

Décompositions des mots tassés et
auto-dualité de l'algèbre des fonctions
quasi-symétriques en mots

*Decompositions of packed words and self duality of Word
Quasi-Symmetric Functions*

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580,
sciences et technologies de l'information et de la communication (STIC)

Spécialité de doctorat : Mathématiques et Informatique
Graduate School : Informatique et sciences du numérique
Référent : Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche Laboratoire interdisciplinaire des
sciences du numérique (Université Paris-Saclay, CNRS), sous la direction de
Florent HIVERT, Professeur et la co-direction de **Viviane PONS**, Maîtresse de
conférences

Thèse soutenue à Paris-Saclay, le 8 Décembre 2022, par

Hugo MLODECKI

Composition du jury

Membres du jury avec voix délibérative

Sylvie Boldo

Directrice de recherche, Université Paris-Saclay

Présidente & Examinatrice

Marcelo Aguiar

Professor, Cornell University

Rapporteur & Examinateur

Loïc Foissy

Professeur, Université du Littoral Côte d'Opale

Rapporteur & Examinateur

Jean-Christophe Aval

Chargé de recherche, Université de Bordeaux

Examinateur

Nantel Bergeron

Professor, York University

Examinateur

Bérénice Delcroix-Oger

Maîtresse de conférences, Université de Montpellier

Examinatrice

Titre : Décompositions des mots tassés et auto-dualité de l’algèbre des fonctions quasi-symétriques en mots

Mots clés : Algèbres de Hopf bidendrifomes, mots tassés, éléments primitifs, éléments totalement primitifs, Ordres partiels

Résumé : Ce travail est fondé sur la théorie des bigèbres bidendrifomes, développée par Foissy, qui sont des algèbres de Hopf particulières où le produit et le coproduit peuvent être scindés en deux. Son théorème principal est : Une bigèbre bidendrifome est générée librement par “l'espace des éléments totalement primitifs” en tant qu’algèbre dendriforme. Une conséquence est l’auto-dualité des bigèbres bidendrifomes.

Parmi les nombreuses algèbres de Hopf, Hivert a défini l’algèbre des fonctions quasi-symétriques en mots, notée WQSym. En prouvant que WQSym est une bigèbre bidendrifome, Novelli-Thibon résolvent la conjecture de Duchamp-Hivert-Thibon sur l’auto-dualité de WQSym. Cependant, comme aucune construction générale de l’ensemble des totalement primitifs n’est formulée, nous n’avons pas d’automorphisme explicite pour le passage de la primaire à la duale.

La question centrale de cette thèse est donc de construire un isomorphisme bidendrifome entre WQSym et sa duale. Cette construction passe par la décomposition des mots tassés à l'aide de deux nouvelles opérations que nous avons définies. En outre, pour illustrer ces décompositions, nous avons créé une nouvelle famille d'objets combinatoires : les forêts d'arbres biplans. Certains sous-ensembles de mots tassés ne peuvent être décomposés par ces opérations. Nous avons prouvé que leurs séries génératrices sont égales aux dimensions de l'espace des élé-

ments totalement primitifs. L’intérêt des forêts biplanes est de faire apparaître visuellement les sous-ensembles de mots tassés indécomposables.

Ces forêts biplanes sont donc la forme idéale pour indexer des nouvelles bases, que nous avons créées, de l’algèbre WQSym et sa duale. En effet, il est aisément d’en extraire un sous-ensemble qui définit deux bases des espaces totalement primitifs de WQSym et sa duale. Enfin, des arbres biplans bicolores permettent d’obtenir un isomorphisme bidendrifome par un simple échange de couleurs, ce qui répond à notre question initiale et constitue le résultat principal de cette thèse.

Après l’obtention de ce résultat, nous nous intéressons aux relations entre les opérations évoquées. Nous remarquons alors fortuitement que ces opérations vérifient des relations semblables à des opérades bien connues (dupliciale déformée, L-algèbre, bigraft) mais qui sont a priori sans lien avec l’opérade dendriforme. Nous prouvons que l’ensemble des mots tassés munis de ces opérations décrit une algèbre sur ces opérades et en donnons des sous-ensembles de générateurs.

Nous terminons cette thèse par une partie expliquant notre démarche expérimentale de recherche utilisant SageMath. Nous décrivons les tutoriels que nous avons conçus sous la forme de notebooks et mis en ligne à disposition des autres chercheurs. Nous y présentons le code qui permet de vérifier tous nos résultats sur des exemples calculés par des algorithmes.

Title : Decompositions of packed words and self duality of Word Quasi-Symmetric Functions

Keywords : Bidendriform Hopf algebras, packed words, primitive elements, totally primitive elements, partial orders

Abstract : This work is founded on the theory of bidendriform bialgebras, developed by Foissy, which are particular Hopf algebras where the product and the coproduct can be split into two parts. His main theorem is : A bidendriform bialgebra is freely generated by “the space of totally primitive elements” as a dendriform algebra. A consequence of this is the self-duality of bidendriform bialgebras.

Among the many Hopf algebras, Hivert defined the algebra of word quasi-symmetric functions, denoted WQSym . By proving that WQSym is a bidendriform bialgebra, Novelli-Thibon solved the Duchamp-Hivert-Thibon conjecture on the self-duality of WQSym . However, since no general construction of the set of totally primitive was formulated, we do not have an explicit isomorphism between the primal and the dual.

The central question of this thesis is the construction of a bidendriform isomorphism between WQSym and its dual. This construction goes through a decomposition of packed words using two new operations that we have defined. Furthermore, to illustrate these decompositions, we have created a new family of combinatorial objects : forests of biplane trees. Some subsets of packed words cannot be decomposed by these operations. We proved that their generating series are equal to the dimensions of the space of

the totally primitive elements. The interest of biplane forests is to visually reveal the subsets of indecomposable packed words.

These biplane forests are therefore the ideal form for indexing the new bases, that we have created, of the algebra WQSym and its dual. In fact, it is easy to extract from them a subset which defines two bases of totally primitives spaces of WQSym and its dual. Finally, bicolored biplane trees allow us to obtain a bidendriform isomorphism by a simple exchange of colors, which answers our initial question and constitutes the main result of this thesis.

After obtaining this result, we study the relationships between the aforementioned operations. We then remark fortuitously that these operations verify relations similar to well-known operads (skew-duplicial, L-algebra, bigraft) but which are unrelated to the dendriform operad. We prove that the set of packed words endowed with these operations describes an algebra over these operads and give subsets of generators.

We end this thesis with a part explaining our experimental approach of research using SageMath. We describe the tutorials that we have designed in the form of notebooks and made available online for other researchers. We present the code that allows to check all our results on examples calculated by algorithms.

Remerciements

Presque 5 ans après le début de mon stage de M2, beaucoup d'émotions se mélangent. C'est seulement au dernier moment, quelques jours avant ma soutenance, que je rédige ces remerciements. Merci d'avance à tous ceux que j'ai oubliés et qui ont pourtant été très importants pour moi !

Je tiens à commencer par remercier Florent Hivert et Viviane Pons qui ont été mes directeur et directrice de thèse après m'avoir encadré lors de mon stage de M2. Avant qu'ils me fassent découvrir la combinatoire algébrique, je n'imaginais pas qu'il était possible de mêler de l'informatique à de l'algèbre à ce point. J'ai donc pu renouer avec les mathématiques qui m'avaient plus manqué que je ne l'aurais cru lors de mes études en informatique. Florent et Viviane sont très complémentaires et ont fait bien plus que m'aider durant toutes ces années. Surtout pour les difficultés liées au travail de recherche et d'enseignement évidemment, mais aussi par leur soutien puissant pour tout ce qui peut arriver autour. J'ai particulièrement apprécié leur engouement quand les choses se passaient bien, pas de meilleur moyen de me motiver à mieux faire. Je les remercie également de m'avoir poussé hors de ma zone de confort pour m'améliorer en tant que chercheur quand cela leur semblait important.

Je poursuis en remerciant Marcelo Aguiar et Loïc Foissy qui ont accepté d'être mes rapporteurs. Grâce à eux, des imprécisions ont pu être évitées dans la version finale du manuscrit. La majorité de mon travail étant fondé sur des résultats de Loïc Foissy, j'ai été honoré de l'attention qu'il a bien voulu accorder à mes travaux. Je souhaite remercier également tous les autres membres de mon jury : Jean-Christophe Aval, Nantel Bergeron, Sylvie Boldo et Bérénice Delcroix-Oger pour avoir accepté de faire partie de mon jury. Par ailleurs, je remercie de nouveau Nantel Bergeron et Jean-Christophe Aval, pour m'avoir accueilli respectivement au Canada au début de ma thèse et à Bordeaux cette année depuis septembre.

Un grand merci à tous les chercheurs et chercheuses qui m'ont à un moment aidé dans cette voix en m'invitant à parler à un séminaire, en me posant des questions sur mes travaux qui m'ont fait ouvrir de nouvelles portes ou encore en me narrant leurs expériences pour enrichir la mienne. La liste n'en est sûrement pas exhaustive, mais je tiens à remercier Nathalie Aubrun, Jeremie Bettinelli, Alin Bostan, Johanne Cohen, Alain Denise, Benjamin Hellouin, Matthieu Josuat Verges, Aaron Lauve, Jean-Christophe Novelli, Amy Pang, Vincent Pilaud, François Pirot, Jean-Yves Thibon, Nicolas Thiéry, Burkhardt Wolff,

Mike Zabrocki. Je souhaite également remercier à part Catherine Dubois, Marie-Madeleine Legendre et Danièle Sarrat dont les enseignements m'ont particulièrement marqués.

