root red stem is also considered as an internal red edge if its right child-edge is an inner edge; and $\widehat{G}(z,u)$ is the series counting blue-rooted ternary trees with respect to the number of nodes and the number of internal red edges, with the difference that the left child-edge of the root node is also considered as an internal red edge if it is an inner edge whose right child-edge is an inner edge. The decomposition at the root node keeping track of the number of internal red edges is represented on Figure 12. It is directly translated into the following system:

$$(36)\qquad \begin{cases} \widehat{F}(z,u) &= z(1+F)+z(1+\widehat{F})\widehat{G}+zu(1+F)\widehat{G}+zu\widehat{G}(1+\widehat{F})\widehat{G} \\ \widehat{G}(z,u) &= z(1+\widehat{F})(1+F)+z(1+\widehat{F})\widehat{G}(1+\widehat{F}) \\ F(z,u) &= z(1+F)(1+\widehat{G})+z\widehat{G}(1+\widehat{F})(1+\widehat{G}) \\ G(z,u) &= z(1+F)^2+z(1+F)\widehat{G}(1+\widehat{F}) \end{cases}$$

From that system, an algebraic equation $P(H(z,u),z,u)=0$ can easily be derived for the generating function $H(z,u)=F(z,u)+G(z,u)$ counting rooted edge-bicolored ternary trees with respect to the number of nodes and the number of internal red edges. Then, the algebraic quasi power theorem can be applied on the algebraic generating function $H(z,u)$. All conditions are easily checked and the algebraic series $\rho(u)$ which we obtain verifies $\mu=-\rho'(1)/\rho(1)=5/54$. This yields the statement of Proposition 5.7 on the distribution of W_c. The result is the same for H_c, because the distribution of H_c is equal to the distribution of W_c, by stability of $\mathcal{T}'_n$ and CompactTransversalDraw under the $\pi/2$ clockwise rotation. This concludes the proof of Proposition 5.7.

3.3. Grid size of quadrangulations. Let us only sketch the analysis of the grid size for the straight-line drawing algorithm DrawQuad (described in Section 3.3). The arguments are very close to the ones used to analyse the grid size of irreducible triangulations. The steps are the following, with Q the embedded rooted quadrangulation and M the primal map of Q. First, recall that the bipolar orientation used to compute the uncomplete transversal structure of Q is the minimal

bipolar orientation of M. Then it is easily shown that a face f of Q is not split into two triangles during the partial triangulation step of DrawQuad iff the contour of f is a ccw circuit in the minimal 2-orientation of Q. As a consequence, the parameter Δ to analyse is the number of ccw faces in a rooted quadrangulation with n faces drawn uniformly at random and endowed with its minimal 2-orientation. Next, the parameter Δ is analysed using the bijection presented in Section 1.3, which is more conveniently considered here as a 2-to-n correspondence between rooted quadrangulations and rooted 1-stem plane trees. It is easily shown that the opening turns only one edge of a ccw face into a stem, yielding a configuration (stem,edge,edge,edge) in a ccw walk around the tree, and such that the three edges form a path away from the root. The probabilistic analysis of the number of such patterns in a random rooted 1-stem plane tree is done again by defining bivariate generating functions and applying the algebraic quasi-power theorem to the system satisfied by these generating functions.

Theorem 5.9 (Grid size of quadrangulations). *Let Q be a rooted quadrangulation with n vertices taken uniformly at random. Then the algorithm* DrawQuad*(Q) outputs asymptotically with high probability a grid $13n/27 \times 13/27$,o up to fluctuations of order $\sqrt{n}$. The same result holds with an ϵ-formulation: for any fixed $\epsilon > 0$, the probability that the width or height are outside of $[\frac{13n}{27}(1-\epsilon), \frac{13n}{27}(1+\epsilon)]$ is asymptotically exponentially small.*

4. Appendix: proof of Lemma 5.3

This section is dedicated to the proof of Lemma 5.3, whose statement is: "given a map M with quadrangular outer face and endowed with a transversal structure, the number of unused abscissas of TransversalDraw(M), (i.e., the number of vertical lines of the grid that bear no vertex of M) is equal to the number of red S-patterns of M". By definition of TransversalDraw(M), the line with abscissa 0 is occupied by the vertices S and W and the line with abscissa f_r is occupied by the vertices N and E. Moreover, recall that the abscissa $\mathrm{Abs}(v)$ of an inner vertex v is obtained by associating with v an oriented red path $\mathcal{P}_r(v)$ (called separating red path of v) and then counting the number of inner faces of M_r on the left of $\mathcal{P}_r(v)$. From that it follows that $1 \leq \mathrm{Abs}(v) \leq f_r - 1$. Hence, the lines $x = 0$ and $x = f_r$ are occupied and, for $1 \leq i \leq f_r - 1$, the line with abscissa i is not used iff there exists no inner vertex of M such that $\mathrm{Abs}(v) = i$. In what follows we show first that each abscissa-candidate $1 \leq i \leq f_r - 1$ can be associated with a unique inner face of M_r, for which we write $\mathrm{Abs}(f) = i$. Then we show that the existence of an inner vertex v of M with $\mathrm{Abs}(v) = i$ only depends on the configuration of the edge at the bottom-right corner of f; the candidate is unoccupied iff this edge is the central edge of a red S-pattern.

