



**HAL**  
open science

# Application of graph combinatorics to rational identities of type A

Adrien Boussicault, Valentin Féray

► **To cite this version:**

Adrien Boussicault, Valentin Féray. Application of graph combinatorics to rational identities of type A. 2008. hal-00339049v1

**HAL Id: hal-00339049**

**<https://hal.science/hal-00339049v1>**

Preprint submitted on 15 Nov 2008 (v1), last revised 21 Jan 2009 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Application of graph combinatorics to rational identities of type $A^\dagger$

Adrien Boussicault<sup>1</sup> and Valentin Féray<sup>1</sup>

<sup>1</sup> Université Paris-Est, Institut d'électronique et d'informatique Gaspard-Monge, 77454 Marne-la-Vallée Cedex 2

To a word  $w$ , we associate the rational function  $\Psi_w = \prod (x_{w_i} - x_{w_{i+1}})^{-1}$ . The main object, introduced by C. Greene to generalize identities linked to Murnaghan-Nakayama rule, is a sum of its images by certain permutations of the variables. The sets of permutations that we consider are the linear extensions of oriented graphs. We explain how to compute this rational function, using the combinatorics of the graph  $G$ . We also establish a link between an algebraic property of the rational function (the factorization of the numerator) and a combinatorial property of the graph (the existence of a disconnecting chain).

**Résumé :** À un mot  $w$ , nous associons la fonction rationnelle  $\Psi_w = \prod (x_{w_i} - x_{w_{i+1}})^{-1}$ . L'objet principal, introduit par C. Greene pour généraliser des identités rationnelles liées à la règle de Murnaghan-Nakayama, est une somme de ses images par certaines permutations des variables. Les ensembles de permutations considérés sont les extensions linéaires des graphes orientés. Nous expliquons comment calculer cette fonction rationnelle à partir de la combinatoire du graphe  $G$ . Nous établissons ensuite un lien entre une propriété algébrique de la fonction rationnelle (la factorisation du numérateur) et une propriété combinatoire du graphe (l'existence d'une chaîne le déconnectant).

**Keywords:** Rational functions, posets, maps

## 1 Introduction

A partially ordered set (poset)  $\mathcal{P}$  is a finite set  $V$  endowed with a partial order. By definition, a word  $w$  containing exactly once each element of  $V$  is called a linear extension if the order of the letters is compatible with  $\mathcal{P}$  (if  $a \leq_{\mathcal{P}} b$ , then  $a$  must be before  $b$  in  $w$ ). To a linear extension  $w = v_1 v_2 \dots v_n$ , we associate a rational function :

$$\psi_w = \frac{1}{(x_{v_1} - x_{v_2}) \cdot (x_{v_2} - x_{v_3}) \cdot \dots \cdot (x_{v_{n-1}} - x_{v_n})}.$$

We can now introduce the main object of the paper. If we denote by  $\mathcal{L}(\mathcal{P})$  the set of linear extensions of  $\mathcal{P}$ , then we define  $\Psi_{\mathcal{P}}$  by:

$$\Psi_{\mathcal{P}} = \sum_{w \in \mathcal{L}(\mathcal{P})} \psi_w.$$

<sup>†</sup> this paper is an extended abstract of the eponymous paper on arXiv ???, which contains all detailed proofs.

## 1.1 Background

The linear extensions of posets contain very interesting subsets of the symmetric group : for example, the permutations smaller than a given one for the Bruhat order if the poset is the one considered in the article (3). In this case, our construction is close to that of Demazure characters (4). S. Butler and M. Bousquet-Mélou characterize the permutations corresponding to acyclic posets, which is exactly the cases where the function we consider is the simplest.

Moreover linear extensions are hidden in a recent formula for irreducible character values of the symmetric group: if we use the notations of (7), the quantity  $N^\lambda(G)$  can be seen as a sum over the linear extensions of the bipartite graph  $G$  (bipartite graphs are a particular case of oriented graphs). This explains the similarity of the combinatorics in article (6) and in this one.

The function  $\Psi_{\mathcal{P}}$  was considered considered by C. Greene (8), who wanted to generalize a rational identity linked to Murnaghan-Nakayama rule for irreducible character values of the symmetric group. He has given in his article a closed formula for planar posets ( $\mu_{\mathcal{P}}$  is the Möbius function of  $\mathcal{P}$ ):

$$\Psi_{\mathcal{P}} = \begin{cases} 0 & \text{if } \mathcal{P} \text{ is not connected,} \\ \prod_{y,z \in \mathcal{P}} (x_y - x_z)^{\mu_{\mathcal{P}}(y,z)} & \text{if } \mathcal{P} \text{ is connected,} \end{cases}$$

However, there is no such formula for general posets, only the denominator of the reduced form of  $\Psi_{\mathcal{P}}$  is known (2). In this article, the first author has investigated the effects of elementary transformations of the Hasse diagram of a poset on the numerator of the associated rational function. He has also noticed, that in some case, the numerator is linked with Schur functions (2, paragraph 4.2) (we can also find multiSchur functions or Schubert polynomials).

In this paper, we obtain some new results on this numerator, thanks to a simple local transformation in the graph algebra, preserving linear extensions.

## 1.2 Main results

**An inductive algorithm** The first main result of this paper is an induction relation on linear extensions (Theorem 4.1). When one applies  $\Psi$  on it, it gives an efficient algorithm to compute the numerator of the reduced fraction of  $\Psi_{\mathcal{P}}$  (the denominator is already known).

**A combinatorial formula** If we iterate our first main result in a clever way, we can describe combinatorially the final result. The consequence is our second main result : if we give to the graph of a poset  $\mathcal{P}$  a rooted map structure, we have a combinatorial non-inductive formula for the numerator of  $\Psi_{\mathcal{P}}$  (Theorem 6.5).

**A condition for  $\Psi_{\mathcal{P}}$  to factorize** Green formula's for the function associated to a planar poset is a quotient of products of polynomials of degree 1. In the non-planar case, the denominator is still a product of degree 1 terms, but not the numerator. So we may wonder when the numerator  $N(\mathcal{P})$  can be factorized.

Our third main result is a partial answer (a sufficient but not necessary condition) to this question : the numerator  $N(\mathcal{P})$  factorizes if there is a chain disconnecting the Hasse diagram of  $\mathcal{P}$  (see Theorem 7.1 for a precise statement). An example is drawn on figure 1 (the disconnecting chain is

(2, 5)). Note that we use here and in the whole paper a unusual convention : we draw the posets from left (minimal elements) to right (maximal elements).

$$N \left( \begin{array}{cc} 4 & 6 \\ \bullet & \bullet \\ 2 & 5 \\ \bullet & \bullet \\ 1 & 3 \\ \bullet & \bullet \end{array} \right) = N \left( \begin{array}{cc} 2 & 5 \\ \bullet & \bullet \\ 1 & 3 \\ \bullet & \bullet \end{array} \right) \cdot N \left( \begin{array}{cc} 4 & 6 \\ \bullet & \bullet \\ 2 & 5 \\ \bullet & \bullet \end{array} \right)$$

Fig. 1: Example of chain factorization

### 1.3 Open problems

**Necessary condition for factorization** The conclusion of the factorization Theorem 7.1 is sometimes true, even when the separating path is not a chain : see for example Figure 2 (the path (5, 6, 3) disconnects the Hasse diagram, but is not a chain).

This equality, and many more, can be easily proved using the same method as Theorem 7.1. Can we give a necessary (and sufficient) condition for the numerator of a poset to factorize into a product of numerators of subposets? Are all factorizations of this kind?

**Characterisation of the numerator** Let us consider a poset  $\mathcal{P}$ , which has only minimal and maximal elements (respectively  $a_1, \dots, a_l$  and  $b_1, \dots, b_r$ ). The numerator  $N(\mathcal{P})$  of  $\Psi_{\mathcal{P}}$  is a polynomial in  $b_1, \dots, b_r$  which degree in each variable can be easily bounded. Thanks Proposition 4.4, we see immediately that  $N(G) = 0$  on some affine subspaces of the space of variables. Unfortunately, these vanishing relations and its degree do not characterize  $N(\mathcal{P})$  up to a multiplicative factor. Is there a bigger family of vanishing relations, linked to the combinatorics of the Hasse diagram of the poset, which characterizes  $N(\mathcal{P})$ ?

This question comes from the following observation: for some particular posets, the numerator is a Schubert polynomial and Schubert polynomials are known to be easily defined by vanishing conditions (9).

## 2 Graphs and posets

### 2.1 Conventions

In this paper, we deal with finite *directed graphs*. So we will use the following definition of graphs :

$$N \left( \begin{array}{ccccc} & & 7 & & \\ & 2 & 5 & 8 & \\ \bullet & \bullet & \bullet & \bullet & \\ & 1 & 3 & 6 & 9 \\ & \bullet & \bullet & \bullet & \\ & & 4 & & \end{array} \right) = N \left( \begin{array}{ccc} & 2 & 5 \\ \bullet & \bullet & \bullet \\ & 1 & 3 \\ \bullet & \bullet & \bullet \end{array} \right) \cdot N \left( \begin{array}{ccccc} & & 7 & & \\ & 5 & 8 & & \\ \bullet & \bullet & \bullet & \bullet & \\ & 3 & 6 & 9 & \\ & \bullet & \bullet & \bullet & \\ & & 4 & & \end{array} \right)$$

Fig. 2: An example of factorization, not contained in Theorem 7.1.

- A finite set of vertices  $V$ .
- A subset of edges  $E$  of  $V \times V$  (set of edges). If  $e \in E$ , we will note by  $\alpha(e) \in V$  the first component of  $e$  (called *origin* of  $e$ ) and  $\omega(e) \in V$  its second component (called *end* of  $e$ ). This means that each edge has an orientation.

To make the figures easier to read,  $\alpha(e)$  is always the left-most extremity of  $e$  and  $\omega(e)$  its right-most one (of course it is not possible if the graph contains a circuit, see definition lower, but this case will not be very interesting for our purpose).

Let  $e = (v_1, v_2)$  be an element of  $V \times V$ . Then we denote by  $\bar{e}$  the element  $(v_2, v_1)$ .

With these definitions of graphs, we have four definitions of *paths*:

	can not go backwards	can go backwards
closed	circuit	loop
non-closed	chain	path

**Definition 2.1** *More precisely,*

**chain** A chain is a sequence of edges  $p = (e_1, \dots, e_k)$  of  $G$  such that  $\omega(e_1) = \alpha(e_2)$ ,  $\omega(e_2) = \alpha(e_3)$ ,  $\dots$  and  $\omega(e_{k-1}) = \alpha(e_k)$ .

**circuit** A circuit is a chain  $p = (e_1, \dots, e_k)$  of  $G$  such that  $\omega(e_k) = \alpha(e_1)$ .

**path** A path is a sequence  $L = e_1, \dots, e_h$  of elements of  $E \cup \bar{E}$  such that  $\omega(e_1) = \alpha(e_2)$ ,  $\omega(e_2) = \alpha(e_3)$ ,  $\dots$  and  $\omega(e_{h-1}) = \alpha(e_h)$ . If  $L$  is a path, then we denote  $L \cap E$  by  $E(L)$ .

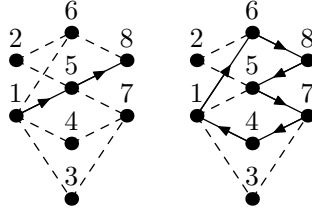
**loop** A loop is a path with the additional property that  $\omega(e_k) = \alpha(e_1)$ .

In all these definitions, we add the condition that all edges and vertices are different (except of course, the equalities in the definition).

**Remark 1** Note that the difference between a loop and a circuit (respectively a path and a chain) is that, in a loop (respectively in a path), an edge can appear in both directions (not only in the direction given by the graph structure). The edges, for which both directions coincide, are exactly the elements of  $E(L)$ .

**Example 1** An example of graph is drawn on figure 3 : the non-dotted edges form a loop  $L$ , such that  $E(L)$  contains 3 edges, whose images by  $\gamma$  are  $(1, 6)$ ,  $(6, 8)$  and  $(5, 7)$ .

The cycle number of a graph  $G$  is  $|E(G)| - |V(G)| + c_G$ , where  $c_G$  is the number of connected components of  $G$ . It is the number of edges that we can remove from  $G$  without increasing  $c_G$ . A connected graph with cycle number 0 is called a tree. Beware that, with this definition, there are no rules for the orientation of the edges of a tree (often, in the literature, an oriented tree is a tree which edges are oriented from the *root* to the *leaves*, but we do not consider such objects here).