Je repense souvent au contre-labo, né pendant le confinement chez Balthazar et Monica avec Daniel qui est venu plusieurs fois aussi. Nous avons eu plusieurs séances de discussions et de réflexions (souvent autour du *Project Euler*). Ces journées terminaient à chaque fois autour d'un bon repas. Merci à vous et à tous les autres doctorants et doctorantes que j'ai pu croiser durant ces années, en particulier Aram, Arthur, Baptiste, Doriann (que je remercie de plus pour être à Bordeaux avec moi cette année), Joël, Jonas, Justine, Noémie.

Je remercie bien sûr, pour leur soutien, tous les membres de l'équipe de combinatoire et interactions du LaBRI, que j'ai rejointe en septembre de cette année.

Dans un autre cadre, un de mes plus grands remerciements va à mes deux parents. Tout d'abord mon père, Marek, pour sa patience qui le définit en tant que personne, qui est pour moi une grande source d'inspiration aussi bien dans ma vie professionnelle que personnelle, mais aussi pour tout le reste ! Et surtout ma mère, Sophie, qui mériterait au moins son propre diplôme. La qualité pédagogique de ce document lui est grandement due. En effet il m'a fallu expliquer et illustrer tous les détails de mes résultats à quelqu'un qui n'y connaît pas grand chose mais qui a fait preuve d'une ténacité sans pareille. Elle m'aura aidé plus que quiconque dans la rédaction de ce manuscrit. Mais elle mérite encore mille mercis supplémentaires pour tout le reste ! Un merci se dirige vers deux amis très proches de mes parents, Pierre-Marie et Jean-Nicolas.

Je remercie également les autres membres de ma famille, Juliette, Agnès et Jean-François (ils comprendront que je nomme la barre à mine dans ces remerciements). Également Sandrine, Yann, Martin et Nina, que je vois plus régulièrement maintenant que je suis à Bordeaux.

Je ne peux pas ne pas parler de musique dans ces remerciements. *Mammouth Brass Band* est le nom d'une fanfare poilue et cuivrée, qui donne envie de danser ! Mais c'est également un groupe d'amis importants pour moi et dont chaque membre mérite un énorme merci ! Il y a un autre groupe que je tiens à remercier, *Pouet*, pour ce côté funk qui manquait à ma vie. Enfin j'ai une pensée pour toutes les personnes avec qui j'ai pu jouer de la musique.

Pour toutes les fois où j'ai pu venir, je remercie la bande *S(o)ki* avec qui je vais bientôt repartir en janvier prochain.

Merci à Yoliste pour les courses à bords d'Alphonse, une expérience inoubliable.

Enfin je remercie mes amis qui me manquent dès que je ne les vois plus pendant plusieurs jours, Zac mon ami de toujours, Laure et Maï-thi qui valent de l'or, Matthieu et Cyprien à qui je me confierais aveuglément, Aersyo ou Rémi ou Arbre ou Pinpaul, et tant d'autres aux pseudos plus ou moins fous !

Table des matières

I	Préliminaire	27
1	Objets combinatoires	29
1.1	Mots	30
1.1.1	Définitions utilitaires	30
1.1.2	Permutations	34
1.1.3	Mots tassés	38
1.2	Forêts et arbres	42
1.2.1	Arbres binaires et arbres plans	42
1.2.2	Décorations des arbres	45
1.2.3	Greffes d'arbres	46
1.3	Grammaires et séries génératrices	49
2	Ordres partiels	55
2.1	Définitions	55
2.2	Ordres faibles sur les permutations	57
2.2.1	Groupe symétrique	57
2.2.2	Ordres faibles droit et gauche	59
2.3	Ordres sur les mots tassés	60
2.3.1	Généralisation de l'ordre faible droit	61
2.3.2	Deux généralisations de l'ordre faible gauche	61
2.3.3	Ordre de raffinement	65
3	Algèbres de Hopf combinatoires	67
3.1	Définitions générales	67
3.2	Exemples d'algèbres de Hopf combinatoires	70
3.2.1	L'algèbre FQSym	70
3.2.2	L'algèbre WQSym	71
3.2.3	L'algèbre $\mathcal{H}_{P,R}^{\mathcal{D}}$	74
3.3	Bigèbres bidendriformes	75
3.3.1	Définitions	75
3.3.2	Reprise des exemples	76
3.3.3	Théorème de Cartier-Milnor-Moore pour les bigèbres bidendriformes	81

II Automorphism of WQSym	83
4 Combinatorics of biplane forests	87
4.1 Dual (Red)	87
4.1.1 Decomposition of packed words through maximums	87
4.1.2 Red-forests from decomposed packed words using ϕ	91
4.2 Primal (Blue)	102
4.2.1 Decompositions of packed words through last letter	103
4.2.2 Blue-forests from decomposed packed words using ψ	106
4.3 Intervals for half shuffle products	114
5 Bases for totally primitive elements	119
5.1 Dual (Red)	119
5.1.1 Decomposition through maximums and totally primitive elements .	119
5.1.2 The new basis \mathbb{P}	120
5.2 Primal (Blue)	122
5.2.1 Decomposition through last letter and totally primitive elements .	122
5.2.2 The new basis \mathbb{O}	123
6 Isomorphism between WQSym and WQSym^*	125
6.1 A combinatorial solution to an algebraic problem	125
6.2 Full decomposition of packed words into bicolored forests	126
6.3 An involution on packed words	134
6.4 Main theorem	136
III Autour de WQSym, résultats et perspectives	137
7 Opérades et mots tassés	141
7.1 Rappels sur les opérades	141
7.1.1 Définitions de bases	141
7.1.2 Exemples d'opérades	142
7.1.3 Algèbre sur une opérade	144
7.2 Une algèbre dupliciale déformée sur les mots tassés	145
7.3 Une L -algèbre sur les mots tassés	147
7.4 Une algèbre Bigraft sur les mots tassés	148
IV Expérimentation avec SageMath	149
8 Notebooks en ligne	153
8.1 Objets combinatoires	153
8.1.1 Mots tassés	153
8.1.2 Arbres biplans	156

8.2	Ordres partiels	156
8.3	Bases de WQSym	158
8.4	Autres bases de WQSym	162
8.5	Séries génératrices	164

Table des figures

1	Arbres binaires de tailles 1, 2 et 3	16
2	Diagrammes de boites : décomposition en descentes globales	20
3	Diagrammes de boites : opérations \blacktriangleright et \blacktriangleleft	21
4	Exemple d'un calcul de changement de base : \mathbb{P} vers \mathbb{R}	22
5	Exemple de l'involution sur un mot en hexadécimal	23
1.1	Diagrammes de boites : sous-mot et facteur	31
1.2	Exemples de diagrammes de boites	31
1.3	Diagrammes de boites : concaténation	32
1.4	Diagrammes de boites : concaténation décalée	32
1.5	Diagrammes de boites : produit de mélange	33
1.6	Diagrammes de boites : décomposition en descentes globales (permutation)	36
1.7	Diagrammes de boites : produit de mélange décalé (permutations)	36
1.8	Diagrammes de boites : produit de mélange des valeurs (permutations)	37
1.9	Diagrammes de boites : décomposition en descentes globales (mot tassé)	40
1.10	Diagrammes de boites : produit de mélange décalé (mots tassés)	40
1.11	Diagrammes de boites : produit de mélange des valeurs (mots tassés)	41
1.12	Arbres binaires de tailles 1, 2 et 3	44
1.13	Arbres plans de tailles 1, 2, 3 et 4	44
1.14	Exemple de bijection entre une forêt plane et un arbre binaire	44
1.15	Arbres plans décorés	45
1.16	Arbre plan numéroté selon l'ordre préfixe	45
1.17	Angles d'un arbre plan	46
1.18	Ordre sur les angles d'un arbre plan	47
1.19	Greffé d'une forêt plane sur un arbre plan	47
1.20	Greffé d'une forêt plane sur une autre forêt plane	48
1.21	Produit de greffe entre deux forêts planes	48
1.22	Demi-produit de greffe entre deux forêts planes	49
2.1	Exemple de diagramme de Hasse	56
2.2	Exemples de sous-posets	57
2.3	Ordre faible droit \leq_R sur les permutations pour les tailles 3 et 4	60
2.4	Ordre faible gauche \leq_L sur les permutations pour les tailles 3 et 4	61
2.5	Ordre faible droit \leq_R sur les mots tassés de taille 3	62

2.6	Ordre faible droit \leq_R sur les mots tassés de composition 221	62
2.7	Ordre faible gauche avec chevauchement \leq_L^* sur les mots tassés de taille 3	63
2.8	Ordre faible gauche \leq_L sur les mots tassés de taille 3	64
2.9	Une composante de l'ordre faible gauche \leq_L sur les mots tassés de taille 5	64
2.10	Ordre de raffinement \leq^* sur les mots tassés de taille 3	65
4.1	Box digrams : the operation \blacktriangleright	88
4.2	Box digrams : the operation \blacktriangleleft	104
4.3	Box digrams : the operation \blacktriangleleft	115
4.4	Box digrams : the operation \blacktriangledown	115
4.5	Component of the left weak order for the two half products of \mathbb{Q}_{12} and \mathbb{Q}_{121} . $\overline{\preceq}$ on the left and $\overline{\succeq}$ on the right.	116
4.6	Component of the right weak order for the two half products of \mathbb{R}_{11} and \mathbb{R}_{121} . $\overline{\preccurlyeq}$ on the left and $\overline{\succcurlyeq}$ on the right.	117
6.1	Commutative diagram of maps on bicolored-packed forests	136
A.1	Change-of-basis matrix from \mathbb{P}_3 to \mathbb{R}_3	175
A.2	Change-of-basis matrix from \mathbb{Q}_3 to \mathbb{R}_3	178
A.3	Change-of-basis matrix from \mathbb{P}_4 to \mathbb{R}_4	184
A.4	Change-of-basis matrix from \mathbb{O}_4 to \mathbb{Q}_4	185
A.5	Change-of-basis matrix from \mathbb{Q}_4 to \mathbb{R}_4	186