Let us start with a few definitions. Given e an inner edge of M_r, the *separating red path* of e, denoted $\mathcal{P}_r(e)$, is defined as the path obtained by concatenating the leftmost outgoing red path of the end-vertex of e, the edge e, and the rightmost ingoing red path of the origin of e. Given an inner face f of M_r, we recall that there are two vertices S_f and N_f of f such that the contour of f consists of a left and a right lateral paths going from S_f to N_f. The *separating red path* of f, denoted $\mathcal{P}_r(f)$, is defined as the concatenation of the leftmost outgoing red path of N_f, of the right lateral path P_2 of f, and of the rightmost ingoing red path of S_f. Define $\mathrm{Abs}(f)$ as the number of faces of M_r on the left of $\mathcal{P}_r(f)$. Observe that the first edge e_f of P_2, called bottom right edge of f, satisfies $\mathcal{P}_r(f) = \mathcal{P}_r(e_f)$. The following facts easily follow from the definition of separating red path of a vertex, and prove useful to show that separating paths do not cross each other.

FACT 5.3. *Let v and v' be two different inner vertices of M. Let $\mathcal{P}_r^{\text{left}}(v)$, $\mathcal{P}_r^{\text{left}}(v')$, $\mathcal{P}_r^{\text{right}}(v)$ and $\mathcal{P}_r^{\text{right}}(v')$ be respectively the leftmost outgoing red paths and rightmost ingoing red paths of v and v'. Then*

- *The paths $\mathcal{P}_r^{\text{left}}(v)$ and $\mathcal{P}_r^{\text{left}}(v')$ do not cross each other, they join at a vertex v'' and then are equal between v'' and N.*
- *The paths $\mathcal{P}_r^{\text{right}}(v)$ and $\mathcal{P}_r^{\text{right}}(v')$ do not cross each other, they join at a vertex v'' and then are equal between v'' and S.*
- *The paths $\mathcal{P}_r^{\text{left}}(v)$ and $\mathcal{P}_r^{\text{right}}(v')$ do not cross each other. In addition, $\mathcal{P}_r^{\text{right}}(v')$ can not meet $\mathcal{P}_r^{\text{left}}(v)$ from the left of $\mathcal{P}_r^{\text{left}}(v)$.*

These facts imply easily that the separating red paths $\mathcal{P}_r(v)$ and $\mathcal{P}_r(v')$ do not cross each other.

CLAIM 5.1. *Let e and e' be two different red edges of M. Then the separating red paths $\mathcal{P}_r(e)$ and $\mathcal{P}_r(e')$ do not cross each other.*

Proof. Lemma 2.5 ensures that an edge e' connecting two vertices of an oriented red path is on the path. Hence only three cases can arise: either e' and $\mathcal{P}_r(e)$ do not intersect, or they intersect at a unique extremity of e', or e' is on $\mathcal{P}_r(e)$. Fact 5.3 allows us to check that $\mathcal{P}_r(e)$ and $\mathcal{P}_r(e')$ do not cross each other for each of the three cases. □

Recall that the separating red path of a face f is the separating red path of its bottom-right edge. Thus, Claim 5.1 ensures that the separating red paths $\mathcal{P}_r(f)$ and $\mathcal{P}_r(f')$ of two different inner faces f and f' of M_r do not cross each other. In addition, it is easy to see that they are different using the fact that the bottom-right edge of a face is not the leftmost outgoing red edge at its origin. As a consequence, $\text{Abs}(f) \neq \text{Abs}(f')$. There are f_r inner faces in M_r, each inner face f clearly satisfying $1 \leq \text{Abs}(f) \leq f_r$. Hence the pigeonhole principle yields:

FACT 5.4. *For each $1 \leq i \leq f_r$, there exists a unique inner face f of M such that $\text{Abs}(f) = i$.*

Thus, the inner faces of M_r are strictly ordered from left to right according to their associated abscissa.

CLAIM 5.2. *The separating red paths of an inner edge e of the red-map M_r and of an inner vertex v of M do not cross each other. In addition, given e an inner edge of M_r, there exists an inner vertex v of M such that $\mathcal{P}_r(v) = \mathcal{P}_r(e)$ iff either e is the rightmost ingoing red edge at its end-vertex or e is the leftmost outgoing red edge at its origin.*