**Fig. 3:** Example of a chain and a loop  $L$  in a graph  $G$  (we recall that orientations are from left to right)

## 2.2 Operations on graphs

The main tool of this paper will consist in removing some edges of graph  $G$ .

**Definition 2.2** If  $E'$  is a subset of the set  $E$  of edges of a graph  $G$ , we will denote by  $G \setminus E'$  the graph  $G'$  with

- the same set of vertices as  $G$ .
- the set  $E_{G'} := E \setminus E'$  as set of edges.

**Definition 2.3** If  $G$  is a graph and  $V'$  a subset of its set of vertices  $V$ ,  $V'$  an induced graph structure : its edges are exactly the edges of  $G$ , which have both their extremities in  $V'$ .

If  $V \setminus V' = \{v_1, \dots, v_l\}$ , this graph will also be denoted  $G \setminus \{v_1, \dots, v_l\}$  (the symbol is the same that in definition 2.2, but it should not be confusing in what follows).

**Definition 2.4 (Contraction)** We denote by  $G/e$  the graph (here, the set of edges can be a multiset) obtained by contracting the edge  $e$  (i.e. in  $G/e$ , there is only one vertex  $v$  instead of  $v_1$  and  $v_2$ , the edges of  $G$  different from  $e$  are edges of  $G/e$  : if their origin and/or end in  $G$  is  $v_1$  or  $v_2$ , it is  $v$  in  $G/e$ ).

Then, if  $\alpha(e) \neq \omega(e)$ ,  $G/e$  is a graph with the same number of connected components and the same cycle number as  $G$ .

## 2.3 Poset, graphs, Hasse diagrams and linear extensions

**Definition 2.5** A partially ordered set (poset)  $\mathcal{P}$  is a set endowed with a binary relation  $\leq$  verifying:

- 1) for all  $x \in \mathcal{P}$ ,  $x \leq x$  (reflexivity);
- 2) if  $x \leq y$  and  $y \leq x$ , then  $x = y$  (antisymmetry);
- 3) if  $x \leq y$  and  $y \leq z$ , then  $x \leq z$  (transitivity).

Given a graph  $G$ , we can consider the binary relation :

$$x \leq_1 y \stackrel{\text{def}}{\iff} \left( x = y \text{ or } \exists e \in E(G) \text{ s.t. } \begin{cases} \alpha(e) = x \\ \omega(e) = y \end{cases} \right)$$

This binary relation can be completed by transitivity. If the graph has no circuit, the resulting relation  $\leq$  is antisymmetric and, hence, endows the set  $V(G)$  with a poset structure, which we will denote  $\text{poset}(G)$ .

The application  $\text{poset}$  is not injective. Among the pre-images of a given poset  $\mathcal{P}$ , there is a minimum one (for the inclusion of edge set), which is called Hasse diagram of  $\mathcal{P}$ .

The definition of linear extensions given in the introduction can be formulated in terms of graphs :

**Definition 2.6** *A linear extension of a graph  $G$  is a total order  $\leq_w$  on the set of vertices  $V$  such that : for each edge  $e$  of  $G$ , one has  $\alpha(e) \leq_w \omega(e)$ .*

The set of linear extension of  $G$  is denoted  $\mathcal{L}(G)$ . Let us also define the formal sum  $\varphi(G) = \sum_{w \in \mathcal{L}(G)} w$ .

We will often see a total order  $\leq_w$  defined by  $v_{i_1} \leq_w v_{i_2} \leq_w \dots \leq_w v_{i_n}$  as a word  $w = v_{i_1} v_{i_2} \dots v_{i_n}$ .

**Remark 2** *If  $G$  contains a circuit, then it has no linear extensions. Else, its linear extensions are the linear extensions of  $\text{poset}(G)$ . Thus considering graphs instead of posets give not more general results. But we have chosen to formulate all our results in terms of graphs because they involve the combinatorics of the graphs and, moreover, the set of Hasse diagrams is not closed by the operations of subsection 2.2.*

The following lemma comes straight forward from the definition :

**Lemma 2.1** *Let  $G$  and  $G'$  be two graphs with the same set of vertices.*

$$E(G) \subset E(G') \text{ and } w \in \mathcal{L}(G') \implies w \in \mathcal{L}(G)$$

$$w \in \mathcal{L}(G) \text{ and } w \in \mathcal{L}(G') \iff w \in \mathcal{L}(G \vee G'),$$

$$\text{where } \begin{cases} V(G \vee G') = V(G) = V(G') \\ E(G \vee G') = E(G) \cup E(G') \end{cases} .$$

## 3 Rational functions on graphs

### 3.1 Definition

Given a graph  $G$  with  $n$  vertices  $v_1, \dots, v_n$ , we are interested in the following rational function  $\Psi(G)$  in the variables  $(x_{v_i})_{i=1 \dots n}$  :

$$\Psi(G) = \sum_{w \in \mathcal{L}(G)} \frac{1}{(x_{w_1} - x_{w_2}) \dots (x_{w_{n-2}} - x_{w_{n-1}})}$$

We also consider the renormalization:

$$N(G) := \Psi(G) \cdot \prod_{e \in E_G} (x_{\alpha(e)} - x_{\omega(e)}).$$

In fact, we will see later that it is a polynomial. Moreover, if  $G$  is the Hasse diagram of a poset,  $\varphi(G) =$

$$\frac{N(G)}{\prod_{e \in E_G} (x_{\alpha(e)} - x_{\omega(e)})}$$
 is a reduced fraction.

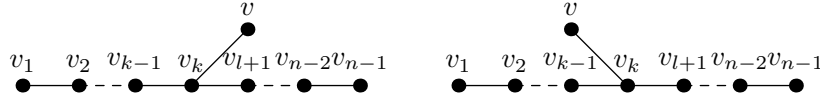


Fig. 4: Graph of proof of Lemma 3.1

### 3.2 Value on forests

Thanks to the following lemma, it is easy to compute  $N$  on forests (note that this results are already present in (2), but the demonstrations here are simpler and make this article self-contained).

**Lemma 3.1 (Pruning-invariance)** *Let  $G$  be a graph with a vertex  $v$  of valence 1 and  $e$  the edge of extremity (origin or end)  $v$ . Then one has*

$$N(G) = N(G \setminus \{v\}).$$

**Proof:** It is enough to prove it in the case where  $G$  has the form of Figure 4. Indeed, the linear extensions of  $G$  are exactly the words obtained from a linear extension of  $G \setminus \{v\}$  by adding  $v$  after (resp. before) the other extremity of  $e$ . If the proposition is true if  $G \setminus \{v\}$  is the graph of a totally ordered set, *i.e.* a line, it will be true for any graph  $G$ .

Suppose that  $G$  is the left-hand graph of Figure 4 (it is similar for the right-hand one). Then

$$\begin{aligned} \mathcal{L}(G) &= \{v_1 \dots v_i v v_{i+1} \dots v_{n-1}, i \geq k\} \\ \text{So } \Psi(G) &= \sum_{i=k}^{n-1} \frac{1}{\left[ \begin{array}{c} (v_1 - v_2) \dots (v_{i-1} - v_i)(v_i - v) \\ \cdot (v - v_{i+1})(v_{i+1} - v_{i+2}) \dots (v_{n-2} - v_{n-1}) \end{array} \right]} \\ &= \frac{1}{(v_1 - v_2) \dots (v_{i-1} - v_i)(v_i - v_{i+1})(v_{i+1} - v_{i+2}) \dots (v_{n-2} - v_{n-1})} \\ &\quad \cdot \left[ \sum_{i=k}^{n-2} \left( \frac{1}{v - v_{i+1}} - \frac{1}{v_i - v} \right) + \frac{1}{v_{n-1} - v} \right] \\ &= \frac{1}{(v_1 - v_2) \dots (v_{n-2} - v_{n-1})} \frac{1}{v - v_k} \end{aligned}$$

The computation gives  $N(G) = 1 = N(G \setminus \{v\})$ , so the proposition is proved.  $\square$

**Proposition 3.2** *If  $T$  is a tree and  $F$  a disconnected forest, one has :*

$$N(T) = 1; \tag{1}$$

$$N(F) = 0. \tag{2}$$



**Proof:** Thanks to the pruning Lemma 3.1, we only have to prove it in the case where  $F$  is a disjoint union of  $n$  points. If  $n = 1$ , it is obvious that  $N(\cdot) = \Psi(\cdot) = 1$ . Else, if we denote by  $c$  the full cycle  $(1 \dots n)$ , one has :

$$\begin{aligned}
\Psi(F) &= \sum_{\sigma \in S(n)} \frac{1}{(x_{\sigma(1)} - x_{\sigma(2)}) \dots (x_{\sigma(n-1)} - x_{\sigma(n)})} \\
&= \frac{1}{n} \sum_{\sigma \in S(n)} \sum_{i=0}^{n-1} \frac{1}{(x_{\sigma \circ c^i(1)} - x_{\sigma \circ c^i(2)}) \dots (x_{\sigma \circ c^i(n-1)} - x_{\sigma \circ c^i(n)})} \\
&= \frac{1}{n} \sum_{\sigma \in S(n)} \frac{\sum_{i=0}^{n-1} x_{\sigma \circ c^i(n)} - x_{\sigma \circ c^i(1)}}{(x_{\sigma(1)} - x_{\sigma(2)}) \dots (x_{\sigma(n-1)} - x_{\sigma(n)})(x_{\sigma(n)} - x_{\sigma(1)})} \\
&= 0.
\end{aligned}$$

□

## 4 The main transformation

In section 3, we have introduced a rational function associated to graphs and we have computed its value on forests. In this section, we will see that the function associated to any graph with at least a cycle can be written using functions associated to graphs with a strictly lower cycle number. So we can study  $\Psi$  using only this induction relation and the value on forests.

### 4.1 Equality on linear extensions

If  $G$  is a finite graph and  $L$  a loop of  $G$ , let us denote by  $T_L(G)$  the following formal alternate sum of subgraphs of  $G$  :

$$T_L(G) = \sum_{\substack{E' \subset E(L) \\ E' \neq \emptyset}} (-1)^{|E'|-1} G \setminus E'.$$

The function  $\varphi(G) = \sum_{w \in \mathcal{L}(G)} w$  can be extended by linearity to the free abelian group spanned by graphs, then one has the following theorem:

**Theorem 4.1** *Let  $G$  be a graph and  $L$  a loop of  $G$  then,*

$$\varphi(G) = \varphi(T_L(G)). \quad (3)$$

Note that all graphs appearing in the right-hand side of (3) have strictly less loops than  $G$ . An example is drawn on figure 5 (to make it easier to read, we did not write the operator  $\varphi$  in front of each graph).

**Remark 3** *In the case where  $E(L) = \emptyset$ , this theorem says that graphs with oriented circuits have no linear extensions (see remark 2).*

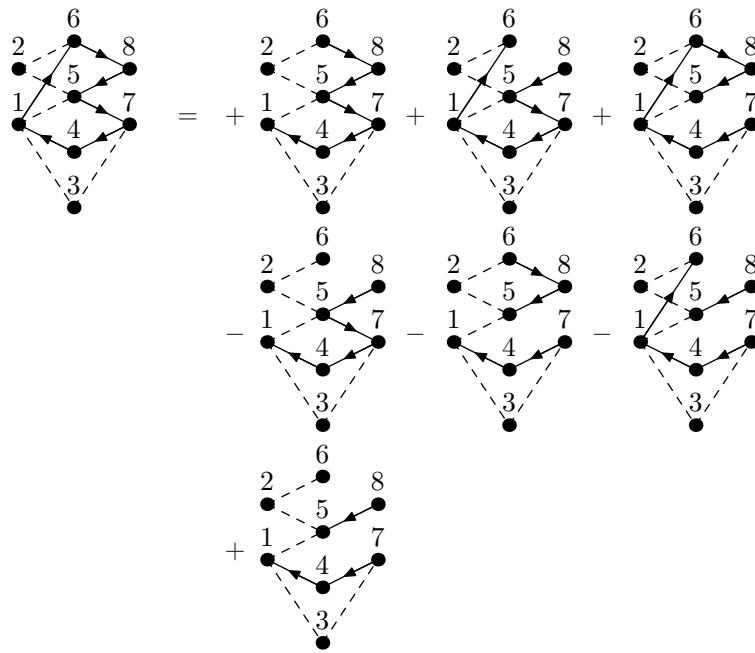


Fig. 5: Example of application of theorem 4.1

If it is a singleton, it says that we do not change the set of linear extensions by erasing an edge if there is a path going from its origin to its end (thanks to transitivity).