Liste des tableaux

1.1	Nombre de permutations de taille inférieure à 9	34
1.2	Nombre de permutations irréductibles de taille inférieure à 9	35
1.3	Nombre de mots tassés de taille inférieure à 9	38
1.4	Nombre de mots tassés irréductibles de taille inférieure à 9	40
1.5	Nombre d'arbres binaires et plans de taille inférieure à 9	43
3.1	Dimensions of homogeneous components for WQSym , Prim and TPrim	85
4.1	Number of biplane forests and biplane trees.	92
4.2	Equivalence between trees of [Foi11] and red-packed trees.	102
6.1	Number of both red-and-blue-irreducible packed words and permutations.	129
7.1	Nombres de Catalan inférieurs à 9.	143
7.2	Nombre d'arbres biplans de taille inférieure à 8.	143
7.3	Nombre de forêts biplanes de taille inférieure à 8.	144
A.1	All packed words of size smaller than 3 and forests associated to it.	176
A.2	Packed words of size 4 and associated forests (part 1).	177
A.3	Packed words of size 4 and associated forests (part 2).	178
A.4	Packed words of size 4 and associated forests (part 3).	179
A.5	Packed words of size 4 and associated forests (part 4).	180
A.6	Packed words of size 4 and associated forests (part 5).	181
A.7	The automorphism of WQSym $_{\leq 3}$	182
A.8	The involution $w \mapsto \widehat{w}$ on packed words of size 4.	183
A.9	The involution $w \mapsto \widehat{w}$ on red-irreducible packed words that are not blue-irreducible of size 5.	183

Introduction

Cette thèse se situe en combinatoire algébrique, à l'interface entre la combinatoire énumérative et l'analyse d'algorithmes d'une part et le calcul symbolique et algébrique d'autre part. L'objectif principal est de comprendre la combinatoire d'une structure algébrique nommée **WQSym** pour «Word Quasi-symmetric functions».

Contexte

La combinatoire est l'étude des différentes configurations possibles d'un ensemble fini d'éléments vérifiant des règles simples. Par exemple, on peut prendre un paquet de cartes et s'intéresser à l'énumération de tous les ordres possibles de ces cartes.

Prenons, par exemple, 8 cartes numérotées de 1 à 8, le nombre d'ordonnancements de ces cartes peut se calculer. Ranger ces cartes dans n'importe quel ordre revient à choisir la première carte (8 choix possibles) puis la deuxième carte parmi les restantes (7 choix possibles) et ainsi de suite jusqu'à la dernière carte. Le nombre de possibilités s'obtient en multipliant le nombre de choix à chaque étape et vaut donc $8 \times 7 \times \dots \times 1 = 8! = 40\,320$. De façon générale, avec n cartes, le nombre d'ordonnancements est donné par la factorielle de n . Ces ordonnancements sont appelés «permutations» et ce sont des objets combinatoires importants dans cette thèse.

Séries génératrices

L'outil algébrique principal en combinatoire est la notion de série génératrice qui enregistre le nombre de configurations, pour toutes les tailles des éléments étudiés, dans une expression algébrique de type polynôme, fraction rationnelle ou série. Si nous revenons sur l'exemple des permutations, nous pouvons définir la série formelle suivante :

$$\mathcal{P}(z) := \sum_{n \geq 0} n! z^n = 1 + z + 2z^2 + 6z^3 + 24z^4 + 120z^5 + \dots \quad (1)$$

Dans cette série, le coefficient du monôme z^i est le nombre de permutations de taille i . Il existe bien 120 façons différentes d'ordonner 5 cartes numérotées de 1 à 5 comme on peut le lire dans le monôme $120z^5$.

Un autre exemple très classique d'objet combinatoire sont les arbres binaires. Un arbre binaire est soit une feuille, soit un nœud avec un fils droit et un fils gauche qui sont également des arbres binaires.

En partant de cette définition, il est possible de calculer le nombre d'arbres binaires T_n d'une taille n donnée si l'on connaît le nombre d'arbres binaires des tailles plus petites par la formule

$$T_n = \sum_{i=0}^{n-1} T_i T_{n-1-i}. \quad (2)$$

L'intérêt des séries génératrices est que cette formule peut se traduire en une équation sur la série génératrice qui s'exprime très simplement :

$$\mathcal{T}_B(z) = 1 + z\mathcal{T}_B(z)^2. \quad (3)$$

Dans le cas des arbres binaires, on obtient une simple équation du second degré, que l'on résout par radical. La solution de cette équation nous donne donc le nombre d'arbres binaires par taille :

$$\mathcal{T}_B(z) = \frac{1 - \sqrt{1 - 4z}}{2z} = 1 + z + 2z^2 + 5z^3 + 14z^4 + 42z^5 + 132z^6 + \dots \quad (4)$$

Voici tous les arbres binaires de tailles 1, 2 et 3 :

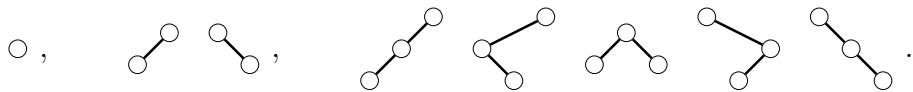


FIGURE 1 – Arbres binaires de tailles 1, 2 et 3.

La définition des arbres ci-dessus peut être formalisée par une grammaire récursive. En théorie des langages, une grammaire est un formalisme permettant de définir une syntaxe et donc un langage formel, c'est-à-dire un ensemble de mots admissibles sur un alphabet donné. On peut généraliser cette idée pour définir des ensembles d'objets combinatoires autre que des mots comme des arbres, des forêts...

Il existe une procédure appelée «méthode symbolique» [NW75] qui permet de traduire directement la grammaire en un système d'équations. Cette méthode a été développée plus avant par Flajolet et Sedgewick [FS09] qui, grâce aux séries génératrices, étudient le comportement asymptotique d'une famille d'objets définie par une grammaire. Cette technique a été utilisées avec succès par Flajolet, Salvy et Zimmerman [FZS88, FSZ91] pour le calcul automatique de complexité de certains algorithmes.

Le principal défaut des séries génératrices est que l'information encodée dans l'exposant d'un monôme est très pauvre : c'est un nombre entier ou éventuellement quelques nombres entiers grâce aux séries génératrices multivariées. On perd en particulier une bonne partie de la structure des objets sous-jacents. En conséquence, avec la méthodologie des séries

génératrices, on sait seulement répondre aux questions du type : quel est le nombre d'opérations si l'entrée de l'algorithme est de taille n où n est un entier ? Ce qui est souvent trop restreint. Nous voudrions par exemple pouvoir compter les permutations ayant un type cyclique donné, le type cyclique d'une permutation étant la partition de n donnée par les longueurs de ses cycles.

Algèbres de Hopf

Les algèbres de Hopf sont une généralisation possible des séries génératrices qui permettent d'enregistrer une information plus fine (comme une liste arbitrairement longue de nombres, une permutation, un arbre, *etc.*). Par exemple, pour compter les permutations en fonction des classes de descentes (qui sont un sous-ensemble de l'ensemble $\{1, \dots, n\}$), Gessel introduit l'algèbre des fonctions quasi-symétriques **QSym** [Ges83]. La duale de cette algèbre est introduite par les six auteurs Gelfand, Krob, Lascoux, Leclerc, Retakh et Thibon en tant qu'algèbre des fonctions symétriques non-commutatives **NCSF** [GKL⁺95].

Les algèbres de Hopf sont des structures formelles dont les bases sont indexées par des objets combinatoires. Dans une telle algèbre, le produit (comme dans les algèbres classiques) correspond à une façon d'assembler des objets et le coproduit à une façon de désassembler ces mêmes objets. Pour guider l'intuition du lecteur, prenons un jeu de cartes. L'ensemble de toutes les coupes possibles (somme formelle) de ce jeu de cartes correspond à un coproduit. Maintenant, supposons que notre paquet de cartes soit déjà coupé en deux paquets plus petits, l'ensemble de toutes les possibilités de mélange américain (somme formelle) de ces deux paquets correspond au produit dans l'algèbre de Hopf.