Proof. The fact that $\mathcal{P}_r(e)$ and $\mathcal{P}_r(v)$ do not cross each other can easily be checked using Fact 5.3. The second statement of the lemma follows from a few observations. Denote by v_1 the origin of e and by v_2 the end-vertex of e. If v is not on $\mathcal{P}_r(e)$ then clearly $\mathcal{P}_r(v)$ is not equal to $\mathcal{P}_r(e)$. If v is on $\mathcal{P}_r(e)$ between v_2 and N, then $\mathcal{P}_r(v) = \mathcal{P}_r(e)$ iff all edges of $\mathcal{P}_r(e)$ between v_1 and v are the rightmost ingoing red edge at their end-vertex. If v is on $\mathcal{P}_r(e)$ between S and v_1, then $\mathcal{P}_r(v) = \mathcal{P}_r(e)$ iff all edges of $\mathcal{P}_r(e)$ between v and v_2 are the leftmost outgoing red edge at their origin. It follows from these observations that $\mathcal{P}_r(v_2) = \mathcal{P}_r(e)$ if e is the rightmost ingoing red edge at its end-vertex, that $\mathcal{P}_r(v_1) = \mathcal{P}_r(e)$ if e is the leftmost outgoing red edge at its origin, and that no vertex v satisfies $\mathcal{P}_r(v) = \mathcal{P}_r(e)$ otherwise. □

Definition. Given a map M with quadrangular outer face and endowed with a transversal structure, a *ccw-internal* edge of M is an inner edge e of M such that the counterclockwise-consecutive edge at each extremity of e has the same color as e. Hence, a ccw-internal red edge is an inner red edge of M that is neither the leftmost outgoing red edge at its origin nor the rightmost ingoing red edge at its end-vertex. Clearly, a red (blue) edge is ccw-internal iff it is the central edge of a red (blue, respectively) S-pattern.

CLAIM 5.3. *The number of abscissas not used by* TRANSVERSALDRAW*(M) is equal to the number of ccw-internal red edges of M. Similarly, the number of ordinates not used by* TRANSVERSALDRAW*(M) is equal to the number of ccw-internal blue edges of M.*

Proof. Let $1 \leq i \leq f_r$ be an abscissa candidate and let f be the unique inner face of M_r such that $\mathrm{Abs}(f) = i$. Recall that the separating red path $\mathcal{P}_r(f)$ is equal to the separating red path of the bottom-right edge e_f of f. Claim 5.2 ensures that i is not the abscissa of any vertex of M iff e_f is a ccw-internal red edge of M. Hence, the number of abscissas not used by TRANSVERSALDRAW(M) is equal to the number of ccw-internal red edges of M that are the bottom-right edge of an inner face of M_r. This quantity is also the number of ccw-internal red edges of M. Indeed, a ccw-internal red edge e is the bottom-right edge of the inner face of M_r on its left, because e is not the leftmost outgoing red edge at its origin. □

This concludes the proof of Lemma 5.3; each unoccupied absciss corresponds to the central edge of a red S-pattern, and each unoccupied ordinate corresponds to the central edge of a blue S-pattern.

Conclusion. This chapter has introduced new straight-line drawing algorithms for irreducible triangulations and quadrangulations, based on transversal structures. Our algorithms are of the face-counting type, like the straight-line drawing algorithm introduced by Schnyder [**102**] for triangulations and improved by Bonichon *et al* [**16**] for any 3-connected maps. Face-counting algorithms have a very simple description (they are easily performed by hand on small examples) and are close to the underlying combinatorics, making it possible to perform a precise probabilistic analysis of the grid size. Our analysis ensures that the grid size used by the algorithms is asymptotically almost surely a square grid (up to small fluctuations). The results are recapitulated in the following table, where the distribution of maps in a family is uniform on the subset with fixed number n of vertices.

Family	Algorithm	Grid size
3-connected	COMPACTSCHNYDERDRAW	$\frac{\sqrt{7}-1}{2}n \times \frac{\sqrt{7}-1}{2}n$
triangulations	COMPACTSCHNYDERDRAW	$\frac{7}{8}n \times \frac{7}{8}n$
irr. triangulations 4-gon	COMPACTTRANSVERSALDRAW	$\frac{11}{27}n \times \frac{11}{27}n$
quadrangulations	DRAWQUAD	$\frac{13}{27}n \times \frac{13}{27}n$

Conclusion et perspectives

Dans cette thèse nous avons exploité quelques jolies propriétés combinatoires des cartes planaires, reposant notamment sur la notion d'α-orientations minimales. Ces propriétés structurelles donnent naissance à des algorithmes très efficaces et élégants sur les cartes planaires. Dans le cas de la génération aléatoire et du codage, les algorithmes découlent de bijections avec des familles d'arbres, ces bijections s'appuyant sur les α-orientations minimales. Dans le cas du dessin, les algorithmes découlent de structures combinatoires (par exemple, les forêts de Schnyder) qui peuvent être formulées comme des α-orientations. Les algorithmes de dessin peuvent ensuite être analysés à l'aide des bijections avec familles d'arbres, comme nous l'avons vu au chapitre 5.

On voit là se former un schéma assez général dans lequel les α-orientations jouent un rôle crucial, à la fois pour énumérer (bijectivement), générer, coder, et dessiner des cartes planaires. Dans une optique de généralisation, il est donc intéressant de considérer les problèmes que nous avons pu résoudre sur certaines α-orientations (e.g. les orientations eulériennes) et regarder dans quel cadre ces résultats pourraient s'étendre.