An other very interesting case of our relation is the following one. Let  $G$  be a graph and  $v_1$  and  $v_2$  two vertices of  $G$  which are not linked by an edge. We can write

$$\varphi(G) = \sum_{\substack{w \in \mathcal{L}(G) \\ v_1 \leq_w v_2}} w + \sum_{\substack{w \in \mathcal{L}(G) \\ v_2 \leq_w v_1}} w \quad (4)$$

This is a special case of our relation on the graph  $G'$  obtained from  $G$  by adding two edges  $e_{1,2} = (v_1, v_2)$  and  $e_{2,1} = (v_2, v_1)$ . This graph contains a circuit so  $\varphi(G') = 0$ . But one also has :

$$\varphi(G') = \varphi(G \cup \{e_{1,2}\}) + \varphi(G \cup \{e_{2,1}\}) - \varphi(G).$$

So  $\varphi(G)$  is the sum of two terms corresponding exactly to equation (4). By iterating this transformation, deleting graph with circuits and erasing edges thanks to transitivity relation we obtain:

$$\varphi(G) = \sum_{w \in \mathcal{L}(G)} \varphi \left( \begin{array}{c} w_1 \longrightarrow w_2 \cdots \cdots \cdots w_{n-1} \longrightarrow w_n \\ \hline \end{array} \right).$$

An immediate consequence is that any relation between the  $\varphi(G)$  can be deduced from Theorem 4.1.

To prove Theorem 4.1, we will need the two following lemma:

**Lemma 4.2** Let  $w \in \mathcal{L}(G \setminus E(L))$ . There exists  $E'(w)$  such that

$$\forall E'' \subset E(L), \quad w \in \mathcal{L}(G \setminus E'') \iff E'(w) \subset E'' \subset E(L).$$

**Proof:** immediate consequence of lemma 2.1. □

**Lemma 4.3** Let  $w \in \mathcal{L}(G \setminus E(L))$ , there exists  $E'' \subsetneq E(L)$  such that

$$w \in \mathcal{L}(G \setminus E'').$$

**Proof:** Suppose that we can find a word  $w$  for which the lemma is false. Since  $w \in \mathcal{L}(G \setminus E(L))$ , the word  $w$  fulfills the relations of the edges of  $L$ , which are not in  $E(L)$ .

But, if  $e \in E(L)$ , one has  $w \notin \mathcal{L}(G \setminus (E(L) \setminus \{e\}))$ . That means that  $w$  does not fulfill the relation corresponding to the edge  $e$ . As  $w$  is a total order, it fulfills the opposite relation :

$$w \in \mathcal{L} \left[ (G \setminus E(L)) \cup \overline{e} \right].$$

Doing the same argument for each  $e \in E(L)$ , one has

$$w \in \mathcal{L} \left[ (G \setminus E(L)) \cup \overline{E(L)} \right].$$

But this graph contains an oriented cycle so the corresponding set of linear extension is empty. □

Let us come back to the proof of Theorem 4.1.

Let  $w$  be a word containing exactly once each element of  $V(G)$ . We will compute his coefficient in  $\varphi(G) - \varphi(T_L(G)) = \sum_{E' \subset E(L)} (-1)^{|E'|} \varphi(G \setminus E')$  :

- If  $w \notin \mathcal{L}(G \setminus E(L))$ , his coefficient is zero in each summand.
- If  $w \in \mathcal{L}(G \setminus E(L))$ , thanks Lemma 4.2, we know that there exists  $E'(w) \subset E(L)$  such that

$$w \in \mathcal{L}(G \setminus E'') \iff E'(w) \subset E'' \subset E(L).$$

So the coefficient of  $w$  in  $\varphi(G) - \varphi(T_L(G))$  is

$$\sum_{E'(w) \subset E'' \subset E(L)} (-1)^{|E''|} = 0 \text{ (because } E'(w) \neq E(L), \text{ Lemma 4.3).}$$

## 4.2 Consequences on Green functions

**Proposition 4.4** *Let  $G$  be the graph containing a loop  $L$ . Then,*

$$N(G) = \sum_{\substack{E' \subset E(L) \\ E' \neq \emptyset}} \left[ (-1)^{|E'|-1} N(G \setminus E') \prod_{e \in E'} (x_{\alpha(e)} - x_{\omega(e)}) \right].$$

By proposition 3.2, one has  $N(T) = 1$  if  $T$  is a tree and  $N(F) = 0$  if  $F$  is a disconnected forest. So this Proposition gives us an algorithm to compute  $N(G)$  : we just have to iterate it with any loops until all the graphs in the right hand side are forests. So, if after iterating transformations of type  $T_L$  on  $G$ , we obtain  $\sum c_F F$ , then:

$$N(G) = \sum_{F \text{ subforest of } G} c_F \prod_{e \in E_G \setminus E_F} (x_{\alpha(e)} - x_{\omega(e)}).$$

In this formula,  $N(G)$  appears as a sum of polynomials. So the computation of  $N(G)$ , using this formula, is easier than a direct application of the definition

$$N(G) = \sum_{w \in \mathcal{L}(G)} \left( \Psi(w) \cdot \prod_{e \in E_G} (x_{\alpha(e)} - x_{\omega(e)}) \right),$$

where the summands may have poles.

We will use this algorithm in the next section on some examples. But it has also a theoretical interest : some properties of  $N$  on forests can be immediately extended to any graph.

**Corollary 4.5** *For any graph  $G$ , the rational function  $N(G)$  is a polynomial. Moreover, if  $G$  is disconnected,  $N(G) = 0$ .*

**Proof:** We make an immediate induction with Proposition 4.4 over the number of loops of  $G$ . Thanks proposition 3.2, the result is true if  $G$  has no loop.

□

In fact, if  $G$  is the Hasse diagram of a connected poset, the fraction  $\Psi(G) = \frac{N(G)}{\prod_{e \in E_G} (x_{\alpha(e)} - x_{\omega(e)})}$  is irreducible (see (2) for a proof of this fact).

The following result can also be proved by induction on the cycle number:

**Proposition 4.6** *Let  $G$  be a graph and  $e$  an edge of  $G$  between two vertices  $v_1$  and  $v_2$ . Then*

$$N(G/e) = N(G)|_{x_{v_1}=x_{v_2}=x_v},$$

where  $v$  is the contraction of  $v_1$  and  $v_2$  in  $G/e$ .

**Proof Proof (by induction on the cycle number of  $G$ ):** If  $G$  is a tree, then the equality is obvious by Proposition 3.2.

If  $G/e$  contains a loop  $L_e$ , then we consider the following loop  $L$  in  $G$ :

- If  $L_e$  does not go through the vertex  $v$  (contraction of  $v_1$  and  $v_2$ ), then  $L_e$  can also be seen as a loop  $L$  of  $G$ .
- Suppose that  $v$  is the end of  $e_i$  and the origin of  $e_{i+1}$  and that they are also the same vertex ( $v_1$  or  $v_2$ ) in  $G$ . Then,  $L_e$  can still be seen as a loop  $L$  of  $G$ .
- Suppose that  $v$  is the end of  $e_i$  and the origin of  $e_{i+1}$  but that these two edges have different extremities ( $v_1$  and  $v_2$ ) in  $G$ . Then we add the edge  $e$  or  $\bar{e}$  to  $L_e$  (between  $e_i$  and  $e_{i+1}$ ) to obtain a loop  $L$  of  $G$ .

Eventually by changing the orientations of  $L_e$  and  $L$ , we can assume that  $e \notin E(L)$  and, as a consequence  $E(L) = E(L_e)$ . By theorem 4.4, one has :

$$\begin{aligned} N(G/e) &= \sum_{\substack{E' \subset E(L_e) \\ E' \neq \emptyset}} (-1)^{|E'|-1} N((G/e) \setminus E'). \prod_{e \in E'} (x_{\alpha(e)} - x_{\omega(e)}) \\ N(G) &= \sum_{\substack{E' \subset E(L) \\ E' \neq \emptyset}} (-1)^{|E'|-1} N(G \setminus E'). \prod_{e \in E'} (x_{\alpha(e)} - x_{\omega(e)}). \end{aligned}$$

As  $e \notin E(L)$ ,

$$(G \setminus E')/e = (G/e) \setminus E' \text{ and } E(L_e) = E(L).$$

This ends the proof by applying the induction hypothesis to the graphs  $G \setminus E'$ . □

An other immediate consequence of Proposition 4.4 is the vanishing relations of  $N(G)$ .

**Corollary 4.7** *Let  $G$  be a graph. Let  $L$  be a loop of  $G$  with  $E(L) = \{e_1, \dots, e_r\}$ . One has*

$$N(G)|_{x_{\alpha(e_i)}=x_{\omega(e_i)}, i=1 \dots r} = 0$$

Unfortunately, this corollary, written for every loop of a graph  $G$ , does not characterize  $N(G)$  up to a multiplicative factor (see paragraph 1.3).

## 5 Some explicit computations of rational functions

### 5.1 Graphs of cycle number 1.

We will consider connected graphs  $G$  with only one loop. Using pruning Lemma 3.1, we can suppose that each vertex of  $G$  has valence 2. We denote by  $\max(G)$  (resp.  $\min(G)$ ) the set of maximal (resp. minimal) elements of  $G$ . The following result was already proved in (2), but we present here a simpler proof using our operation.

**Proposition 5.1** *If  $G$  is a connected graph with vertices of valence 2, then*

$$N(G) = \sum_{v \in \min G} x_v - \sum_{v' \in \max(G)} x_{v'}.$$

**Proof:**  $G$  has only one loop  $L$  (we only have to choose the orientation). In the right-hand side of equation (3), we have two kinds of terms :

- If  $|E'| = 1$ ,  $G \setminus E'$  is a tree and  $N(G \setminus E') = 1$ .
- If  $|E'| > 1$ ,  $G \setminus E'$  is a disconnected forest and  $N(G \setminus E') = 0$ .

Then

$$N(G) = \sum_{e \in E(L)} (x_{\alpha(e)} - x_{\omega(e)})$$

The sum above can be simplified and is equal to  $\sum_{v \in \min G} x_v - \sum_{v' \in \max(G)} x_{v'}$ . □

### Example 2

$$N \left( \begin{array}{c} \bullet_2 \quad \bullet_4 \\ \diagdown \quad \diagup \\ \bullet_3 \\ \diagup \quad \diagdown \\ \bullet_1 \quad \bullet_5 \end{array} \right) = (x_1 - x_3) N \left( \begin{array}{c} \bullet_2 \quad \bullet_4 \\ \diagdown \quad \diagup \\ \bullet_3 \\ \diagup \quad \diagdown \\ \bullet_1 \end{array} \right) + (x_2 - x_4) N \left( \begin{array}{c} \bullet_2 \quad \bullet_4 \\ \diagdown \quad \diagup \\ \bullet_3 \\ \diagup \quad \diagdown \\ \bullet_5 \end{array} \right) \quad (5)$$

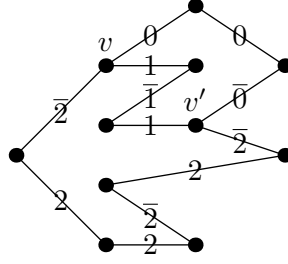
$$+ (x_4 - x_5) N \left( \begin{array}{c} \bullet_2 \quad \bullet_4 \\ \diagdown \quad \diagup \\ \bullet_3 \\ \diagup \quad \diagdown \\ \bullet_1 \end{array} \right) \pm N \left( \begin{array}{c} \text{disconnected} \\ \text{graphs} \end{array} \right) \quad (6)$$

$$= (x_1 - x_3) + (x_2 - x_4) + (x_4 - x_5) \quad (7)$$

$$= x_1 + x_2 - x_3 - x_5. \quad (8)$$

### 5.2 Graphs with cycle number 2.

Let  $G$  be a connected graph with a cycle number equal to 2. Thanks to pruning Lemma 3.1, we can suppose that  $G$  has no vertices of valence 1. As  $|E_G| = |V_G| + 1$ , the graph has, in addition of vertices of valence 2, either two vertices  $v$  and  $v'$  of valence 3 or one vertex of valence 4. We will only look here of the case where there are two vertices of valence 3 and the edges can be partitioned into three paths  $p_0$ ,  $p_1$  and  $p_2$  from  $v$  to  $v'$  (the other cases are easier because the loops have no edges in common).