C'est l'idée qui sous-tend la définition de l'algèbre de Hopf **FQSym** [MR95, DHT01] dont la base est indexée par les permutations. Dans la suite du manuscrit, nous étudierons cette algèbre de façon détaillée. Voici un exemple du produit de mélange (inspiré du mélange des cartes) et du coproduit correspondant. Dans le produit, on remarque en particulier que l'ordre de 1 et 2 (en noir) reste toujours le même ainsi que l'ordre de 3 et 4 (en rouge). Cet exemple est repris et expliqué plus en détail dans l'[exemple 1.25](#).

$$12 \sqcup 43 = 1243 + 1423 + 1432 + 4123 + 4132 + 4312. \quad (5)$$

Le coproduit est lui défini en découplant la permutation en deux à toutes les positions possibles.

$$\Delta(2431) = \epsilon \otimes 2431 + 1 \otimes 321 + 12 \otimes 21 + 132 \otimes 1 + 2431 \otimes \epsilon. \quad (6)$$

Dans une algèbre de Hopf, le produit \cdot et le coproduit Δ vérifient une relation de compatibilité $\Delta(a \cdot b) = \Delta(a) \cdot \Delta(b)$ qui signifie que l'on obtient le même résultat si on commence par composer puis découper que si l'on découpe puis compose. Par exemple, sur les permutations, sur un exemple minimal, cela donne :

$$\Delta(1 \cdot 1) = \Delta(12 + 21) \tag{7}$$

$$= \epsilon \otimes 12 + 1 \otimes 1 + 12 \otimes \epsilon + \epsilon \otimes 21 + 1 \otimes 1 + 21 \otimes \epsilon \tag{8}$$

$$\Delta(1) \cdot \Delta(1) = (\epsilon \otimes 1 + 1 \otimes \epsilon) \cdot (\epsilon \otimes 1 + 1 \otimes \epsilon) \tag{9}$$

$$= \epsilon \otimes 12 + \epsilon \otimes 21 + 1 \otimes 1 + 1 \otimes 1 + 12 \otimes \epsilon + 21 \otimes \epsilon. \tag{10}$$

Cette coexistence du produit et du coproduit au sein d'une même structure algébrique fait que la **duale** d'une algèbre de Hopf est aussi une algèbre de Hopf. En effet, on définit l'algèbre duale d'une algèbre de Hopf en "inversant" les notions de produits et coproduits. Voilà par exemple, le produit et le coproduit dans l'algèbre duale de **FQSym**. On remarque cette fois que l'ordre relatif des deux nombres placés aux deux premières positions (et aux deux dernières positions) reste le même tandis que le coproduit "découpe" la permutation selon les valeurs et non plus selon les positions comme c'était le cas dans la primale.

$$12 \sqsubseteq 21 = 1243 + 1342 + 1432 + 2314 + 2413 + 3412, \tag{11}$$

$$\Delta(2431) = \epsilon \otimes 2431 + 1 \otimes 132 + 21 \otimes 21 + 231 \otimes 1 + 2431 \otimes \epsilon. \tag{12}$$

Au milieu des années 2000, la communauté s'est employée à définir des structures d'algèbre de Hopf sur une très grande variété d'objets combinatoires, par exemple **QSym** pour les compositions d'entiers [Ges83], **FQSym** pour les permutations [MR95, DHT01], **PBT** pour les arbres binaires [LR98], Connes-Kreimer pour les forêts générales [CK98], **WQSym** pour les mots tassés [Hiv99], **PQSym** pour les fonctions de parking [NT03]... Dans un deuxième temps, la communauté s'est tournée vers la question de mettre de l'ordre dans cette zoologie de structures algébriques. Deux approches sont possibles : la première consiste à chercher les relations que peuvent présenter deux algèbres de Hopf (sous-algèbre, quotient, morphisme, dualité). La deuxième approche cherche à enrichir la structure de Hopf, par exemple en raffinant les opérations introduisant ainsi de nouveaux types d'algèbres, par exemple, via la théorie des opérades. C'est dans cette mouvance que se situe le présent travail.

Ainsi la question centrale de cette thèse porte sur l'auto-dualité de l'algèbre de Hopf **WQSym**, cela revient à construire un isomorphisme explicite entre **WQSym** et son algèbre duale.

En 1999, dans le but de trouver une version non commutative des fonctions quasi-symétriques de Gessel, Hivert [Hiv99] définit l'algèbre des fonctions quasi-symétriques en mots notée **WQSym**, algèbre indexée par les mots tassés (ou de manière équivalente les surjections ou encore les partitions ordonnées d'ensembles). Rappelons que les fonctions quasi-symétriques interviennent, entre autres, dans l'énumération des permutations ainsi que dans l'étude de la théorie des représentations des algèbres de Hecke. L'auto-dualité de l'algèbre **WQSym** est conjecturée en 2001 par Duchamp-Hivert-Thibon [DHT01] en même temps que la liberté de l'algèbre de Lie primitive de **FQSym**.

Bigèbres Bidendriformes

Pour répondre à ces questions, Foissy a eu l'idée de scinder en deux non seulement le produit mais aussi le coproduit. Il a ainsi défini une notion de bigèbres [Foi07] où les demi-produits et demi-coproduits vérifient des relations de compatibilité, toutes plus fines que celles des algèbres de Hopf. Le grand intérêt de cette notion est que la structure d'une bigèbre bidendriforme est extrêmement rigide. Elle ne dépend en effet que de la suite des dimensions des composantes d'un degré donné.

Grâce à cette rigidité, il parvient à démontrer plusieurs propriétés sur les bigèbres bidendriformes. La majorité de ces propriétés sont des conséquences de son théorème principal :

Une bigèbre bidendriforme est générée librement par l'espace qu'il nomme “l'espace des totalement primitifs” en tant qu'algèbre dendriforme.

En effet, une question fondamentale lorsque l'on travaille avec une algèbre est de définir un ensemble d'éléments permettant d'engendrer par le produit l'ensemble de l'algèbre. Dans une algèbre de Hopf, on peut trouver cet ensemble par la méthode des **éléments primitifs**. Par exemple, dans **FQSym**, l'algèbre est engendrée par les permutations sans descentes globales lorsqu'on se place dans une certaine base. La structure, plus rigide, des bigèbres bidendriformes permet de définir un ensemble de générateurs plus restreints (les totalement primitifs) qui engendent l'algèbre en utilisant les demi-produits. Foissy illustre son théorème par un exemple sur **FQSym**. En revanche, il ne donne pas de construction générale pour obtenir l'espace des totalement primitifs.

En conséquence, ce théorème, que l'on peut qualifier de théorème de rigidité, implique l'auto-dualité d'une bigèbre bidendriforme, car l'algèbre et sa duale sont générées librement par leurs espaces de totalement primitifs, et que ces derniers sont de mêmes dimensions. Par ailleurs, les travaux de Foissy donnent une construction pour obtenir un isomorphisme bidendriforme à partir d'un isomorphisme linéaire entre les espaces de totalement primitifs. En particulier, cela réduit le problème algébrique à un problème combinatoire : construire des bases des éléments totalement primitifs et les mettre en bijection.

Auto-dualité de **WQSym**

En 2006, Novelli-Thibon [NT06] prouvent que **WQSym** est une bigèbre bidendriforme. La conjecture de 2001 est donc résolue, **WQSym** est auto-duale. Cependant, contrairement à **FQSym** où l'inversion des permutations correspond à un isomorphisme bidendriforme vers sa duale, aucun isomorphisme explicite entre **WQSym** et sa duale n'existe encore.

Un premier isomorphisme entre **WQSym** et sa duale est construit par Vargas [Var19] en 2019, la construction se base sur des ordres partiels. Cependant, elle ne fait pas intervenir d'éléments totalement primitifs et n'est pas compatible avec la structure bidendriforme. Non seulement une partie de la structure est donc perdue dans l'isomorphisme de Vargas, mais sa construction n'est donc, a priori, pas généralisable aux bigèbres bidendriformes.

Contributions

Le problème principal auquel nous nous sommes consacrés est la construction d'un isomorphisme bidendrifforme explicite entre **WQSym** et sa duale. Pour ce faire il nous a fallu chercher des bases de l'algèbre adaptées aux structures bidendrifformes définies par Foissy. Une question intermédiaire était donc de décrire un sous-ensemble de mots tassés ayant la bonne cardinalité par degrés. Cette question est de nature purement combinatoire.

Combinatoire des mots tassés et des arbres biplans

Dans toute la combinatoire en lien avec les algèbres de Hopf, par exemple sur les permutations ou les mots, l'idée de "décomposition" est fondamentale. On cherche à décomposer au maximum les objets combinatoires pour obtenir des éléments dits "irréductibles" qui permettent d'engendrer les éléments de l'algèbre. C'est le cas par exemple de la décomposition en descentes globales sur les permutations, définie par Aguiar et Sottile [AS05], que nous avons généralisée aux mots tassés. La décomposition en descentes globales d'un mot est unique et tous les facteurs sont dits irréductibles (voir figure 2).

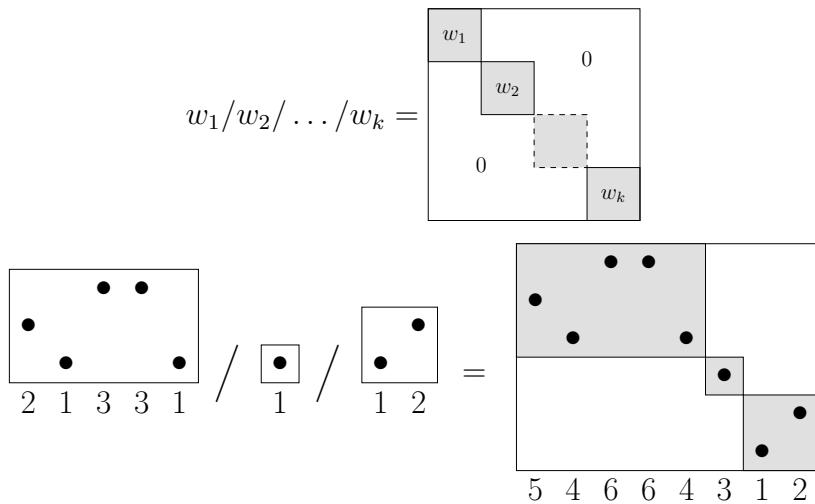


FIGURE 2 – Diagrammes de boîtes : décomposition en descentes globales.