Ainsi, au chapitre 1, nous avons mentionné le comptage d'orientations bipolaires planes sommées sur toutes les cartes ayant un certain nombre de sommets et de faces. Ce problème nous a été inspiré par la méthode bijective introduite par N. Bonichon [**15**] pour compter le nombre de forêts de Schnyder sommés sur toutes les triangulations ayant un certain nombre de sommets. Dans les deux cas, la solution repose sur une décomposition arborescente de ces structures. Il serait intéressant d'étudier si ces méthodes bijectives se formalisent dans un cadre général de décompositions arborescentes, ce qui permettrait notamment d'espérer compter les structures transversales sommées sur toutes les triangulations irréductibles d'une certaine taille.

Au chapitre 2, nous nous sommes concentrés sur le calcul d'α-orientations et plus particulièrement d'α-orientations minimales. Il y a peu d'espoir qu'un algorithme générique linéaire permette de calculer l'α-orientation minimale étant donnée une fonction α, ou même de trouver une α-orientation s'il en existe une. En effet, le problème se rapproche d'un problème de flot ou encore de calcul d'un couplage parfait, des problèmes qui ont reçu beaucoup d'attention et dont la meilleure complexité connue reste surlinéaire. Cependant, pour les structures qui nous intéressent {orientations eulériennes, orientations bipolaires, forêts de Schnyder, structures transversales}, nous avons décrit des algorithmes linéaires calculant la structure minimale. Ce qui est frappant est que ces algorithmes se ressemblent beaucoup. Ils procèdent itérativement par conquête, la zone conquise augmentant à chaque étape par le choix d'un certain sommet (ou chemin, ou face) sur le front de conquête puis orientation des arêtes qui lui sont incidentes. Dans ces conditions, il est frustrant de devoir à chaque fois recommencer les preuves, qui ont beaucoup

de points communs. Il serait plus satisfaisant de décrire un algorithme générique trouvant la α-orientation minimale pour une fonction α donnée, et qui se spécialise en un algorithme linéaire pour chacune des structures étudiées dans cette thèse. Nous avons actuellement une ébauche d'algorithme, qui repose sur un principe d'épluchage de cycle analogue à celui utilisé pour calculer l'orientation eulérienne minimale.

Les bijections présentées au chapitre 3 appellent à une meilleure compréhension et classification. Il semble par exemple surprenant de ne pas réussir à appliquer la méthode basée sur les α-orientations minimales aux familles de cartes 3-connexes et aux triangulations irréductibles, d'autant que ces deux familles s'énumèrent bien et qu'elles se prêtent à un autre type de bijection. Il semble qu'il existe deux mondes assez distincts: celui des bijections de type enraciné, pour lesquelles l'arbre se retrouve par un parcours en profondeur sur une α-orientation minimale, et celui des bijections de type non enraciné, pour lesquelles l'arbre se retrouve de manière purement locale à partir d'une orientation dérivée d'une α-orientation minimale. À ce titre, la bijection avec arbres étiquetés de Marcus et Schaeffer [**86**] semble rentrer plutôt dans la deuxième catégorie (une fois reformulée de manière adéquate). Comme cette bijection se généralise en genre supérieur, il serait intéressant de réfléchir à une possible extension en genre supérieur des bijections de type non enraciné présentées en Section 2. Une autre extension possible concerne les cartes avec bord, par exemple les triangulations du disque. De manière équivalente, on cherche à énumérer des cartes dont toutes les faces internes ont degré 3 et la face externe a degré fixe $k \geq 3$. Pour les triangulations, une jolie méthode bijective a été mise en place par Poulalhon et Schaeffer [**95**]. L'idée consiste à intégrer la face polygonale dans l'arbre bourgeonnant, ce qui implique que cette face n'est pas la face externe à la fin de la clôture de l'arbre en une carte (on parle alors de carte annelée car il y a deux faces marquées: la face polygonale et la face externe). Cette méthode semble bien s'appliquer aux bijections de type enraciné présentées en Section 1. Il semble un peu plus difficile d'appliquer la méthode aux bijections de type non enraciné, pour le moment nous avons des formulations bijectives qui sont sans doute correctes mais attendent encore une preuve rigoureuse.

Enfin, il reste de nombreuses questions à explorer dans le domaine du dessin de graphes. Nous avons montré que les algorithmes par comptage de faces fournissent une grille plus petite dans le cas moyen que dans le cas le pire. Cependant la taille de la grille reste linéaire par rapport au nombre de sommets. D'autres algorithmes de dessin, qui reposent sur la notion de plus long chemin depuis un sommet source [**78**], ont peut-être un meilleur comportement dans le cas moyen. Il serait intéressant d'effectuer des simulations de ces algorithmes et d'observer la taille moyenne de la grille: par exemple une grille en $cn^\alpha \times cn^\alpha$ pour $\alpha < 1$ correspondrait à une compacité bien meilleure par rapport aux algorithmes par comptage de faces. Reste à savoir si une analyse de tels algorithmes de dessin peut être menée en utilisant nos outils bijectifs, comme cela a été fait pour les algorithmes par comptage de faces.

Bibliography

[1] C. Banderier, P. Flajolet, G. Schaeffer, and M. Soria. Random maps, coalescing saddles, singularity analysis, and airy phenomena. *Random Structures and Algorithms*, 19(3-4):194–246, 2001.