**Fig. 6:** Example of a graph  $G$  with cycle number 2.

For  $i = 0, 1, 2$ , let us denote by  $E_i$  (resp. by  $E_{\bar{i}}$ ) the set of edges of the path  $p_i$  which appear in the same (resp. opposite) orientation in the graph and in the path  $p_i$  (see the figure 6 for an example, we have written of each edge the index of the set it belongs to). If  $I = \{i_1, \dots, i_l\} \subset \{0, 1, 2, \bar{0}, \bar{1}, \bar{2}\}$ , we consider the following alternate sum of graphs :

$$G_I = \sum_{\emptyset \neq E'_1 \subset E_{i_1}, \dots, \emptyset \neq E'_l \subset E_{i_l}} (-1)^{|E'_1|-1} \dots (-1)^{|E'_l|-1} G \setminus (E'_1 \cup \dots \cup E'_l).$$

Applying Theorem 4.1 with the loop  $L = \bar{p}_1 \cdot p_2$  (with  $E(L) = E_{\bar{1}} \cup E_2$ ), we obtain :

$$\varphi(G) = \varphi(G_{\bar{1}}) + \varphi(G_2) - \varphi(G_{2, \bar{1}}).$$

In each graph in  $G_{\bar{1}}$ , the path  $p_1$  is cut but the two other paths still exist, so they all contain the loop  $\bar{p}_0 \cdot p_2$ . If we apply Theorem 4.1 with this loop, we obtain:

$$\begin{aligned} \varphi(G_{\bar{1}}) &= \sum_{E' \subset E_{\bar{1}}} (-1)^{|E'|-1} \varphi(G \setminus E') \\ &= \sum_{E' \subset E_{\bar{1}}} (-1)^{|E'|-1} \left( \sum_{E'' \subset E_{\bar{0}}} (-1)^{|E''|-1} \varphi((G \setminus E') \setminus E'') \right. \\ &\quad \left. + \sum_{E'' \subset E_2} (-1)^{|E''|-1} \varphi((G \setminus E') \setminus E'') \right) \\ &\quad - \sum_{\substack{E'' \subset E_2 \\ E''' \subset E_{\bar{0}}}} (-1)^{|E''|-1} (-1)^{|E'''|-1} \varphi((G \setminus E') \setminus (E'' \cup E''')) \\ &= \varphi(G_{\bar{0}, \bar{1}}) + \varphi(G_{2, \bar{1}}) - \varphi(G_{2, \bar{0}, \bar{1}}) \end{aligned}$$

In a similar way, all graphs in  $G_2$  contains the loop  $p_1 \cdot \bar{p}_0$  and one has  $\varphi(G_2) = +\varphi(G_{2, \bar{0}}) + \varphi(G_{1, 2}) - \varphi(G_{1, 2, \bar{0}})$ . The graphs in  $G_{2, \bar{1}}$  have no loops, so, finally:

$$\begin{aligned} \varphi(G) &= \varphi(G_{\bar{0}, \bar{1}}) + \varphi(G_{2, \bar{1}}) - \varphi(G_{2, \bar{0}, \bar{1}}) \\ &\quad + \varphi(G_{2, \bar{0}}) + \varphi(G_{1, 2}) - \varphi(G_{1, 2, \bar{0}}) - \varphi(G_{2, \bar{1}}); \\ &= \varphi(G_{\bar{0}, \bar{1}}) - \varphi(G_{2, \bar{0}, \bar{1}}) + \varphi(G_{2, \bar{0}}) + \varphi(G_{1, 2}) - \varphi(G_{1, 2, \bar{0}}). \end{aligned}$$

If we apply  $\Psi$  to this equality, we keep only connected graphs and obtain :

$$\Psi(G) = \Psi(G'_{\overline{0},\overline{1}}) + \Psi(G'_{2,\overline{0}}) + \Psi(G'_{1,2}),$$

where  $G'_I = \sum_{e_1 \in E_{i_1}, \dots, e_l \in E_{i_l}} G \setminus \{e_1, \dots, e_l\}$ . As all graphs in the expression of  $G'_I$  are trees, we obtain (by using  $X_e$  instead of  $x_{\alpha(e)} - x_{\omega(e)}$ ) :

$$\begin{aligned} N(G) &= \sum_{e_{\overline{0}} \in E_{\overline{0}}, e_{\overline{1}} \in E_{\overline{1}}} X_{e_{\overline{0}}} \cdot X_{e_{\overline{1}}} + \sum_{e_{\overline{0}} \in E_{\overline{0}}, e_2 \in E_2} X_{e_{\overline{0}}} \cdot X_{e_2} + \sum_{e_1 \in E_1, e_2 \in E_2} X_{e_1} \cdot X_{e_2} \\ &= \left( \sum_{e_{\overline{0}} \in E_{\overline{0}}} X_{e_{\overline{0}}} \right) \left( \sum_{e_{\overline{1}} \in E_{\overline{1}}} X_{e_{\overline{1}}} \right) + \left( \sum_{e_{\overline{0}} \in E_{\overline{0}}} X_{e_{\overline{0}}} \right) \left( \sum_{e_2 \in E_2} X_{e_2} \right) \\ &\quad + \left( \sum_{e_1 \in E_1} X_{e_1} \right) \left( \sum_{e_2 \in E_2} X_{e_2} \right). \end{aligned}$$

One can notice that, if  $E_{\overline{0}}$  is empty (that is to say that there is a chain from  $v$  to  $v'$ ), the polynomial  $N(G)$  is the product of two polynomials on degree 1. This is a particular case of our third main result (Theorem 7.1).

### 5.3 Simple bipartite graphs

**Definition 5.1** A graph is said to be bipartite if its set of vertices can be partitioned in two sets  $V_1$  and  $V_2$  such that  $E \subset V_1 \times V_2$ .

Moreover, a bipartite graph is said complete if  $E = V_1 \times V_2$ .

In this section we will look at bipartite graphs such that  $|V_1| = 2$ . Thanks to the pruning Lemma 3.1, we can suppose that it is a complete bipartite graph. The complete bipartite graph with  $|V_1| = 2$  and  $|V_2| = n$  is unique up to isomorphism and will be denoted  $G_{2,n}$ .

**Proposition 5.2** Let us call  $a_1, a_2$  (resp.  $b_1, \dots, b_n$ ) the variables associated to the vertices  $v_1^1$  and  $v_1^2$  of  $V_1$  (resp.  $(v_2^i)_{1 \leq i \leq n}$  of  $V_2$ ), one has :

$$N(G_{2,n}) = \sum_{i=1}^n \left( \prod_{j < i} (b_j - a_1) \cdot \prod_{k > i} (b_k - a_2) \right).$$

**Proof:** For each  $h = 1, 2$  and  $i = 1, \dots, n$ , we denote by  $e_{h,i}$  the edge  $(v_1^h, v_2^i)$ . We will show, by induction of  $n$ , that, by applying several times theorem 4.1, we obtain the following equality (which is drawn on figure 7 for  $n = 4$ ; we omit the  $\varphi$  for clearness) :

$$\begin{aligned} \varphi(G_{2,n}) &= \sum_{i=1}^n \varphi(G_{2,n} \setminus \{e_{2,1}, \dots, e_{2,i-1}, e_{1,i+1}, \dots, e_{1,n}\}) \\ &\quad - \sum_{i=1}^{n-1} \varphi(G_{2,n} \setminus \{e_{2,1}, \dots, e_{2,i}, e_{1,i+1}, \dots, e_{1,n}\}). \quad (9) \end{aligned}$$



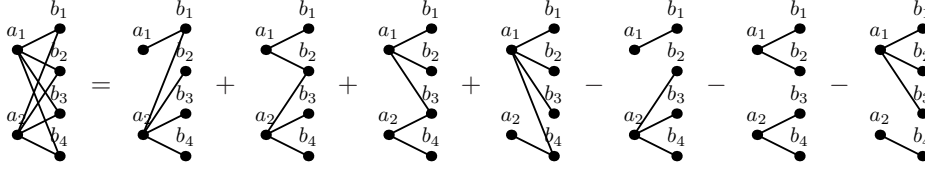


Fig. 7: Decomposition of  $\varphi(G_{2,4})$ .

For  $n = 1$ , the statement is obvious. Let us suppose that our formula is true for  $n$  and consider the graph  $G_{2,n+1}$ . As the equality at rank  $n$  is obtained by applying 4.1 to the graph  $G_{2,n}$ , we can apply it with the same loops in  $G_{2,n+1}$  (which contains canonically in  $G_{2,n}$ ). We obtain :

$$\varphi(G_{2,n+1}) = \sum_{i=1}^n \varphi(G_{2,n+1} \setminus \{e_{2,1}, \dots, e_{2,i-1}, e_{1,i+1}, \dots, e_{1,n}\}) - \sum_{i=1}^{n-1} \varphi(G_{2,n+1} \setminus \{e_{2,1}, \dots, e_{2,i}, e_{1,i+1}, \dots, e_{1,n}\}) \quad (10)$$

The graphs of the first line have still number loop 1. Applying theorem 4.1 with the cycle  $e_{2,i}, \overline{e_{1,i}}, e_{1,n+1}, \overline{e_{2,n+1}}$  give the following equality :

$$\begin{aligned} \varphi(G_{2,n+1} \setminus \{e_{2,1}, \dots, e_{2,i-1}, e_{1,i+1}, \dots, e_{1,n}\}) = \\ \varphi(G_{2,n+1} \setminus \{e_{2,1}, \dots, e_{2,i-1}, e_{2,i}, e_{1,i+1}, \dots, e_{1,n}\}) \\ + \varphi(G_{2,n+1} \setminus \{e_{2,1}, \dots, e_{2,i-1}, e_{1,i+1}, \dots, e_{1,n}, e_{1,n+1}\}) \\ - \varphi(G_{2,n+1} \setminus \{e_{2,1}, \dots, e_{2,i-1}, e_{2,i}, e_{1,i+1}, \dots, e_{1,n}, e_{1,n+1}\}). \end{aligned}$$

Using this formula for each  $i$ , the first summand balances with the negative term in (10) (except for  $i = n$ ) and the two other summands are exactly what we wanted. This ends the induction and Formula (10) is true for any  $n$ .

Note that the graphs of its right hand side have no loops and that only the ones of the first line are connected. We just have to apply  $\Psi$  to this equality, and use the value of  $\Psi$  on forests (Proposition 3.2) to finish the proof of the proposition.  $\square$

Note that this case is very interesting because the function  $N$  can be expressed as a Schubert polynomial (see (2, Proposition ?)).

**Remark 4** *Our algorithm allows us to write  $\varphi(G)$  as a sum of terms of the kind  $\pm\varphi(F)$ , with  $F$  subforests of  $G$ . But, in the three examples of this section, all trees have 0 or +1 as coefficients after iteration of transformations of kind  $T_L$  on  $G$ . We will see in the next section that this is possible for any graph  $G$  with a clever choices of loops.*

## 6 A combinatorial formula for N

To compute the polynomial  $N$  of a graph  $G$ , we only have to find the coefficient of trees in a formal linear combination of forests obtained by iterating transformations  $T_L$  on  $G$ . But there are many possible choices in these iterations and these coefficients depend on these choices.

A way to avoid this problem is to give to  $G$  a rooted map structure and to look at the particular decomposition introduced in the paper (6, section 3). With these particular choices, we have a combinatorial description of the trees with coefficient  $+1$ , all other trees having 0 as coefficient.

### 6.1 Rooted maps

**Definition 6.1** A (combinatorial oriented) map is a connected graph with, for each vertex  $v$ , a cyclic order on the edges whose origin or end is  $v$ . This definition is natural when the graph is drawn on a two dimensional surface (for example the planar posets of Section 7).

It is more convenient when we deal with maps, to consider edges as couples of two half-edges  $(h_1, h_2)$ , the first one of extremity  $\alpha(e)$  and the second one of extremity  $\omega(e)$ . Then the map structure is given by a permutation  $\sigma$  of all the half-edges, whose orbits correspond to the sets of half-edges with the same extremity.

A rooted map is a map with an external half-edge  $h_0$ , that is to say an half-edge which do not belong to any edge, but has an extremity (which will be denoted by  $\star$ ) and a place in the cyclic order given by this extremity.

**Remark 5** In this section, as cyclic orders of edges around vertices matter, we can not use the convention that the extremity of an edge is always on its origin's right (we did not assume any condition on compatibility between the orientations of the edges and the map structure, see open problem ??). So, when the orientation is important, we will use arrows.

Recall that, to compute  $N(G)$ , a naive algorithm is to choose any loop of the graph, apply proposition 4.4. If the graph has a rooted map structure, it is interesting to choose loops with additional properties. Our choices will not involve the orientation of the edges of the map. So we will define a notion of admissible loop in a (not necessary oriented) rooted map.