Parmi les nombreuses bases connues de **WQSym**, certaines sont dites multiplicatives et le produit est défini par la concaténation décalée des indices (voir le paragraphe 2.7 de [NT06] pour des exemples). On peut en déduire que le nombre de mots tassés irréductibles de taille n est égal à la dimension de l'espace des primitifs de l'algèbre au degré n . Cependant, si l'on souhaite définir des bases compatibles avec la structure bidendrifforme, on a besoin d'un découpage supplémentaire de ces éléments irréductibles. En effet, on souhaite définir une famille d'éléments dits "totalement irréductibles" qui puisse indiquer une base des "espaces des totalement primitifs".

Pour mener à bien cette recherche, nous avons créé deux opérations binaires \blacktriangleright et \blacktriangleleft qui agissent sur les mots tassés (voir lemme 6.6). L'opération \blacktriangleright donne un sens particulier à la position des maximums et l'opération \blacktriangleleft donne un sens particulier à la valeur de la dernière lettre.

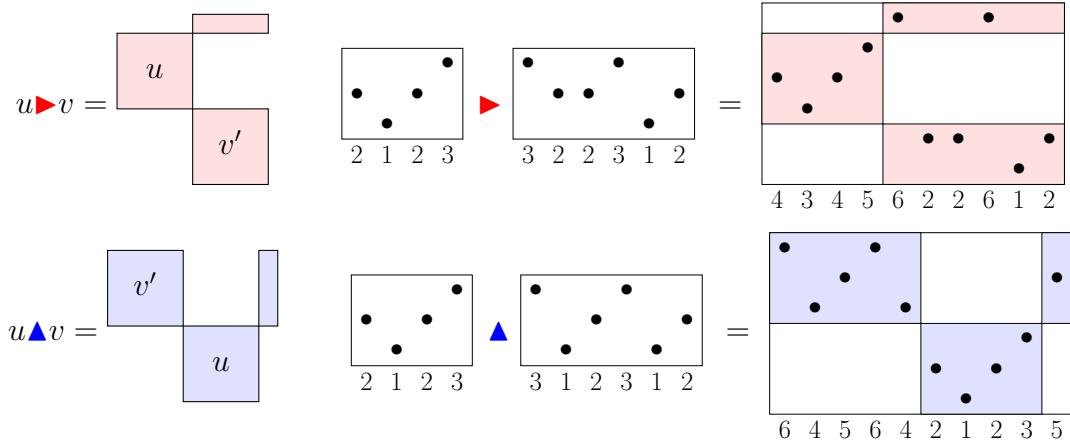


FIGURE 3 – Diagrammes de boîtes : opérations \blacktriangleright et \blacktriangleleft .

La concaténation décalée / est à la décomposition en descentes globales ce que l'opération \blacktriangleright (resp. \blacktriangleleft) est à la décomposition rouge (resp. bleu). Nous avons donc défini deux nouvelles décompositions des mots qui font intervenir ces nouvelles opérations. Nous avons défini les mots irréductibles rouges (resp. bleus) comme les mots irréductibles ne pouvant pas s'écrire sous la forme $u\blacktriangleright v$ (resp. $u\blacktriangleleft v$). Nous avons enfin prouvé que les cardinaux de ces ensembles sont égaux aux dimensions de “l'espace des totalement primitifs”.

Pour décrire ces décompositions, nous avons introduit une nouvelle famille d'objets combinatoires : les **forêts d'arbres biplans**. Les décompositions précédentes appliquées récursivement permettent de prouver les bijections suivantes :

- Les forêts biplanes et les mots tassés,
- Les arbres biplans et les mots tassés irréductibles,
- Les arbres biplans sans fils gauche à la racine et les mots tassés irréductibles rouges ou bleus (en fonction de la couleur de l'arbre).

Chaque “branche” d'un arbre correspond, d'une part du point de vu combinatoire, à une opération sur les mots eux-mêmes, et d'autre part, à une opération algébrique compatible avec la structure bidendrifforme.

Nouvelles bases de **WQSym**

Ces arbres biplans sont donc la forme idéale pour indexer des nouvelles bases (\mathbb{P} et \mathbb{O}), que nous avons créées, de l'algèbre **WQSym** et sa dual, car il est aisément d'en extraire un sous-ensemble qui définit les deux premières bases des espaces totalement primitifs de

WQSym et sa duale. On peut voir en [figure 4](#) un exemple d'un calcul de changement de base depuis la nouvelle base \mathbb{P} vers la base \mathbb{R} déjà connue.

$$\begin{aligned} \mathbb{P} \circlearrowleft_{(1,3)} \circlearrowright_1 &= \mathbb{P} \circlearrowleft_1 \circlearrowright_{(1,3)} = (\mathbb{P} \circlearrowleft_1 \circlearrowright_{(1,3)} - \mathbb{P} \circlearrowleft_1 \circlearrowright_1) \circlearrowleft \Phi_{1,3}(\mathbb{P} \circlearrowleft_1) \\ &= \mathbb{R}_{14342} + \mathbb{R}_{41342} + \mathbb{R}_{43142} + \mathbb{R}_{43412} - \mathbb{R}_{24341} - \mathbb{R}_{42341} - \mathbb{R}_{43241} \end{aligned}$$

FIGURE 4 – Exemple d'un calcul de changement de base : \mathbb{P} vers \mathbb{R} .

Ce qu'il manque est une bijection entre les mots tassés irréductibles rouges et les mots tassés irréductibles bleus. Pour cela nous avons imaginé une construction qui alterne les trois opérations \blacktriangleright , \blacktriangleleft et $/$. Nous avons alors obtenu des forêts d'arbres bicolores. Le grand intérêt de cette construction, c'est qu'une fois que l'on sait la réaliser, un simple échange de couleurs permet d'obtenir un isomorphisme bidendriforme, ce qui répond à notre question initiale. De plus, cet isomorphisme correspond naturellement à une involution sur les mots tassés. Dans la [figure 5](#), à partir d'un mot w écrit en hexadécimal, on peut voir la décomposition alternant les trois opérations, la représentation en diagramme de boîtes et l'arbre rouge-bleu associé. Nous avons ensuite appliqué l'échange de couleurs et représenté sous forme de diagramme de boîtes et de décomposition le mot associé w' à ce nouvel arbre bleu-rouge.

Lien entre bigèbre bidendriforme et d'autres opérades

Nous avons donc travaillé sur **WQSym** en tant que bigèbre bidendriforme. Dans notre construction, \blacktriangleright et $/$ vérifient de façon inattendue des relations similaires à l'opérade duplique déformée [BDO20]. À symétrie près, les relations entre \blacktriangleleft et $/$ sont les mêmes. Ces relations font apparaître une nouvelle application de l'opérade duplique déformée appliquée sur des mots tassés.

Comme dit plus haut, nous avons introduit une nouvelle famille d'objets combinatoires : les **forêts d'arbres biplans**. Après une recherche sur OEIS [SI20], il a été constaté que le nombre d'arbres biplans à n noeuds est égal à la dimension de la composante homogène de la L -algèbre [Ler07] libre à un seul générateur.

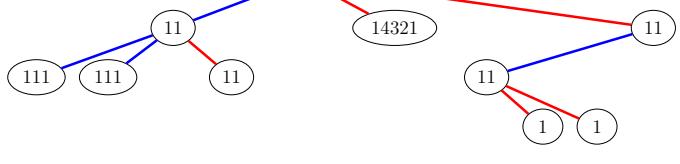
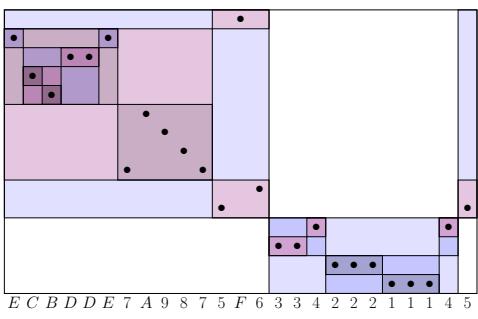
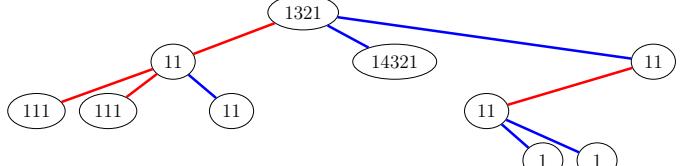
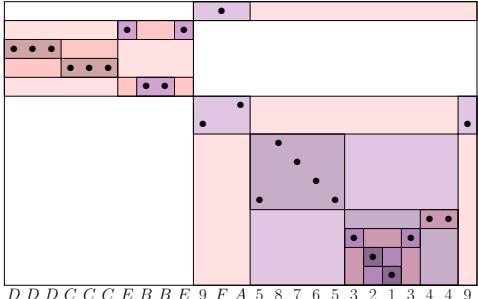
Nous prouvons que l'ensemble des mots tassés, munis des nouvelles opérations que nous avons définies, décrit une algèbre sur ces opérades et en donnons des sous-ensembles de générateurs.