[2] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. Algorithms for drawing graphs: an annotated bibliography. *Computational Geometry: Theory and Applications*, 4:235–282, 1994.

[3] G. Di Battista, R. Tamassia, and L. Vismara. Output-sensitive reporting of disjoint paths. *Algorithmica*, 23(3):302–340, 1999.

[4] R. J. Baxter. Dichromatic polynomials and Potts models summed over rooted maps. *Annals of Combinatorics*, 5:17, 2001.

[5] E. Bender, Z. Gao, and N. Wormald. The number of labeled 2-connected planar graphs. *Electron. J. Combin.*, 9:1–13, 2002.

[6] E. A. Bender. The number of three-dimensional convex polyhedra. *Amer. Math. Monthly*, 94(1):7–21, 1987.

[7] O. Bernardi. Bijective counting of tree-rooted maps and shuffles of parenthesis systems. *Electr. J. Combinatorics*, 14(9), 2006.

[8] O. Bernardi. *Combinatoire des cartes et polynôme de Tutte*. PhD thesis, Université Bordeaux I, 2006.

[9] O. Bernardi. Bijective counting of Kreweras walks and loopless triangulations, 2007. To appear in the Journal of Combinatorial Theory, Series A.

[10] T. Biedl and F. J. Brandenburg. Drawing planar bipartite graphs with small area. In *Proceedings of CCCG, Windsor*, pages 105–108, 2005.

[11] O. Bodini, É. Fusy, and C. Pivoteau. Random sampling of plane partitions. In Renzo Pinzani and Vincent Vajnovszki, editors, *Gascom 2006*, pages 124–135, Dijon, France, 2006. LE2I.

[12] M. Bodirsky, É. Fusy, M. Kang, and S. Vigerske. An unbiased pointing operator for unlabeled structures, with applications to counting and sampling. In *18th ACM-SIAM Symposium on Discrete Algorithms, New Orleans*, pages 356–365, 2007.

[13] M. Bodirsky, C. Groepl, and M. Kang. Generating labeled planar graphs uniformly at random. In *Proceedings of ICALP 2003*, number 2719 in Lecture Notes in Computer Science, pages 1095–1107. Springer Verlag, 2003.

[14] B. Bollobas. *Modern graph theory*, volume 184. Springer, 1998.

[15] N. Bonichon. A bijection between realizers of maximal plane graphs and pairs of non-crossing Dyck paths. *Discrete Mathematics*, 298:104–114, 2005. FPSAC'02 Special Issue.

[16] N. Bonichon, S. Felsner, and M. Mosbah. Convex drawings of 3-connected planar graphs - (extended abstract). In *Graph Drawing: 12th International Symposium, GD 2004*, volume 3383 of *LNCS*, pages 60–70, 2005.

[17] N. Bonichon, C. Gavoille, and N. Hanusse. An information-theoretic upper bound of planar graphs using triangulation. In 20^{th} *Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 2607 of *Lecture Notes in Computer Science*, pages 499–510. Springer-Verlag, 2003.

[18] N. Bonichon, C. Gavoille, N. Hanusse, D. Poulalhon, and G. Schaeffer. Planar graphs, via well-orderly maps and trees. In 30^{th} *International Workshop, Graph - Theoretic Concepts in Computer Science (WG)*, volume 3353 of *Lecture Notes in Computer Science*, pages 270–284. Springer-Verlag, 2004.

[19] D. Boulatov and V. Kazakov. The Ising model on a random planar lattice: the structure of the phase transition and the exact critical exponents. *Phys. Lett.*, B 187:379–384, 1987.

[20] M. Bousquet-Mélou. Four classes of pattern-avoiding permutations under one roof: Generating trees with two labels. *Elect. J. Comb.*, 2002.

[21] M. Bousquet-Mélou and A. Jehanne. Polynomial equations with one catalytic variable, algebraic series, and map enumeration. *Journal of Combinatorial Theory. Series B*, 96:623–672, 2006.

[22] M. Bousquet-Mélou and G. Schaeffer. The degree distribution in bipartite planar maps: applications to the Ising model, 2002. 32pp.

[23] J. Bouttier. *Physique statistique des surfaces aléatoires et combinatoire bijective des cartes planaires*. PhD thesis, Université Paris 6, 2005.

[24] J. Bouttier, P. Di Francesco, and E. Guitter. Combinatorics of hard particles on planar graphs. *Nucl. Phys.*, B 655:313–341, 2003.

[25] E. Brehm. 3-orientations and Schnyder 3-tree-decompositions. Master's thesis, Freie Universitaet Berlin, 2000. `http://www.math.tu-berlin.de/~felsner/Diplomarbeiten/brehm.ps.gz`.

[26] E. Brézin, C. Itzykson, G. Parisi, and J.-B. Zuber. Planar diagrams. *Commun. Math. Phys.*, 59:35–51, 1978.

[27] W.G. Brown. Enumeration of triangulations of the disk. *Proceedings of the London Mathematical Society*, pages 746–768, 1964.