By definition, a loop  $L$  of a rooted map is admissible of type 1 (see figure 8) if:

- The vertex  $\star$  is a vertex of the loop, that is to say that  $\star$  is the extremity of a half-edge  $h_i$  of  $e_i$  and of a half-edge  $h_{i+1}$  of  $e_{i+1}$  for some  $i$  ;
- The cyclic order at  $\star$  restricted to the set  $\{h_0, h_i, h_{i+1}\}$  is the cyclic order  $(h_0, h_{i+1}, h_i)$ .

If  $L$  satisfies the first condition, exactly one loop among  $L$  and  $\bar{L}$  is admissible (where  $\bar{L}$  is  $L$  with the opposite orientation).

If a rooted map has no admissible loops of type 1, it is of the form of the figure 9. In this case, we call admissible loops of type 2 the admissible loops of its "legs"  $M_1, \dots, M_h$  (of type 1 or 2, this defines the admissible loops by induction). Note that this definition has a sense because the legs have a canonical

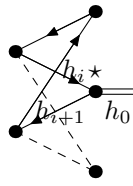


Fig. 8: Example of a map  $M$  with an admissible loop of type 1.

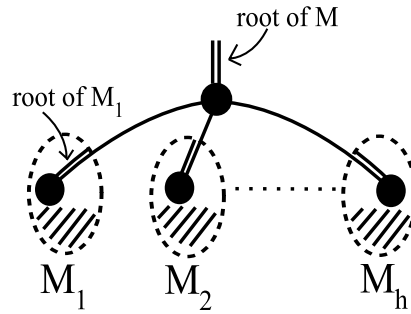


Fig. 9: A generic map  $M$  without admissible loops of type 1

external half-edge and are rooted maps. An example of an admissible loop of type 2 is drawn on Figure 10

A rooted map without admissible loops has no loops at all, hence it is a tree.

**Remark 6** *The second condition in the definition of admissible of type 1 says that the root must be at the left of the loop. The first condition is only technical, because if the loop does not go through  $\star$ , we can not define “to be on the left of the loop”.*

*For a planar map this can be avoided because any loop split the plan into two regions, so the left side of an oriented loop is well-defined. In this case, we can call admissible any loop such that the root is at the left of the loop even if the the loop does not go through  $\star$  and the confluence of the algorithm in the next paragraph will still be true.*

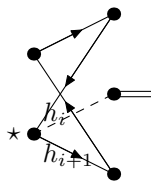


Fig. 10: Example of a map  $M'$  with an admissible loop of type 2.

## 6.2 Decomposition of rooted maps

Consider the following algorithm :

**Input** a rooted map  $M$ .

**Variable**  $S$  is a formal linear combination of submap of  $M$ .

**Initialization**  $S = M$ .

**Iterated step** Choose a map  $M_0$  with a non-zero coefficient  $c_{M_0}$  in  $S$  which is not a forest and  $L$  an admissible loop of  $M_0$ . Apply  $T_L$  to  $M_0$  in  $S$  and keep only the connected graphs in the right-hand side, (they have a natural induced rooted map structure). Formally,

$$S := S - c_{M_0} M_0 + c_{M_0} \delta(T_L(M_0)),$$

where  $\delta$  is the linear operator defined by :

$$\delta(M') = \begin{cases} M' & \text{if } M' \text{ connected} \\ 0 & \text{else} \end{cases}.$$

**End** We iterate this until  $S$  is a linear combination of subtrees of  $M$ .

**Output**  $S$ .

**Definition-Theorem 6.1** *This algorithm always terminates and is confluent. Let  $D(M)$  be its output.*

**Proof Idea of the proof:** The termination is obvious : all maps in  $T_L(M_0)$  have a lower cycle number than  $M_0$ .

For the confluence, the maps with a cycle number equal to 2 and such that different loops have edge in common play a similar role to critical peaks in rewriting theory. We just have to check our result on these maps. There are infinitely many maps of this kind, but, as in paragraph 5.2, one computation is enough to deal with the general case.  $\square$  For a complete proof, see (6, definition-theorem 3.1.1 and 3.2.1, together with remark 2).

**Proposition 6.2** *Let  $M$  be a rooted map.*

$$\Psi(D(M)) = \Psi(M)$$

**Proof:** We have to check that  $\Psi(S)$  is an invariant of our algorithm. This is trivial because operators  $T_L$  and  $\delta$  let  $\Psi$  invariant (see Theorem 4.1 and Proposition 4.5).  $\square$

**Example 3** *Let  $M$  be the complete bipartite graph  $G_{2,3}$  ( $V_1 = \{a_1, a_2\}$ ,  $V_2 = \{b_1, b_2, b_3\}$ ) with the following rooted map structure :*

- *If we denote by  $e_{1,i}$  (resp  $e_{2,i}$ ) the edge between  $a_1$  (resp.  $a_2$ ) and  $b_i$ , the cyclic order around the vertex  $a_1$  (resp.  $a_2$ ) is  $(e_{1,1}, e_{1,2}, e_{1,3})$  (resp.  $(e_{2,1}, e_{2,2}, e_{2,3})$ ).*

- The root has extremity  $b_2$  and is located before  $e_{2,2}$ .

The loop  $L = (e_{2,2}, e_{2,1}, e_{1,1}, e_{1,2})$  with  $E(L) = \{e_{2,1}, e_{1,2}\}$  (drawn on Figure 8) is admissible (of type 1). So, with this choice, after the first iteration of step 1 of our decomposition algorithm, we have :

$$S = \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_1 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_2 \quad b_2 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_3 \\ \diagdown \quad \diagup \\ \bullet \end{array} = \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_1 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_2 \quad b_2 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_3 \\ \diagdown \quad \diagup \\ \bullet \end{array} + \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_1 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_2 \quad b_2 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_3 \\ \diagdown \quad \diagup \\ \bullet \end{array} - \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_1 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_2 \quad b_2 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_3 \\ \diagdown \quad \diagup \\ \bullet \end{array}$$

The two firsts graph have each an admissible loop : the first one of type 1 ( $L = (\overline{e_{2,2}}, e_{2,3}, \overline{e_{1,3}}, e_{1,2})$  with  $E(L) = \{e_{2,3}, e_{1,2}\}$ ), the second one of type 2 ( $L = (e_{2,3}, \overline{e_{1,3}}, e_{1,1}, \overline{e_{2,1}})$  and  $E(L) = \{e_{2,3}, e_{1,1}\}$ , see figure 10). So the algorithm ends after two other iterations and we obtain :

$$D(M) = \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_1 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_2 \quad b_2 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_3 \\ \diagdown \quad \diagup \\ \bullet \end{array} + \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_1 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_2 \quad b_2 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_3 \\ \diagdown \quad \diagup \\ \bullet \end{array} + \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_1 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_2 \quad b_2 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_3 \\ \diagdown \quad \diagup \\ \bullet \end{array} + \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_1 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_2 \quad b_2 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_3 \\ \diagdown \quad \diagup \\ \bullet \end{array} - \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_1 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_2 \quad b_2 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_3 \\ \diagdown \quad \diagup \\ \bullet \end{array} \quad (11)$$

$$= \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_1 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_2 \quad b_2 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_3 \\ \diagdown \quad \diagup \\ \bullet \end{array} + \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_1 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_2 \quad b_2 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_3 \\ \diagdown \quad \diagup \\ \bullet \end{array} + \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_1 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_2 \quad b_2 \\ \diagdown \quad \diagup \\ \bullet \end{array} \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ a_1 \quad b_3 \\ \diagdown \quad \diagup \\ \bullet \end{array} \quad (12)$$

Note that, after cancellation, the coefficient of trees are 0 or +1. In the next paragraph we will show a general result (the sign is a particular case of (6, Proposition 3.3.1)) and characterize combinatorially the trees with a coefficient +1.

### 6.3 Coefficients in $D(M)$

To compute the polynomial  $N$ , we only have to compute the coefficient of spanning trees in  $D(M)$ . In this section, we will link this coefficient with a combinatorial property of the tree  $T$ .

**Definition 6.2** If  $T$  is a spanning subtree of a rooted map  $M$ , the tour of the tree  $T$  beginning at  $h_0$  defines an order on the half-edges which do not belong to  $T$ . The definition is easy to understand on a figure : for example, on Figure 11, the tour is  $(h_1^1, h_2^1, h_1^2, h_2^2, h_1^3, h_1^4, h_2^3, h_2^4)$ . (see (1) for a precise definition).

We recall that  $D(M)$  does not depend on the admissible loop chosen at step 1 of the decomposition algorithm. A good choice to compute the coefficient of a given spanning tree  $T \subsetneq M$  is given by lemma 6.3. Given an edge  $e$  of  $M \setminus T$ , it is well-known that there exists a unique loop (up to the orientation) denoted  $L^M(e)$  (or  $L(e)$  if there is no confusion) such that  $L(e) \subset (E_T \cup \{e\})$ .

**Lemma 6.3** There exists an edge  $e_0 \in M \setminus T$  such that, with the good orientation,  $L(e_0)$  is admissible. Moreover,

$$e_0 \in E(L(e_0)) \iff \begin{array}{l} \text{The first half-edge of } e_0 \text{ appears in} \\ \text{the tour of } T \text{ before the second one.} \end{array}$$

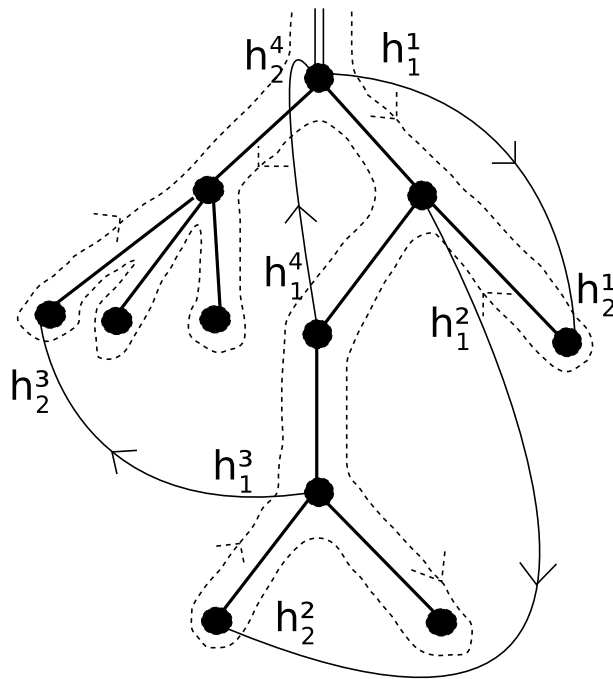


Fig. 11: Tour of a spanning tree of a map.

**Proof:** The proof of the lemma, by induction on the size of  $M$ , can be divided in three cases :

1. If there is an edge of  $M \setminus T$  whose origin or end is  $\star$  (the extremity of the external half-edge), then  $\star$  is a vertex of the loop  $L(e)$  and either  $L(e)$  or  $\overline{L(e)}$  is admissible of type 1.
2. Else, let  $T_1, \dots, T_l$  be the connected component of  $T \setminus \{\star\}$ . If there is an edge  $e$  whose extremities are in two different  $T_i$ , the  $L(e)$  is going through  $\star$  and  $e$  suits in the lemma.
3. Else,  $M \setminus \{\star\}$  has as many connected components as  $T \setminus \{\star\}$ . Let us denote them by  $M_i \supset T_i (1 \leq i \leq l)$ . There exists an  $j$ , such that  $M_j \supsetneq T_j$ . In this case  $M$  has no admissible loop of type 1, but by induction there exists  $e \in M_j \setminus T_j$  such that  $L^{M_j}(e)$  is admissible in  $M_i$ . By definition, this loop is admissible of type 2 in  $M$ . But  $L^{M_j}(e) = L^M(e)$ , so the proof of the lemma is over.

The second part of the proof is easy in the two first case (see figure 11). For the third one, it is again an immediate induction.  $\square$

This helps us to compute all coefficients of trees in  $D(M)$  :

**Proposition 6.4** *Let  $M$  be a rooted map and  $T$  a spanning tree of  $M$ .*

- *If there is an edge  $e = (h_1, h_2) \in M \setminus T$  such that  $h_2$  appears before  $h_1$  in the tour of  $T$ , then the coefficient of  $T$  in  $D(M)$  is 0.*
- *Else, the coefficient of  $T$  in  $D(M)$  is +1 ( $T$  will be said good).*

For example, the spanning tree of figure 11 is good. Note that the property of being a good spanning tree does not depend on the orientation of the edges of the tree, but only on the orientation of those which do not belong to it.

**Proof:** We will prove this proposition by induction over the number of edges in  $M \setminus T$ . If  $M = T$ ,  $T$  is good and the result is obvious.