Méthodologie

Notre processus de recherche se base sur une approche par **expérimentation par ordinateur**, en utilisant le logiciel SageMath. Cette phase permet de guider les pistes de

$w = DDDCCCEBBE9FA587653213449$

$w = ((111/111) \blacktriangleright (11 \blacktriangle 11)) \blacktriangleright ((14321 / (((1/1) \blacktriangle 11) \blacktriangleright 11)) \blacktriangle 1321)$



$w' = ((111/111) \blacktriangle (11 \blacktriangleright 11)) \blacktriangle (((((1/1) \blacktriangleright 11) \blacktriangle 11) / 14321) \blacktriangleright 1321)$

$w' = ECBDE7A9875F633422211145$

FIGURE 5 – Exemple de l’involution sur un mot en hexadécimal

recherche en implantant des tests de plus en plus précis. Tous les objets combinatoires que nous avons définis ont été implantés sous forme d’objets SageMath, ce qui nous a permis de faire des tests sur d’autres propriétés et de répondre plus facilement à des questions annexes auxquelles nous n’avions pas initialement pensé. Une fois que tous les tests sont validés, nous avons une base solide et une compréhension forte des interactions entre les objets pour nous aider à prouver nos résultats. Nous avons écrit en tout environ 8000 lignes de codes qui concernent l’algèbre **WQSym** et les objets combinatoires que nous utilisons. Une partie de ce code a pour vocation d’intégrer le logiciel SageMath pour que toute la communauté puisse l’utiliser. Il y a un travail d’intégration à SageMath de deux tickets, sur les mots tassés et sur plusieurs bases de **WQSym**. Des tutoriels sous la forme de notebooks sont également disponibles sur GitHub pour que ce code soit visible par les autres chercheurs.

Plan de la thèse

La partie I intitulée “Préliminaire” est composée de trois chapitres et présente les outils nécessaires à la compréhension de cette thèse. Nous y définissons quelques classes d’objets combinatoires dans le chapitre 1, à savoir les permutations, les mots tassés et les arbres binaires et plans. Ensuite, nous abordons la notion d’ordre partiel dans le chapitre 2. Cette notion permet principalement d’avoir une meilleure approche des objets combinatoires qui nous intéressent. Enfin, nous terminons cette partie en présentant trois algèbres de Hopf combinatoires dans le chapitre 3 : **FQSym**, **WQSym**, $\mathcal{H}_{P,R}^D$. De plus, dans ce troisième chapitre, nous énonçons deux théorèmes structuraux de Foissy qui prouvent, entre autres, l’auto-dualité des bigèbres bidendriformes (théorème 3.27 et corollaire 3.28).

La partie II, en anglais car issue de notre article [Mlo22a] éponyme de cette thèse et dont nous sommes le seul auteur, constitue le résultat principal de cette thèse. Nous y développons tous les outils nécessaires à la construction d’un isomorphisme bidendriforme entre **WQSym** et sa duale (équivalent à l’automorphisme bidendriforme de **WQSym**). Cette partie se divise en trois chapitres.

Le chapitre 4 est dévolu à la construction combinatoire de forêts biplanes rouges et bleues (définitions 4.32 et 4.67) qui sont nos premiers ingrédients clé. Elles représentent une décomposition récursive des mots tassés selon leurs descentes globales (lemme 1.45) et la position des lettres maximales (lemme 4.7) ou de la valeur de la dernière lettre (lemme 4.46). Nous prouvons que le cardinal d’un certain sous-ensemble spécifique de ces arbres biplans correspond à la dimension de l’espace des primitifs (théorème 4.40), de même pour un plus petit sous-ensemble de ces arbres à la dimension de l’espace des totalement primitifs (théorème 4.76).

Dans le chapitre 5 nous construisons deux nouvelles bases (\mathbb{O} et \mathbb{P} définitions 5.5 et 5.13) de **WQSym** et sa duale, qui contiennent chacune une base des éléments primitifs et totalement primitifs (théorèmes 5.7 et 5.15). Pour ce faire, nous décomposons l’espace des éléments totalement primitifs en une certaine somme directe qui correspond à la décomposition combinatoire des mots tassés (lemmes 5.4 et 5.12).

Cette partie s’achève avec le chapitre 6 où nous expliquons pourquoi avoir les bases \mathbb{O} et \mathbb{P} est suffisant pour construire une infinité d’automorphismes bidendriforme de **WQSym**. Enfin, nous donnons un isomorphisme explicite basé sur une involution sur les mots tassés. La définition de cette bijection requiert une nouvelle sorte de forêts mélangeant les notions de forêts rouges et bleues que l’on nomme des forêts bicolores.

La partie III est consacrée aux questions qui sont soulevées par les résultats de la partie II. En effet, nous nous sommes penché sur une de ces questions, nous avons étudié les liens entre **WQSym** et certaines opérades qui n’ont a priori pas de lien direct avec l’opérade dendriforme.

Pour finir, la partie IV expose notre utilisation de SageMath dans notre méthodologie de recherche. Nous avons réalisé 5 notebooks que nous présentons dans cette partie.

Les 3 premiers notebooks permettent, entre autres, de vérifier nos résultats de la [partie II](#). Ceci étant un de leurs aspects positifs, un autre étant que ce code est à la disposition des chercheurs en combinatoire algébrique qui peuvent l'utiliser pour leurs propres recherches similaires.

Les 2 derniers notebooks témoignent de notre méthodologie de recherche. Nous y trouverons des conjectures que nous avons pu écarter. Faisant face à de nombreuses classes combinatoires différentes, nous avons énuméré ces objets de plusieurs façons différentes ce qui nous a permis d'en avoir une vision plus générale et une compréhension plus fine.

Première partie

Préliminaire

Chapitre 1

Objets combinatoires

Les références de base pour les études des objets combinatoires sont ces deux livres [Lot02, FS09]. Les auteurs y décrivent avec précision de nombreux objets combinatoires et les liens qui existent entre ces objets.

La première définition de [FS09] est celle d'une classe combinatoire.

Définition 1.1. Une **classe combinatoire** \mathcal{C} est un ensemble fini ou dénombrable sur lequel une fonction de taille est définie telle que :

- la taille d'un élément est un nombre positif,
- le nombre d'éléments d'une taille donnée est fini.

On dit que la classe est graduée.

Dans ce chapitre, nous redéfinissons uniquement les objets combinatoires qui nous serons utiles pour les travaux de cette thèse.

Nous commencerons par les **permutations** et les **mots tassés**, que nous voyons tous deux comme des mots particuliers sur l'alphabet des entiers naturels.

Nous nous intéresserons ensuite aux **arbres binaires** et aux **arbres plans**. Pour chacune de ces classes combinatoires, nous verrons des façons de les assembler et de les désassembler. Nous scinderons ces assemblages comme dans [Lod01] pour obtenir des **demi-produits**. Enfin, nous terminerons ce chapitre par l'étude des **séries génératrices** des classes combinatoires que nous aurons définies.

1.1 Mots

Nous redonnons ici les définitions basiques liées aux mots. Les permutations sont des mots particuliers simples à apprécier. L'étude de ces permutations nous aidera pour la compréhension de l'étude des mots tassés. La classe des mots tassés étant la classe combinatoire la plus centrale de cette thèse, nous avons besoin d'être le plus précis possible pour en faciliter la lecture.

1.1.1 Définitions utilitaires

Un **alphabet**, que l'on va noter X , est un ensemble totalement ordonné. Les éléments de cet ensemble X sont appelés des **lettres**. Dans cette thèse, l'unique alphabet employé sera l'alphabet des entiers strictement positifs $\mathbb{N}_{>0}$. En pratique, dans les exemples, nous utiliserons toujours des lettres entre 1 et 9 pour éviter les confusions.

Un **mot** w est une suite finie de lettres. Nous allons noter $w = w_1 w_2 \cdots w_n$ une suite de n lettres. On pourra ainsi désigner la lettre en position $i \in [1 \dots n]$ par la notation w_i . Par exemple prenons $w = 14313$, alors w_2 désigne la lettre 4.

La **taille** (ou **longueur**) $|w|$ d'un mot w est définie par son nombre de lettres n , on notera $|w| = n$. Remarquons qu'il existe un mot de taille 0 appelé le **mot vide** que l'on notera ϵ .

La **lettre maximale** $\max(w)$ d'un mot w est définie par la plus grande lettre de l'alphabet présente dans le mot w . Par convention, nous dirons que la lettre maximale du mot vide est nulle, $\max(\epsilon) = 0$.

Soit un mot $w = w_1 w_2 \cdots w_n$,

- un **sous-mot** de w provient d'une sélection de ses lettres. Plus formellement, pour toute suite $I = [i_1, i_2, \dots, i_k]$ vérifiant $1 \leq i_1 < i_2 < \dots < i_k \leq n$ le mot $w_I := w_{i_1} w_{i_2} \dots w_{i_k}$ est un sous-mot de w . Par exemple si w est de taille au moins 7 alors $w' = w_2 w_4 w_7$ est un sous-mot de w . Il existe donc au plus 2^n sous-mots différents de w . De plus, on notera $w|^{\leq j}$ (resp. $w|^{\geq j}$) le sous-mot de w contenant toutes les lettres inférieures (resp. strictement supérieures) à une valeur j .
- un **facteur** de w provient d'une sélection non vide de lettres successives de w . Par exemple si w est de taille au moins 5 alors $w'' = w_2 w_3 w_4 w_5$ est un facteur de w . Il existe donc au plus $\binom{n+1}{2}$ facteurs différents de w . De plus, on notera $w|_{\leq i}$ (resp. $w|_{>i}$) le **préfixe** (resp. **suffixe**) de w contenant toutes les i premières lettres (resp. lettres strictement après la $i^{\text{ème}}$ lettre).

Exemple 1.2. Prenons par exemple $w = 121351$ comme mot de *taille* 6 dont la *lettre maximale* est 5. Nous pouvons dire que 215 est un *sous-mot* de w , mais pas un *facteur*, alors que 135 en est un. Enfin deux autres *sous-mots particuliers* sont par exemple $w|^{\leq 2} = 1211$ et $w|^{\geq 2} = 35$ et deux autres *facteurs particuliers* sont par exemple $w|_{\leq 2} = 12$ et $w|_{>2} = 1351$.