[28] L. Castelli-Aleardi, O. Devillers, and G. Schaeffer. Optimal succinct representations of planar maps. In *Proceedings of the 22th ACM Symposium on Computational Geometry (SoCG)*, pages 309–318, 2006.

[29] R. Cerf and R. Kenyon. The low-temperature expansion of the Wulff crystal in the 3D Ising model. *Comm. Math. Phys.*, 222(1):147–179, 2001.

[30] P. Chassaing and G. Schaeffer. Random planar lattices and integrated superBrownian excursion. *Probab. Theory Related Fields*, 128(2):161–212, 2004.

[31] N. Chiba, T. Nishizeki, S. Abe, and T. Ozawa. A linear algorithm for embedding planar graphs using pq-trees. *J. Comput. Syst. Sci.*, 30(1):54–76, 1985.

[32] M. Chrobak and G. Kant. Convex grid drawings of 3-connected planar graphs. *Int. Journal Compt. Geom. Appl.*, 7(3):211–223, 1997.

[33] R. Cori and B. Vauquelin. Planar maps are well labeled trees. *Canad. J. Math.*, 33(5):1023–1042, 1981.

[34] M. Datar, A. Gionis, P. Indyk, and R. Motwani. Maintaining stream statistics over sliding windows. *SIAM J. Comput.*, 31(6):1794–1813, 2002.

[35] M. de Berg, E. Mumford, and B. Speckmann. On rectilinear duals for vertex-weighted plane graphs. In *GD '05: Proceedings of the Symposium on Graph Drawing*. Springer-Verlag, 2006.

[36] H. de Fraysseix, P. Ossona de Mendez, and J. Pach. A left-first search algorithm for planar graphs. *Discrete Comput. Geom.*, 13:459–468, 1995.

[37] H. de Fraysseix and P. Ossona de Mendez. On topological aspects of orientations. *Discrete Mathematics*, 229(1-3):57–72, 2001.

[38] H. De Fraysseix, P. Ossona de Mendez, and P. Rosenstiehl. Bipolar orientations revisited. *Discrete Appl. Math.*, 56(2-3):157–179, 1995.

[39] H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990.

[40] A. Denise, M. Vasconcellos, and D. J. A. Welsh. The random planar graph. *Congr. Numer.*, 113:61–79, 1996. Festschrift for C. St. J. A. Nash-Williams.

[41] A. Denise and P. Zimmermann. Uniform random generation of decomposable structures using floating-point arithmetic. *Theoretical Computer Science*, 218(2):233–248, 1999.

[42] M. Drmota. Systems of functional equations. *Random Structures and Algorithms*, 10(1-2):103–124, 1997.

[43] P. Duchon, P. Flajolet, G. Louchard, and G. Schaeffer. Boltzmann samplers for the random generation of combinatorial structures. *Combinatorics, Probability and Computing*, 13(4–5):577–625, 2004. Special issue on Analysis of Algorithms.

[44] S. Even and R. E. Tarjan. Computing an st-numbering. *Theoret. Comput. Sci.*, 2:339–344, 1976.

[45] I. Fáry. On straight line representations of planar graphs. *Acta Sci. Math. (Szeged)*, 11:229–233, 1948.

[46] S. Felsner. Convex drawings of planar graphs and the order dimension of 3-polytopes. *Order*, 18:19–37, 2001.

[47] S. Felsner. Geodesic embeddings and planar graphs. *Order*, 20(2):135–150, 2003.

[48] S. Felsner. Lattice structures for planar graphs. *Electron. J. Comb.*, 11(1):Research paper R15, 24p, 2004.

[49] P. Flajolet. Counting by coin tossings. In M. Maher, editor, *Proceedings of ASIAN'04 (Ninth Asian Computing Science Conference)*, volume 3321 of *Lecture Notes in Computer Science*, pages 1–12, 2004. (Text of Opening Keynote Address.).

[50] P. Flajolet, É. Fusy, and C. Pivoteau. Boltzmann sampling of unlabelled structures. In *Proceedings of the fourth Workshop on Analytic Algorithms and Combinatorics*, pages 201–211. SIAM, 2007.

[51] P. Flajolet and A. Odlyzko. Singularity analysis of generating functions. *SIAM J. Discrete Math.*, 3:216–240, 1990.

[52] P. Flajolet and R. Sedgewick. Analytic combinatorics. Available at `http://algo.inria.fr/flajolet/Publications/`. Preliminary version of the forthcoming book.

[53] P. Flajolet, P. Zimmerman, and B. Van Cutsem. A calculus for the random generation of labelled combinatorial structures. *Theoretical Computer Science*, 132(1-2):1–35, 1994.

[54] M. Fleury. Deux problèmes de géométrie élémentaire. *Journal de mathématiques élémentaires*, pages 257–261, 1883.

[55] P.J. Frederico. The number of polyhedra. *Phiipps Res. Repts.*, 30:220–231, 1975.

[56] É. Fusy. Entrelacs, cartes et polytopes: énumération en tenant compte des symétries, 2003. Masters report.

[57] É. Fusy. Quadratic exact size and linear approximate size random generation of planar graphs. *Discrete Mathematics and Theoretical Computer Science*, AD:125–138, 2005.