Let  $T$  be a covering tree of rooted map  $M$  such that  $M \setminus T$  contains at least one element. We use lemma 6.3 and divide the proof in two cases :

**Case  $e_0 \notin E(L(e_0))$  :** In this case the spanning tree  $T$  can not be good. Moreover,  $E(L(e_0)) \subset T$ , so every map appearing in  $T_{L(e_0)}(M)$  does not contain  $T$ . But this remains true when we apply operators of kind  $T_L$ . No graph appearing in  $D(M)$  contains  $T$ . In particular, the coefficient of  $T$  is 0.

**Case  $e_0 \in E(L(e_0))$  :** In this case, one has :

$$T_{L(e_0)}(M) = M \setminus \{e_0\} + \text{maps which do not contain } T.$$

$$\text{So } D(M) = D(M \setminus \{e_0\}) + \sum_{M' \not\supset T} D(M').$$

As in the previous case, the second summand has a contribution 0 to the coefficient of  $T$  in  $D(M)$ . By induction hypothesis, the first one has contribution +1 if  $T$  is a good spanning tree of  $M \setminus \{e_0\}$  and 0 else. But, by definition of good spanning trees, it is immediate that :

$$T \text{ is a good spanning tree of } M. \iff \begin{array}{l} T \text{ is a good spanning tree of } M \setminus \{e_0\} \\ \text{and the first half-edge of } e_0 \text{ appears} \\ \text{before its second in the tour of } T. \end{array}$$

But as  $e_0 \in E(L(e_0))$ , the second condition of the right hand side is true by lemma 6.3. Finally, the coefficient of  $T$  is +1 if  $T$  is a good spanning subtree of  $M$  and 0 else.

□

#### 6.4 Combinatorial formula and an application

We are now ready to state our second main result : for this , we have to give a rooted map structure to our  $G$ . This is possible in multiple ways (choice of the map structure and of the place of the root).

**Theorem 6.5** *The polynomial  $N$  associated to the underlying graph  $G$  of a rooted map  $M$  is given by the following combinatorial formula :*

$$N(G) = \sum_{\substack{T \text{ good spanning} \\ \text{tree of } M}} \left[ \prod_{\substack{e \in E(G) \\ e \notin T}} (x_{\alpha(e)} - x_{\omega(e)}) \right]. \quad (13)$$

**Proof:**

$$\begin{aligned} N(G) &= \prod_{e \in E(G)} (x_{\alpha(e)} - x_{\omega(e)}) \cdot \Psi(G) \\ &= \prod_{e \in E(G)} (x_{\alpha(e)} - x_{\omega(e)}) \cdot \Psi \left( \sum_{\substack{T \text{ good spanning} \\ \text{tree of } M}} T \right) \\ &= \prod_{e \in E(G)} (x_{\alpha(e)} - x_{\omega(e)}) \cdot \left( \sum_{\substack{T \text{ good spanning} \\ \text{tree of } M}} \frac{1}{\prod_{e \in E(T)} (x_{\alpha(e)} - x_{\omega(e)})} \right) \\ &= \sum_{\substack{T \text{ good spanning} \\ \text{tree of } M}} \left[ \prod_{\substack{e \in E(G) \\ e \notin T}} (x_{\alpha(e)} - x_{\omega(e)}) \right]. \end{aligned}$$

□



Of course, the good spanning trees depend on the map structure chosen on the graph  $G$ . So the theorem implies that the right member does not depend on it, which is quite surprising.

Here is a little example of an application of Theorem 6.5 (see open problem 1.3 for a motivation of this corollary). The choice of a good map structure makes the proof quite easy.

**Corollary 6.6** *Let  $v$  be a vertex of a graph  $G$ . The function  $N(G)$  is a polynomial in  $x_v$  of degree lower than  $\text{val}(v)$ , where  $\text{val}(v)$  is the number of edges whose origin or end is  $v$ . Moreover the coefficient of  $x_v^{\text{val}(v)-1}$  is :*

$$\begin{cases} N(G \setminus \{v\}) & \text{if } v \text{ is a minimal element of } \text{poset}(G); \\ (-1)^{\text{val}(v)-1} N(G \setminus \{v\}) & \text{if } v \text{ is a maximal element of } \text{poset}(G); \\ 0 & \text{else.} \end{cases}$$

**Proof:** Eventually by considering  $G_{op}$  (obtained from  $G$  by changing the direction of its edges), we can suppose that  $v$  is the origin of an edge  $e$ . We will give to  $G$  an rooted map structure  $M$  which fulfills the following conditions (it is always possible, but not necessarily in a unique way):

- The root has extremity  $v$  and is located just after  $e$ .
- If  $v$  is the end of another edge, then the edge just before  $e$  (denoted by  $e'$ ) has  $v$  as end and not as origin.

An immediate consequence of the first condition is that  $e$  belongs to every good spanning tree of  $M$ . If  $v$  is not a minimum, the second condition must be fulfilled and  $e'$  is also in every good spanning tree. Each summand in (6.5) has a degree lower or equal to  $\text{val}(v) - 2$ , since, if  $T$  is a good spanning tree,  $M \setminus T$  has at most  $\text{val}(v) - 2$  edges of extremity  $v$ . The proof is finished in this case.

In the case where  $v$  is a minimum, recall that every good spanning tree  $T$  of  $M$  contains  $e$ . If  $T$  contributes in formula (6.5) to the term of degree  $\text{val}(v) - 1$  (in  $x_v$ ) of  $N(G)$ , then it does not contain any other edge of extremity  $v$ . So  $T \setminus \{v\}$  is a spanning tree of  $G \setminus \{v\}$ . But if  $T'$  is a spanning tree of  $G \setminus \{v\}$ , the definition of good spanning trees implies the following equivalence :

$$T \cup \{e\} \text{ good spanning tree of } M \iff T \text{ good spanning tree of } M \setminus \{v\},$$

where  $M \setminus \{e\}$  has the following rooted map structure : the map structure is induced by the one of  $M$  and the root is at the former place of the second half-edge of  $e$ . But, for such trees,

$$\begin{aligned} M \setminus (E_T \cup \{e\}) &= ((M \setminus \{v\}) \setminus E_T) \cup \{\text{val}(v) - 1 \text{ edges of origin } v\} \\ \prod_{e'' \in M \setminus (E_T \cup \{e\})} (x_{\alpha(e'')} - x_{\omega(e'')}) &= \prod_{e'' \in (M \setminus \{v\}) \setminus E_T} (x_{\alpha(e'')} - x_{\omega(e'')}) \cdot (x_v^{\text{val}(v)-1} + \dots). \end{aligned}$$

Using formula (6.5) for  $M$  and  $M \setminus \{v\}$ , the corollary is proved.  $\square$

## 7 A condition of factorization

In the previous section, we have given an additive formula for the numerator of the reduced fraction  $\Psi_P$ . Green formula for planar posets (see subsection 1.1) and the example of Figure 1 show that, in some cases, it can also be written under a factorized form. In this section we give a simple graphical condition on a graph  $G$ , which implies the factorization of  $N(G)$ .

### 7.1 Chain factorization

**Remark 7** In this section, we will assume that all the graphs are connected, have no circuits and no transitivity relation (an edge going from the beginning to the end of a chain). This is always the case for Hasse diagrams of posets so we do not lose in generality. With this assumption, if we consider a chain  $c$ , there is no edges between the vertices of the chain except of course the edges of the chain itself.

Let  $G$  be a graph,  $c$  a chain of  $G$ ,  $V_c$  the set of vertices of  $c$  (including the origin and the end of the chain) and  $G_1, \dots, G_k$  be all the connected component of  $G \setminus V_c$ . The complete subgraphs  $\overline{G}_i = G_i \cup V_c$  (for  $1 \leq i \leq k$ ) will be called region of  $G$ . Consider, for example, the graph of Figure 12 and the chain  $c = (1, 2, 13, 3, 4, 5, 6, 14)$ . The graph  $G \setminus V_c$  has four connected component drawn on Figure 13.

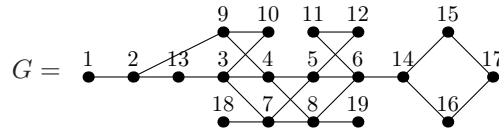


Fig. 12: Example of a graph  $G$  with a chain  $c = (1, 2, 13, 3, 4, 5, 6, 14)$ .

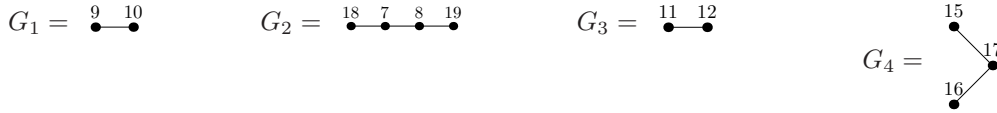


Fig. 13: The connected components  $G_1, G_2, G_3, G_4$  of  $G \setminus V_c$ .

Finally, the graph  $G$ , respectively to the chain  $c$ , has four regions presented on Figure 14.

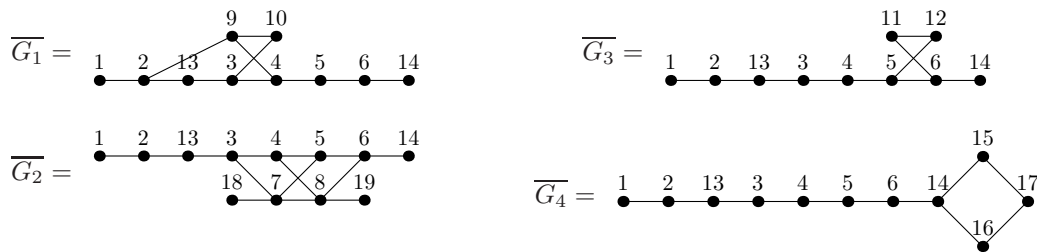


Fig. 14: Regions of  $G$ , with respect to the chain  $c$ .

We can now state our third main result:

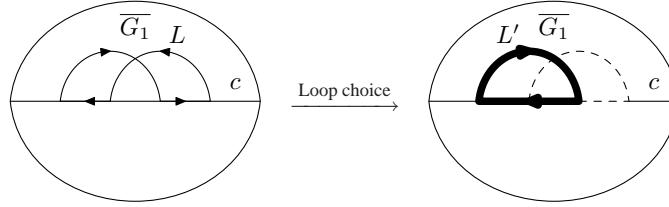
**Theorem 7.1** *Let  $G$  be a graph,  $c$  a chain of  $G$  and  $\overline{G_1}, \overline{G_2}, \dots, \overline{G_k}$  be the corresponding regions of  $G$ . Then one has :*

$$N(G) = \prod_{j=1}^k N(\overline{G_j}).$$

For example, the numerator of the rational function associated to the graph 12 can be factorized into four non-trivial factors.

*Proof.* The central idea is to apply Theorem 4.1 on loops  $L$  contained in a region and such that  $E(L) \cap c = \emptyset$ . This means that the edges of  $c$  can appear in  $L$ , but only in the *wrong* direction: so, when we apply Proposition 4.4, we do not *touch* to the chain  $c$ .

The first step is to prove the existence of such loops. This is done in Lemma 7.2 (see Figure 15 for an illustration).



**Fig. 15:** Good choice of loop

**Lemma 7.2** *Let  $G$  be a graph and  $c$  a chain of  $G$ . Denote by  $\overline{G_1}, \dots, \overline{G_k}$  the corresponding regions. If  $\overline{G_1}$  is not a tree, there exists a loop  $L$  in  $\overline{G_1}$  such that  $E(L) \cap c = \emptyset$ .*

**Proof:** Choose any loop  $L_0$  of  $\overline{G_1}$ . Two cases have to be examined:

- 1) The loop  $L_0$  has no vertices in common with  $c$ . Nothing has to be done.
- 2) The loop  $L_0 = (e_1, \dots, e_l)$  has at least one vertex in common with  $c$ . As a loop is not transformed if one makes a cyclic permutation of its edges, one can assume that  $c_1 = \text{ext}_1(e_1)$  is a vertex of  $c$ . Let us denote by  $h$  the smallest index such that  $c_2 = \text{ext}_2(e_h)$  is also a vertex of  $c$  (it necessarily exists because  $\text{ext}_2(e_l) = \text{ext}_1(e_1)$  is a vertex of  $c$ ). But there is a subchain (eventually empty)  $c'$  of  $c$  going from  $c_1$  to  $c_2$  (resp. from  $c_2$  to  $c_1$ ) if  $c_1 \leq c_2$  (resp. if  $c_2 \leq c_1$ ). Now, we just have to define  $L$  as:

$$L = \begin{cases} (e_1, \dots, e_l) \cdot \overline{c'} & \text{if } c_1 \leq c_2 \\ (e_l, \dots, e_1) \cdot \overline{c'} & \text{if } c_2 \leq c_1 \end{cases},$$

where  $\overline{c'}$  denotes the chain  $c'$  in the other direction (this implies that all the edges of  $c'$  are in the *wrong* direction in  $L$ , so  $E(L) \cup c = \emptyset$ ).