Dans cette thèse, nous illustrerons nos propos à l'aide de **diagrammes de boîtes** aux coordonnées cartésiennes. Sur ces diagrammes, les positions se lisent de la gauche vers la droite et les valeurs de bas en haut.

Nous commençons dans la [figure 1.1](#) par illustrer l'[exemple 1.2](#) qui est simple.

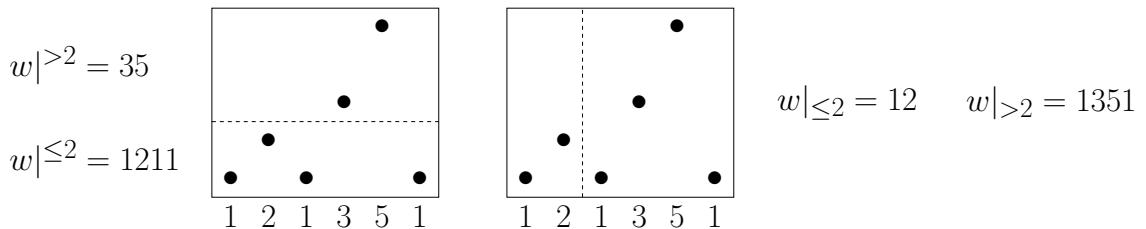


FIGURE 1.1 – Diagrammes de boîtes : sous-mot et facteur.

Ces diagrammes nous serviront à appréhender visuellement des décompositions complexes de mots grâce à différentes couleurs. La transparence ordonnera les décompositions.

Exemple 1.3. Nous avons dans la [figure 1.2](#) plusieurs exemples :

- la représentation du mot 313,
- la représentation du mot 214313,
- la représentation en diagrammes de boîtes nous permet de visualiser des décompositions complexes de mots comme par exemple la factorisation-rouge du mot 3415251 (voir [lemme 4.7](#)),
- enfin, la représentation du cas général de la factorisation-rouge-bleue (voir [définition 6.8](#)) dans laquelle il peut être vu clairement que la factorisation-bleue est faite après la factorisation-rouge et seulement sur une partie grâce à la transparence des boîtes.

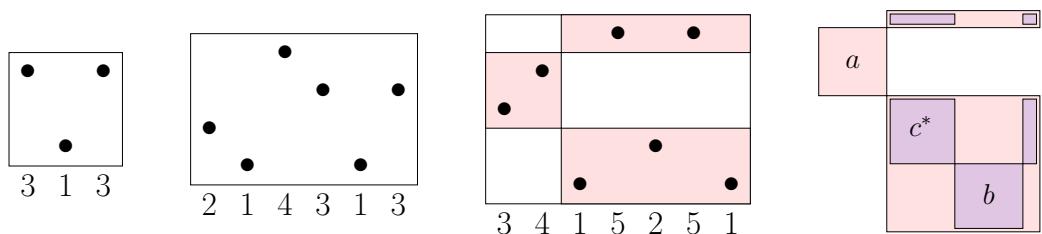


FIGURE 1.2 – Exemples de diagrammes de boîtes.

Définition 1.4. La **concaténation** de deux mots u et v que l'on notera $u \cdot v$ est obtenue en écrivant le mot v à la suite du mot u .

Exemple 1.5. Par exemple, prenons $u = 4112$ et $v = 313$, nous avons alors $u \cdot v = 4112313$.

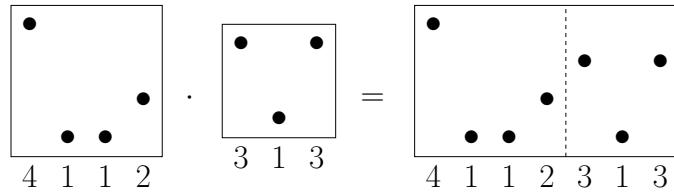


FIGURE 1.3 – Diagrammes de boîtes : concaténation.

Définition 1.6. L'**augmentation** (ou **décalage**) d'un mot w d'une valeur i , qui sera notée $w^{[i]}$, est obtenue en ajoutant la valeur i à chacune des lettres.

Exemple 1.7. Prenons $w = 4112$ que nous augmentons de la valeur 3, nous obtenons alors $w^{[3]} = 7445$.

Définition 1.8. Nous pouvons alors définir la **concaténation décalée** à gauche u/v (resp. à droite $u\backslash v$) de deux mots u et v . On augmente le mot de gauche (resp. droite) de la valeur maximale de l'autre mot, puis on effectue une concaténation. Formellement nous écrivons : $u/v := u^{[\max(v)]} \cdot v$ (resp. $u\backslash v := u \cdot v^{[\max(u)]}$).

Exemple 1.9. Prenons $u = 4112$ et $v = 313$ nous avons $u/v = 7445313$ et $u\backslash v = 4112757$.

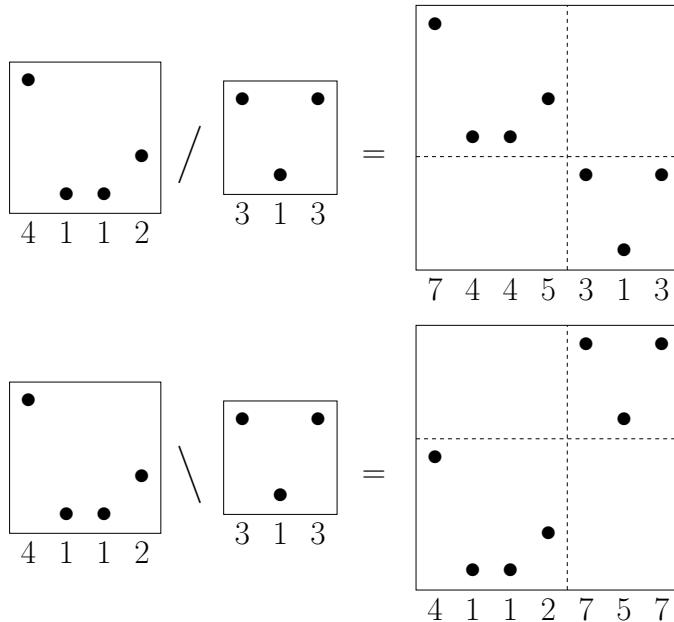


FIGURE 1.4 – Diagrammes de boîtes : concaténation décalée.

Jusqu'ici, on a vu des opérations sur les mots dont le résultat est un seul mot. Comme nous allons le voir prochainement, en combinatoire, on a souvent des opérations plus complexes, dont le résultat est une combinaison linéaire formelle d'objets. Plus formellement,

cela revient à se placer dans un espace vectoriel dont la base est l'ensemble des objets. On considérera que le corps de base de l'espace vectoriel est un corps quelconque \mathbb{K} de caractéristique nulle.

Définition 1.10. Le **produit de mélange** de deux mots u et v se note $u \sqcup v$ et représente la somme formelle de tous les mélanges possibles des deux mots u et v . Ce mélange se définit récursivement par le cas de base $u \sqcup \epsilon := \epsilon \sqcup u := u$ et le cas général :

$$ua \sqcup vb := (u \sqcup vb) \cdot a + (ua \sqcup v) \cdot b \quad (1.1)$$

où u et v sont des mots, a et b sont des lettres et ua et vb sont les mots obtenus par les concaténations $u \cdot a$ et $v \cdot b$.

Exemple 1.11. Dans l'exemple suivant, nous représentons les lettres du second mot en rouge pour mieux visualiser le résultatat de l'opération.

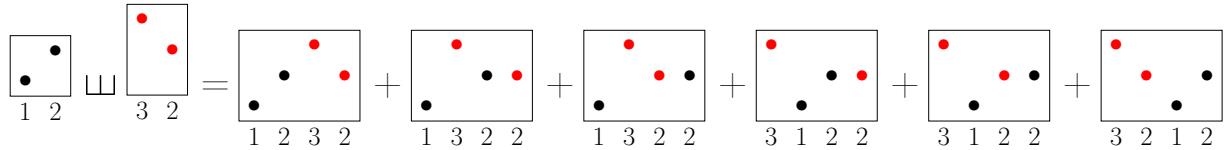


FIGURE 1.5 – Diagrammes de boîtes : produit de mélange.

$$12 \sqcup 32 = 1232 + 1322 + 1322 + 3122 + 3122 + 3212, \quad (1.2)$$

$$= 1232 + 2 \times 1322 + 2 \times 3122 + 3212. \quad (1.3)$$

Il est important de remarquer que, lors du mélange, toutes les lettres du premier mot (resp. du second mot) restent dans le même ordre dans tous les mélanges possibles. Nous remarquons de plus que, dans cet exemple, certains mots apparaissent deux fois même s'ils proviennent de deux mélanges différents. En pratique, dans la suite de cette thèse, les produits de mélanges se feront toujours sur des mots dont les ensembles de lettres sont disjoints. (Pour plus d'exemples de produits de mélange, voir [exemples 1.25 et 1.48](#).)

Ce produit qui mélange les positions des lettres peut être découpé en deux demi-produits qui seront utiles pour la [section 3.3](#) sur les algèbres bidendrifformes. Pour cela, nous pouvons remarquer que la définition récursive du produit de mélange ([équation \(1.1\)](#)) contient deux termes. On va alors définir ces deux termes comme des demi-produits, ainsi la somme de ces deux demi-produits sera égale au produit complet.

Le demi-produit de mélange droit (resp. gauche) correspond à tous les mélanges possibles dont la dernière lettre provient du mot de droite (resp. gauche).