[58] É. Fusy. Transversal structures on triangulations, with application to straight-line drawing. In *Proceedings of Graph Drawing, LNCS 3843*, pages pp. 177–188, 2005. Full paper with proofs available at `http://arxiv.org/abs/math.CO/0602163`.

[59] É. Fusy. Counting d-polytopes with (d+3) vertices. *Electronic Journal of Combinatorics*, 13(1), 2006.

[60] É. Fusy. Straight-line drawing of quadrangulations. In *Proceedings of Graph Drawing*, 2006.

[61] É. Fusy. Enumeration of unrooted maps using tree-decomposition. *Séminaire Lotharingien de Combinatoire*, B54Al, 2007. 44 pages. Available at `http://www.emis.de/journals/SLC/wpapers/s54Afusy.pdf`.

[62] É. Fusy. A linear approximate-size random sampler for labeled planar graphs, 2007. Available at `http://arxiv.org/abs/0705.1287`.

[63] É. Fusy and F. Giroire. Estimating the number of active flows in a data stream over a sliding window. In *Proceedings of the fourth Workshop on Analytic Algorithms and Combinatorics*, pages 223–231. SIAM, 2007.

[64] É. Fusy, D. Poulalhon, and G. Schaeffer. Dissections and trees, with applications to optimal mesh encoding and to random sampling. In *16th Annual ACM-SIAM Symposium on Discrete Algorithms*, January 2005. Full paper with proofs available at `http://algo.inria.fr/fusy/Articles/FuPoScArticle.pdf`.

[65] É. Fusy, D. Poulalhon, and G. Schaeffer. Bijective counting of plane bipolar orientations, 2007. To appear in the proceedings of Eurocomb'07.

[66] J.-F. Le Gall. The topological structure of scaling limits of large planar maps, 2006. Available at `http://arxiv.org/abs/math/0607567`.

[67] I. Gessel. Super ballot numbers. *J. Symbolic Comput.*, 14(2/3):179–194, 1992.

[68] O. Giménez and M. Noy. Asymptotic enumeration and limit laws of planar graphs, 2004. 14 pages.

[69] F. Giroire. Order statistics and estimating cardinalities of massive data sets. In Conrado Martnez, editor, *International Conference on Analysis of Algorithms*, volume AD of *DMTCS Proceedings*, pages 157–166. Discrete Mathematics and Theoretical Computer Science, 2005.

[70] C. Gotsman. On the optimality of valence-based connectivity coding. *Computer Graphics Forum*, 22(1):99–102, 2003.

[71] B. Grünbaum. *Convex Polytopes*, volume 221 of *Graduate Texts in Math.* Springer-Verlag, New York, 2003. Second edition prepared by V. Kaibel, V. Klee and G. M. Ziegler (original edition: Interscience, London 1967).

[72] F. Harary and E. Palmer. *Graphical Enumeration*. Academic Press, New York, 1973.

[73] X. He. On finding the rectangular duals of planar triangulated graphs. *SIAM J. Comput.*, 22:1218–1226, 1993.

[74] X. He. Grid embedding of 4-connected plane graphs. *Discrete Computational Geometry*, pages 339–358, 1997.

[75] J.E. Hopcroft and R.M. Karp. An o(nsa) algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2:225–231, 1973.

[76] C. Huemer and S. Kappes. A binary labelling for plane laman graphs and quadrangulations. In *Proceedings of EWCG, Delphi*, pages 83–86, 2006.

[77] G. Kant. Drawing planar graphs using the canonical ordering. *Algorithmica*, 16:4–32, 1996. (also *FOCS'92*).

[78] G. Kant and X. He. Regular edge labeling of 4-connected plane graphs and its applications in graph drawing problems. *Theoretical Computer Science*, 172(1-2):175–193, 1997.

[79] A. Khodakovsky, P. Alliez, M. Desbrun, and P. Schroeder. Near-optimal connectivity encoding of polygonal meshes. *Graphical Model*, 64:3–4, 2002.

[80] S. Khuller, J. Naor, and P. N. Klein. The lattice structure of flow in planar graphs. *SIAM J. Discrete Math.*, 6(3):477–490, 1993.

[81] K. Kuratowski. Sur le problème des courbes gauches en topologie. *Fund. Math.*, 15:271–283, 1930.

[82] A. Lempel, S. Even, and I. Cederbaum. An algorithm for planarity testing of graphs. In *Theory of Graphs, Int. Symp (New York)*, pages 215–232, 1967.

[83] K. Lloyd. The number of d-polytopes with $d+3$ vertices. *Mathematika*, 17:120–132, 1970.

[84] J.-F. Marckert and G. Miermont. Invariance principles for random bipartite planar maps, 2005. Available at `http://arxiv.org/abs/math/0504110`.

[85] J.-F. Marckert and A. Mokkadem. Limit of normalized quadrangulations: The brownian map, 2006. Available at `http://arxiv.org/abs/math/0403398`.

[86] M. Marcus and G. Schaeffer. Une bijection simple pour les cartes orientables. Available at `http://www.lix.polytechnique.fr/~schaeffe/Biblio/MaSc01.ps`, Manuscript, 10pp.