For example, the loop  $L = ((2, 4), (4, 10), (10, 9), (9, 12), (12, 2))$  from the figure 16 have an edge in the region  $P_1 = \{2, 4, 5, 8, 9, 10\}$ . Using the previous algorithm, we can chose a new loop

$$L' = ((2, 5), (5, 8), (8, 9), (9, 10), (10, 4), (4, 2))$$

verifying all the properties of the Lemma 7.2.

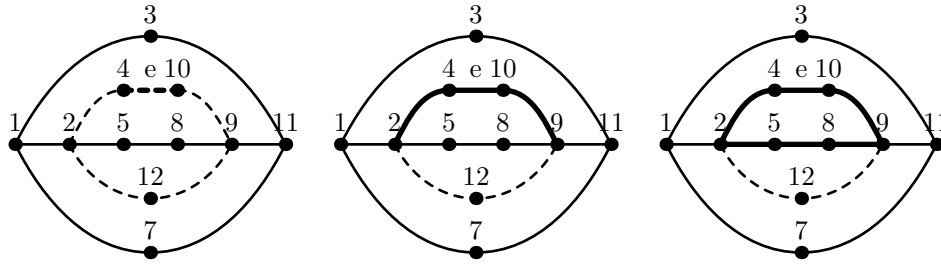


Fig. 16: Loop choice when a cycle have edges in common with the chain.

□

We make a proof by induction on  $k$ : if  $k = 1$ , then the result is trivial.

Suppose now that our proposition is true for  $k = n - 1$ . Let  $G$  be a graph and  $c$  a chain of  $G$ , such that there are  $n$  associated regions  $\overline{G}_1, \dots, \overline{G}_n$ .

If  $\overline{G}_1$  is a tree, one can prune it (remove successively vertices of arity 1 and the edge having it as extremity) to obtain the chain  $c$ . We can remove the same vertices and edges from the whole graph  $G$  because the removed vertices are not linked with an other  $G_i$  (as the  $G_i$  are different connected components of  $G \setminus V_c$ ). Thanks to the pruning-invariance Lemma 3.1, one has :

$$N(G) = N\left(\bigcup_{i=2}^n \overline{G}_i\right) = \prod_{i=2}^k N(\overline{G}_i),$$

where the second equality is due to the induction hypothesis. The theorem is proved in this case.

Let us come back to the general case. If  $\overline{G}_1$  is not a tree, we will use Proposition 4.4 with loops in  $\overline{G}_1$  and apply the previous case:

Lemma 7.2 gives us a loop  $L_1$  of  $\overline{G}_1$  such that  $E(L_1) \cap c = \emptyset$ . Applying Proposition 4.4 on  $L$ , one has

$$N(G) = \sum_{\substack{E_1 \subset E(L_1) \\ E_1 \neq \emptyset}} \pm N(G \setminus E_1) \left( \prod_{e \in E_1} x_e \right).$$

For the set(s)  $E_1$  such that  $\text{bar}G_1 \setminus E_1$  has a cycle, we can iterate the previous operation by choosing a new loop in  $\overline{G_1} \setminus E_1$ . We obtain recursively that

$$N(G) = \sum_{\substack{E_1 \subset E(L_1) \\ E_1 \neq \emptyset}} \left( \dots \left( \sum_{\substack{E_p \subset E(L_p) \\ E_p \neq \emptyset}} \pm N(G \setminus (E_1 \cup \dots \cup E_p)) \prod_{e \in E_1 \cup \dots \cup E_p} (x_{\alpha(e)} - x_{\omega(e)}) \right) \right),$$

where  $L_i$  is a loop of  $\overline{G_1} \setminus \{E_1 \cup \dots \cup E_{i-1}\}$  and the graphs  $\overline{G_1} \setminus (E_1 \cup \dots \cup E_p)$  have no cycle. Of course,  $L_i$  depends on the  $E_j, j < i$  and even the integer  $p$  depends on the  $E_j$ .

Some of the graphs  $G \setminus (E_1 \cup \dots \cup E_p)$  are disconnected (if and only if  $\overline{G_1} \setminus (E_1 \cup \dots \cup E_p)$ ). The value of  $N$  on these graphs is 0. So they do not appear in the formulas 14 and 15.

Each connected graph  $G \setminus (E_1 \cup \dots \cup E_p)$  contains the chain  $c$  (thanks to the assumption  $E(L_i) \cap c = \emptyset$ ). The associated regions are  $\overline{G_2}, \dots, \overline{G_n}$  and some regions which union is  $\overline{G_1} \setminus (E_1 \cup \dots \cup E_p)$ . We can do as if the last regions were only one region  $\overline{G_1} \setminus (E_1 \cup \dots \cup E_p)$  and use the case where  $\overline{G_1}$  is a tree:

$$N(G \setminus (E_1 \cup \dots \cup E_p)) = N(\overline{G_2}) \cdot \dots \cdot N(\overline{G_n}).$$

Finally,

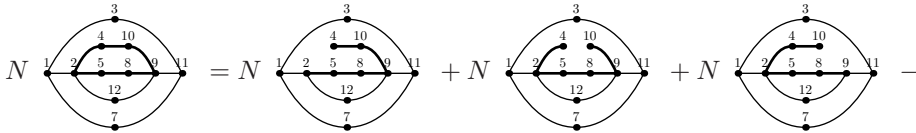
$$N(G) = \left( \sum_{\substack{E_1 \subset E(L_1) \\ E_1 \neq \emptyset}} \dots \sum_{\substack{E_p \subset E(L_p) \\ E_p \neq \emptyset}} \pm \prod_{e \in E_1 \cup \dots \cup E_p} (x_{\alpha(e)} - x_{\omega(e)}) \right) \cdot N(\overline{G_2}) \cdot \dots \cdot N(\overline{G_n}), \quad (14)$$

where the sum is restricted to the sequences  $E_i, 1 \leq i \leq p$  such that  $G \setminus (E_1 \cup \dots \cup E_p)$  is connected. But we can compute  $N(\overline{G_1})$  by choosing exactly the same loops and iterating proposition 4.4 :

$$N(\overline{G_1}) = \sum_{\substack{E_1 \subset E(L_1) \\ E_1 \neq \emptyset}} \dots \sum_{\substack{E_p \subset E(L_p) \\ E_p \neq \emptyset}} \pm \prod_{e \in E_1 \cup \dots \cup E_p} (x_{\alpha(e)} - x_{\omega(e)}), \quad (15)$$

where the sum is restricted to the sequences  $E_i, 1 \leq i \leq p$  such that  $\overline{G_1} \setminus (E_1 \cup \dots \cup E_p)$  is connected, or equivalently such that  $G \setminus (E_1 \cup \dots \cup E_p)$  is connected.

This ends the proof of Theorem 7.1.



## 7.2 Complete factorization of planar posets

In his paper (8), C. Greene has given a closed expression for the sum  $\Psi(G)$  when  $G$  is the minimal graph (Hasse diagram) of a *planar* poset. In this case,  $N(G)$  is a product of terms of degree 1 (Theorem 7.3). We will see that this factorization property is a consequence of Theorem 7.1 and give a new proof of Greene's Theorem.

Let us begin by defining precisely planar posets:

**Definition 7.1** We will say that the drawing of an oriented graph (without circuit) is *ordered-embedded* in  $\mathbb{R} \times \mathbb{R}$  if

- the origin of an edge is always at the left of its end.
- the edges are straight lines.

A graph  $G$  is said *planar* if it can be ordered embedded in  $\mathbb{R} \times \mathbb{R}$  without edge-crossings. If  $G$  is a graph, we denote by  $G_{0,\infty}$  the graph obtained from  $G$  by adding :

- A vertex  $0$  (called *minimal vertex*) and, for each vertex  $v$  of  $G$  which is not the end of any edge of  $G$ , an edge going from  $0$  to  $v$ .
- A vertex  $\infty$  (called *maximal vertex*) and, for each vertex  $v$  of  $G$  which is not the origin of any edge of  $G$ , an edge going from  $v$  to  $\infty$ .

A graph  $G$  is said *super-planar* if the graph  $G_{0,\infty}$  is planar.

A poset  $\mathcal{P}$  is planar if its minimal graph  $G$  is super-planar.

Almost all drawings of this paper (except in section 6 are ordered-embedded in  $\mathbb{R} \times \mathbb{R}$ . See Figure 17 and 18 for examples of super-planar and non super-planar graphs.

Note that the complete subgraph on a subset of vertices of a super-planar graph is super-planar (note that, however, if we erase some edges, we can obtain a non super-planar graph). In particular, the regions of a super-planar graph with respect to a chain are the graph of super-planar graphs.

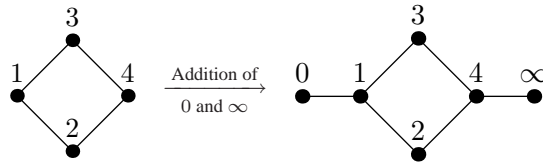


Fig. 17: The graph  $G$  is super-planar.

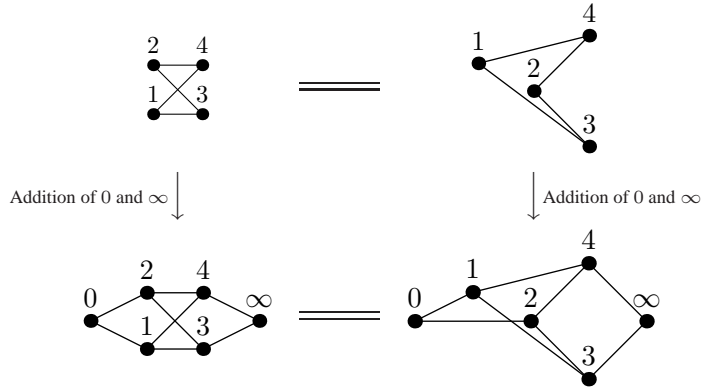


Fig. 18: The graph  $G$  is planar, but not super-planar.

Moreover, a graph with one cycle and without vertices with arity 1 is super-planar if and only if it has a unique maximal and a unique minimal element. In this case, we will call it a diamond (an example is drawn on Figure 19).

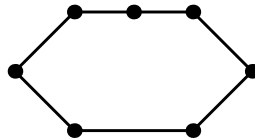


Fig. 19: A diamond

These definitions are relevant because there is a closed formula for  $\Psi_{\mathcal{P}}$  for planar posets:

**Theorem 7.3 (Greene (8))** *Let  $P$  be a planar poset, then*

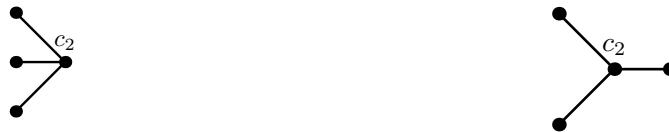
$$\Psi_{\mathcal{P}} = \begin{cases} 0 & \text{if } P \text{ is not connected,} \\ \prod_{y,z \in P} (x_y - x_z)^{\mu_P(y,z)} & \text{if } P \text{ is connected.} \end{cases},$$

where  $\mu(x, y)$  denotes the Möbius function of the poset  $P$ .

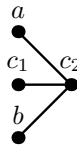
We will show that we can find disconnecting chains in any super-planar graphs, explaining the fact that the function  $N(G)$  can be factorized into factors of degree 1.

**Proposition 7.4** *Let  $G$  be a super-planar oriented graph with a number of cycle greater than 1, then there is a chain of  $G$ , separating  $G$  in two non-trivial regions (with at least one cycle).*

**Proof:** Eventually by pruning it, one can assume that  $G$  has no vertices with arity 1. As it has at least two cycles, it has one vertex  $c_2$  of arity 3 or more. So, up to a left-right symmetry, we are in one of the two following cases (in the second case, we assume that  $c_2$  is the end of *exactly* 2 edges).



In the first case, let us label the vertices as below:



In the second case, we define by induction  $c_i$  for  $i \geq 3$ : we choose for  $c_i$  any vertex such that there is an edge of origin  $c_{i-1}$  and of end  $c_i$ . For a  $k \geq 3$ , one can not define  $c_{k+1}$  if  $c_k$  is not the origin of any edge. Then, as  $c_{k+1}$  is not a free vertex, it is the end of an edge coming from a vertex  $b \neq c_{k-1}$ . Finally, we call  $c_1$  and  $a$  the origins of the two edges whose ends are  $c_2$ : which one is  $c_1$  and which one is  $a$  depends of whether  $b$  is above or below  $c_{k-1}$  (see the figure below).