[87] M. Mihail and P. Winkler. On the number of eulerian orientations of a graph. In *SODA '92: Proceedings of the third annual ACM-SIAM symposium on Discrete algorithms*, pages 138–145, 1992.

[88] E. Miller. Planar graphs as minimal resolutions of trivariate monomials. *Documenta Math.*, 7:43–90, 2002.

[89] K. Miura, S. Nakano, and T. Nishizeki. Grid drawings of four-connected plane graphs. *Disc. Comput. Geometry*, 26(2):73–87, 2001.

[90] R.C. Mullin and P.J. Schellenberg. The enumeration of c-nets via quadrangulations. *J. Combin. Theory*, 4:259–276, 1968.

[91] A. Nijenhuis and H. S. Wilf. *Combinatorial Algorithms*. Academic Press, second edition, 1978.

[92] P. Ossona de Mendez. *Orientations bipolaires*. PhD thesis, Ecole des Hautes Etudes en Sciences Sociales, Paris, 1994.

[93] D. Poulalhon and G. Schaeffer. A bijection for triangulations of a polygon with interior points and multiple edges. *Theoret. Comput. Sci.*, 307(2):385–401, 2003.

[94] D. Poulalhon and G. Schaeffer. *Chapter of Lothaire: Applied Combinatorics on Words (Encyclopedia of Mathematics and its Applications)*. Cambridge University Press, New York, NY, USA, 2005.

[95] Dominique Poulalhon and Gilles Schaeffer. Optimal coding and sampling of triangulations. *Algorithmica*, 46(3-4):505–527, 2006.

[96] J. Propp. Lattice structure for orientations of graphs, 1993. Manuscript.

[97] J. Propp. Generating random elements of finite distributive lattices. *Elect. J. Comb.*, 4(2), 1997. R15, 12p.

[98] G. Schaeffer. Bijective census and random generation of Eulerian planar maps with prescribed vertex degrees. *Electron. J. Combin.*, 4(1):# 20, 14 pp., 1997.

[99] G. Schaeffer. *Conjugaison d'arbres et cartes combinatoires aléatoires*. PhD thesis, Université Bordeaux I, 1998.

[100] G. Schaeffer. Random sampling of large planar maps and convex polyhedra. In *Annual ACM Symposium on Theory of Computing (Atlanta, GA, 1999)*, pages 760–769 (electronic). ACM, New York, 1999.

[101] W. Schnyder. Planar graphs and poset dimension. *Order*, 5:323–343, 1989.

[102] W. Schnyder. Embedding planar graphs on the grid. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 138–148, San Francisco, California, 1990.

[103] A. Schrijver. *Combinatorial Optimization*. Springer, 2003.
[104] B. Speckmann and M. Van Kreveld. On rectangular cartograms, 2004. Technical report.
[105] R. P. Stanley. *Enumerative combinatorics. Vol. 1*, volume 49 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 1997. With a foreword by Gian-Carlo Rota, Corrected reprint of the 1986 original.
[106] S. K. Stein. Convex maps. In *Proceedings of the American Mathematical Society*, volume 2, pages 464–466, 1951.
[107] R. Tamassia and I. G. Tollis. A unified approach to visibility representations of planar graphs. *Discrete Comput. Geom.*, 1(4):321–341, 1986.
[108] R. Tamassia and I. G. Tollis. Planar grid embedding in linear time. *IEEE Trans. on Circuits and Systems*, CAS-36(9):1230–1234, 1989.
[109] B. A. Trakhtenbrot. Towards a theory of non–repeating contact schemes (russian). In *Trudi Mat. Inst. Akad. Nauk SSSR 51*, pages 226–269, 1958.
[110] W. T. Tutte. A census of Hamiltonian polygons. *Canad. J. Math.*, 14:402–417, 1962.
[111] W. T. Tutte. A census of planar triangulations. *Canad. J. Math.*, 14:21–38, 1962.
[112] W. T. Tutte. A census of slicings. *Canad. J. Math.*, 14:708–722, 1962.
[113] W. T. Tutte. A census of planar maps. *Canad. J. Math.*, 15:249–271, 1963.
[114] W.T. Tutte. How to draw a graph. In *Proceedings London Mathematical Society*, volume 13(3), pages 743–768, 1963.
[115] K. Wagner. Bemerkungen zum Vierfarbenproblem. *Jahresbericht. German. Math.-Verein.*, 46:26–32, 1936.
[116] T. R. S. Walsh. Counting labelled three-connected and homeomorphically irreducible two-connected graphs. *J. Combin. Theory*, 32(B):1–11, 1982.
[117] H. Whitney. 2-isomorphic graphs. *Amer. J. Math.*, 54:245–254, 1933.
[118] H. Zhang and X. He. Compact visibility representation and straight-line grid embedding of plane graphs. In *Proc. WADS'03*, pages 493–504. Lecture Notes in Computer Science, vol. 2748, Springer-Verlag, 2003.
[119] F. Zickfeld and S. Felsner. Schnyder woods and orthogonal surfaces. In *Proceedings of Graph Drawing*, 2006.

Index