In every case, the  $c_i$  are the vertices of a chain  $c$  of  $G$ , which can be extended to a maximal chain  $c_{\max}$ . Recall that with Greene's definition of a planar graph, the graph  $G_{0,\infty}$ , i.e. can still be ordered-embedded in the plan. Then there is a chain in  $G_{0,\infty}$  containing  $v_0$ ,  $c_{\max}$  and  $v_\infty$ . It splits  $G_{0,\infty}$  into at least two regions, one containing  $a$  and one containing  $b$ . The same is true for the chain  $c_{\max}$  in  $G$ . But, as  $G$  has no vertices, the corresponding regions have at least one cycle. □

**Corollary 7.5** *Let  $G$  be a connected super-planar poset. By iterating chain factorization, one can write  $N(G)$  as a product of numerators of rational functions associated to diamonds.*



**Proof:** Proposition 7.4 and Theorem 7.1 imply that  $N(G)$  can be factorized as the product of numerators of subgraphs with one loop. As these subgraphs are super-planar, after pruning, they are diamonds, which ends the proof.  $\square$

Note that for a diamond

$$N(D) = x_{\min(D)} - x_{\max(D)},$$

or equivalently,

$$\Psi(D) = \prod_{y,z \in P} (x_y - x_z)^{\mu_D(y,z)},$$

where  $\mu_D$  is the Möbius function of the poset associated to the diamond  $D$ .

The last property can be extended to any planar poset thanks to the following compatibility between disconnecting chain and Möbius function :

**Proposition 7.6** *Let  $P$  be a poset,  $c$  a chain of the Hasse diagram of  $P$  (i.e. the minimal graph representing  $P$ ),  $P_1, \dots, P_n$  the  $n$  region associated with  $c$ , and  $i, j$  two different elements of  $P$ , then*

$$\mu_{i,j}(P) = \begin{cases} -1 & \text{if } i \preceq j, \\ \sum_{k=1}^n \mu_{i,j}(P_k) & \text{otherwise.} \end{cases}$$

We assume that  $\mu_{i,j}(Q) = 0$  if  $i \notin Q$  or  $j \notin Q$ .

The proof is postponed to paragraph 7.3.

This proposition together with corollary 7.5 proves Green theorem. In fact, it can be generalised. Indeed, the poset of the figure 20 is not planar but can be factorised and the numerator can be expressed with the Möbius function : this is the case of any gluing of diamonds along chains.

$$\begin{aligned} N \left( \begin{array}{c} \text{Diagram 1} \end{array} \right) &= N \left( \begin{array}{c} \text{Diagram 2} \end{array} \right) \cdot N \left( \begin{array}{c} \text{Diagram 3} \end{array} \right) \\ &\quad \cdot N \left( \begin{array}{c} \text{Diagram 4} \end{array} \right) \\ &= (x_1 - x_4) \cdot (x_2 - x_5) \cdot (x_3 - x_6) \end{aligned}$$

**Fig. 20:** Chain factorisation ( $c = (2, 3, 4, 5)$ ).

### 7.3 Chain and Möbius function

This paragraph is the proof of the technical Proposition 7.6

**Proof:**

When  $i \leq j$  (there is an edge from  $i$  to  $j$  in the Hasse diagram of the poset), one always has  $\mu_{i,j}(P) = -1$ .

When  $i \leq j$ , but  $i \not\leq j$ , four cases have to be examined:

**first case**  $i, j$  do not belong to  $V_c$  and in different regions of the poset ;

**second case**  $i, j$  do not belong to  $V_c$ , but are in the same region of the poset ;

**third case**  $i$  is an element of  $V_c$ , but  $j$  is not ;

**fourth case**  $i$  and  $j$  are two elements of  $V_c$ .

Figure 21, 22, 23 and 24 summarize the four cases. Note that the case where  $i$  does not belong to  $V_c$ , but  $j$  does, can be obtained from the third one by considering the opposite poset.

Let  $P_1, \dots, P_n$  be the  $n$  regions associated with  $P$ .

We denote by  $[a, b]_P$  the set

$$[a, b]_P = \{k \mid a \leq_P k \leq_P b\},$$

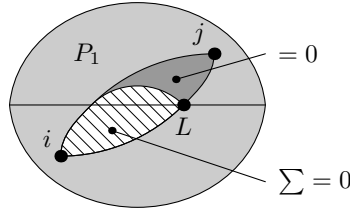
and by  $[a, b[_P$  the set

$$[a, b[_P = \{k \mid a \leq_P k <_P b\}.$$

Note that  $[i, j]_{P_1} = [i, j]_P \cap P_1$ . This property is not true for any poset associated to a complete sub-graph of  $G$ , the fact that  $P_1$  is a region defined by a disconnecting chain is here very important.

If  $[i, j]_P$  has a non-empty intersection with  $V_c$ , we denote by  $L$  the maximal element of this intersection.

- 1) Suppose that  $i \in P_2 \setminus V_c$  and  $j \in P_1 \setminus V_c$ . We want to prove that  $\mu_P(i, j) = 0$  and we assume (proof by induction) that it is true for any  $j' \in P_1 \setminus V_c$  such that  $j' < j$ .



**Fig. 21:** Case 1:  $i \notin c$  and  $j$  is not in the same region than  $j$ .

As  $i \leq j$ , there is a chain in the Hasse diagram of  $P$  going from  $i$  to  $j$ . As  $c$  is a chain separating  $P_1$  and  $P_2$ , any chain from  $i$  to  $j$  intersect  $c$ . Thus  $L$  exists and any element between  $i$  and  $j$  which is not in  $P_1$ , is lower or equal to  $L$ . So

$$[i, j]_P \cap (P_2 \cup \dots \cup P_m) \subseteq [i, L]_P \subseteq [i, j]_P.$$

By definition of the Möbius function we obtain,

$$\mu_{i,j}(P) = - \sum_{k \in [i,L]_P} \mu_{i,k}(P) - \sum_{k \in [i,j]_{[P \cap P_1] \setminus [i,L]_P}} \mu_{i,j}(P)$$

As

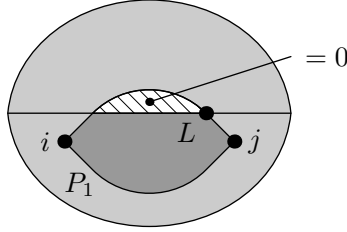
$$\sum_{k \in [i,L]_P} \mu_{i,k}(P) = 0$$

one has :

$$\mu_{i,j}(P) = - \sum_{k \in [i,j]_{[P \cap P_1] \setminus [i,L]_P}} \mu_{i,j}(P) \quad (16)$$

By induction hypothesis,  $\mu_{i,j}(P) = 0$ .

2) Suppose now that  $i, j \in P_1 \setminus V_c$ . We want to prove that  $\mu_{i,j}(P) = \mu_{i,j}(P_1)$



**Fig. 22:** Case 2:  $i \notin c$  and  $j$  is in the same region than  $i$ .

By definition of the Möbius function, we have,

$$\mu_{i,j}(P) = - \sum_{k \in [i,j]_{[P \cap (P_2 \cup \dots \cup P_m \setminus V_c)]}} \mu_{i,j}(P) - \sum_{k \in [i,j]_{P_1}} \mu_{i,j}(P).$$

The case 1 gives:  $\sum_{k \in [i,j]_{[P \cap (P_2 \cup \dots \cup P_m \setminus V_c)]}} \mu_{i,j}(P) = 0$ . Therefore,

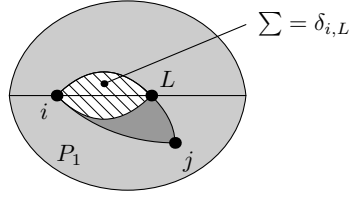
$$\mu_{i,j}(P) = \sum_{k \in [i,j]_{P_1}} \mu_{i,j}(P). \quad (17)$$

and an immediate induction proves that  $\mu_{i,j}(P) = \mu_{i,j}(P_1)$ .

3) Suppose that  $i \in c$  and  $j \in P_1 \setminus V_c$ . As  $i \in V_c \cap [i, j]_P$ , the set is not empty and  $L$  exists.

We will prove now that  $\mu_{i,j}(P) = \mu_{i,j}(P_1)$  by induction on  $j$ . As

$$\sum_{k \in [i,L]_P} \mu_{i,k}(P) = \delta_{i,L}$$



**Fig. 23:** Case 3:  $i \in c$  and  $j \notin c$ .

one has:

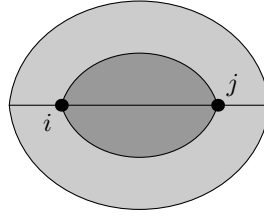
$$\mu_{i,j}(P) = - \sum_{k \in [i,j]_P \setminus [i,L]_P} \mu_{i,k}(P) - \delta_{i,L}$$

Similarly,

$$\mu_{i,j}(P_1) = - \sum_{k \in [i,j]_{P_1} \setminus [i,L]_{P_1}} \mu_{i,k}(P_1) - \delta_{i,L}$$

But  $[i,j]_P \setminus [i,L]_P = [i,j]_{P_1} \setminus [i,L]_{P_1}$  (see the proof of case 1), so an immediate induction on  $j$  finishes the proof in this case.

4) Suppose that  $i \in c$  and  $j \in c$ . We want to prove by induction on  $j$  ( $i \not\preceq j$ ) that  $\mu_{i,j}(P) = \sum_{l=1}^n \mu_{i,j}(P_l)$ .



**Fig. 24:** Case 4:  $i$  and  $j$  belong to  $V_c$ .

By definition of the Möbius function, we have

$$\mu_{i,j}(P) = - \sum_{k \in [i,j]_P} \mu_{i,k}(P)$$

Using case 3 of this proof and the induction hypothesis, we know that  $\mu_{i,k}(P) = \sum_{l=1}^n \mu_{i,k}(P_l)$  if  $k \in [i,j]_P$  except for:

**k = i** In this case,  $\mu_{i,i}(P) = \mu_{i,i}(P_l) = 1$ , thus  $\mu_{i,i}(P) = \sum_{l=1}^n \mu_{i,i}(P_l) - (n-1)$ .

**k = i<sub>1</sub>** where  $i_1$  is defined by  $i_1 \in c$  and  $i \preceq i_1$ . In this case, one has  $\mu_{i,i_1}(P) = \mu_{i,i_1}(P_l) = -1$ , thus  $\mu_{i,i_1}(P) = \sum_{l=1}^n \mu_{i,i_1}(P_l) + (n-1)$ .

Finally, one has:

$$\mu_{i,j}(P) = \sum_{l=1}^n \left( - \sum_{k \in [i,j]_P \setminus c} \mu_{i,k}(P_l) \right) - (n-1) + (n-1).$$

Using the definition of the Möbius function for the  $P_l$ , this ends the proof of the proposition.

□

## References

- [1] O. Bernardi, *A characterization of the Tutte polynomial via combinatorial embedding*, Annals of Combinatorics, Vol **12**(2), pp 139-153 (2008).
- [2] A. Boussicault, *Operations on Posets and Rational Identities of Type A*. Poster session of 19th Formal Power Series and Algebraic Combinatorics (2007).
- [3] M. Bousquet-Mélou and S. Butler, *Forest-like permutations*, Annals of Combinatorics **11** (2007) 335-354.
- [4] M. Demazure, *Une nouvelle formule des caractères*, Bull. Sci. Math., **98** (1974) 163-172.
- [5] R. Diestel, *Graph Theory*, Graduate Texts in Mathematics **173** (2005), Springer-Verlag Heidelberg.
- [6] V. Féray, *Combinatorial interpretation and positivity of Kerov's character polynomials*, Journal of Algebraic Combinatorics, (2008) to appear.
- [7] V. Féray and P. Śniady. *Asymptotics of characters of symmetric groups related to Stanley-Féray character formula*, arXiv:math/0701051 (2007).
- [8] C. Greene, *A rational function identity related to the Murnaghan-Nakayama formula for the characters of  $S_n$* , J. Alg. Comb. **1** 3 (1992), 235-255.
- [9] A. Lascoux, *Schubert and MacDonalld polynomials, a parallel*, preprint (2008), available online: [www-igm.univ-mlv.fr/~al/ARTICLES/MsriExp.pdf](http://www-igm.univ-mlv.fr/~al/ARTICLES/MsriExp.pdf)
- [10] W. T. Tutte, *A census of planar map*, Canad. J. Math. **14**, (1963) 21-38.