Définition 1.12 ([Lod01, Exemple 5.4.(a)]). Les deux **demi-produits de mélange** gauche \prec et droite \succ sont définis respectivement sur les mots par :

$$ua \prec vb := (u \sqcup vb) \cdot a \quad \text{et} \quad ua \succ vb := (ua \sqcup v) \cdot b, \quad (1.4)$$

où u et v sont des mots, a et b sont des lettres et ua et vb sont les mots obtenus par les concaténations $u \cdot a$ et $v \cdot b$.

Exemple 1.13. Reprenons les mêmes mots que dans l'[exemple 1.11](#). Dans l'[équation \(1.2\)](#), quand la dernière lettre est noire, elle provient du mot de gauche, quand elle est rouge elle provient du mot de droite. Ainsi :

$$12 \prec 32 = 1322 + 3122 + 3212 \quad \text{et} \quad 12 \succ 32 = 1232 + 1322 + 3122.$$

Ces produits mélange donc les positions des lettres des deux mots. Nous verrons par la suite que pour les permutations et mots tassés, nous pourrons, en plus, définir un produit de mélange sur les valeurs.

1.1.2 Permutations

Définition 1.14. Une **permutation** σ de taille n est un mot où chaque lettre entre 1 et n apparaît exactement une fois. Par convention le mot vide ϵ est une permutation.

Pour $n \in \mathbb{N}$, on note \mathfrak{S}_n l'ensemble des $n!$ permutations de taille n .

Enfin $\mathfrak{S} := \bigsqcup_{n \in \mathbb{N}} \mathfrak{S}_n$ désigne l'ensemble de toutes les permutations.

n	1	2	3	4	5	6	7	8	9	OEIS [SI20]
\mathfrak{S}_n	1	2	6	24	120	720	5 040	40 320	362 880	A000142

TABLE 1.1 – Nombre de permutations de taille inférieure à 9.

Exemple 1.15. Voici toutes les permutations de tailles 1, 2 et 3 :

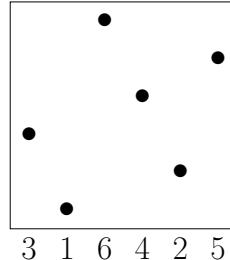
$$1, \quad 12 \quad 21, \quad 123 \quad 132 \quad 213 \quad 231 \quad 312 \quad 321.$$

Définition 1.16. La **standardisation** $\text{std}(w)$ d'un mot w consiste à numérotter les lettres des plus petites aux plus grandes et de la gauche vers la droite. On obtient alors une permutation que l'on nommera le standardisé.

Exemple 1.17. Soit $w = 429727$, nous numérotions d'abord les 2 qui sont les plus petites lettres de la gauche vers la droite par un 1 puis un 2. Ensuite comme il n'y a pas de 3 dans le mot, on numérote le 4 par un 3 et ainsi de suite comme montré ci-dessous jusqu'à avoir $\text{std}(w) = 316425$.

$$\begin{array}{ccccccc} 4 & 2 & 9 & 7 & 2 & 7 \\ \hline . & 1 & . & . & 2 & . \\ 3 & 1 & . & . & 2 & . \\ 3 & 1 & . & 4 & 2 & 5 \\ 3 & 1 & 6 & 4 & 2 & 5 \end{array}$$

Un mot w est une permutation si et seulement si $\text{std}(w) = w$. Dans la représentation en diagramme de boîtes, une permutation a exactement un point par ligne et par colonne, comme on peut le voir sur le diagramme suivant sur la permutation $\text{std}(w)$.



Définition 1.18. [AS05] Une **descente globale** d'une permutation σ est une position c telle que toutes les lettres avant et à la position c sont strictement plus grandes que toutes les lettres après la position c .

Exemple 1.19. Les descentes globales de la permutation $\sigma = 64785312$ sont les positions 5 et 6. En effet, toutes les lettres du préfixe 64785 sont plus grandes que le suffixe 312 et cela est aussi vrai pour 647853 et 12. Cet exemple est illustré figure 1.6.

Définition 1.20. Une permutation σ est **irréductible** si elle est non vide et qu'elle n'a pas de descente globale.

n	1	2	3	4	5	6	7	8	9	OEIS [SI20]
p_n	1	1	3	13	71	461	3 447	29 093	273 343	A003319

TABLE 1.2 – Nombre de permutations irréductibles de taille inférieure à 9.

Exemple 1.21. Voici toutes les permutations irréductibles de tailles 1, 2, 3 et 4 :

$$\begin{array}{ccccccccc} 1, & & 12, & & 123 & 132 & 213, \\ 1234 & 1243 & 1324 & 1342 & 1423 & 1432 & 2134 & 2143 & 2314 \end{array}$$

$$2413 \quad 3124 \quad 3142 \quad 3214.$$

Lemme 1.22. Toute permutation σ admet une unique factorisation en descentes globales notée $\sigma = \sigma_{(1)} / \sigma_{(2)} / \dots / \sigma_{(k)}$ tel que $\sigma_{(i)}$ est irréductible pour tout i .

Exemple 1.23. La décomposition en descentes globales de la permutation $\sigma = 64785312$ est donc $\sigma = 31452/1/12$ comme on peut le voir sur la figure 1.6. La permutation $n \cdot n-1 \cdot \dots \cdot 1$ se décompose comme $1/1/\dots/1$.

Nous avons déjà vu le produit de mélange sur les mots (définition 1.10). Pour l'appliquer aux permutations, il suffit d'appliquer une augmentation sur l'une des deux permutations de façon analogue à la concaténation décalée. Nous pourrons également définir des demi-produits de mélange sur les permutations, cela nous sera utile pour la section 3.3 sur les algèbres bidendrifiques.

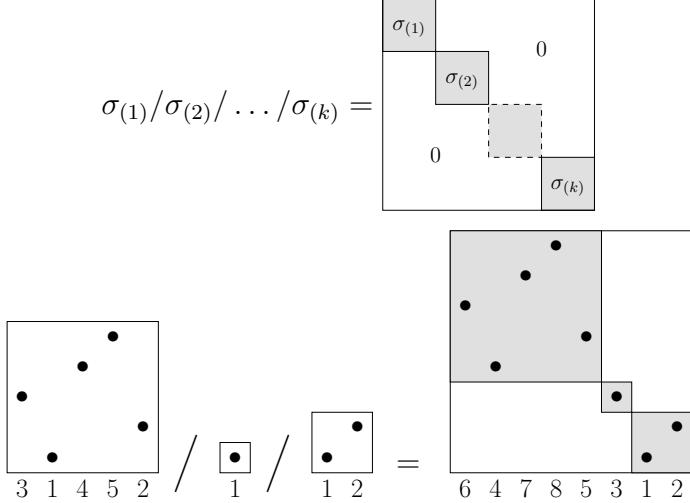


FIGURE 1.6 – Diagrammes de boîtes : décomposition en descentes globales (permutation).

Définition 1.24. Le **produit de mélange décalé** à droite sur deux permutations se note $\sigma \bar{\sqcup} \mu$ et se définit par $\sigma \bar{\sqcup} \mu := \sigma \sqcup \mu^{[\max(\sigma)]}$. Toutes les lettres de la permutation de droite μ sont augmentées de la valeur maximale de la permutation de gauche σ .

Exemple 1.25. $12 \bar{\sqcup} 21 = 12 \sqcup 43 = 1243 + 1423 + 1432 + 4123 + 4132 + 4312$.

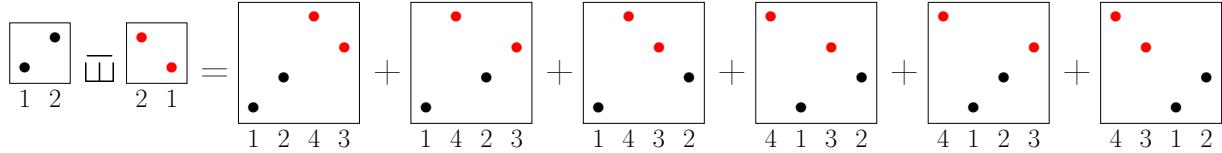


FIGURE 1.7 – Diagrammes de boîtes : produit de mélange décalé (permutations).

On peut remarquer que dans tous les mots w de cette somme formelle on a les sous-mots standardisés $w|^{≤ 2} = 12$ et $\text{std}(w|^{> 2}) = \text{std}(43) = 21$ qui correspondent aux deux mots initiaux. Cette remarque prendra tout son sens avec la notion de coproduit dual ([définition 3.7](#)) que nous traiterons dans le [chapitre 3](#) sur les algèbres de Hopf.

Définition 1.26. Les deux **demi-produits de mélange décalés** à droite se notent $\sigma \bar{\prec} \mu$ et $\sigma \bar{\succ} \mu$ et se définissent par $\sigma \bar{\prec} \mu := \sigma \prec \mu^{[\max(\sigma)]}$ et $\sigma \bar{\succ} \mu := \sigma \succ \mu^{[\max(\sigma)]}$ où toutes les lettres de la permutation de droite μ sont augmentées de la valeur maximale de la permutation de gauche σ .

Exemple 1.27. $12 \bar{\prec} 21 = 1432 + 4132 + 4312 \quad \text{et} \quad 12 \bar{\succ} 21 = 1243 + 1423 + 4123$.

Nous avons défini un produit de mélange sur les positions, nous pouvons également le faire sur les valeurs et, comme avant, couper ce produit en deux demi-produits.

Définition 1.28. Le **produit de mélange des valeurs** (ou **produit de convolution**) de deux permutations σ et μ se note $\sigma \boxplus \mu$ et se définit par :

$$\sigma \boxplus \mu := \sum_{\substack{\sigma' \cdot \mu' \in \mathfrak{S} \\ \sigma = \text{std}(\sigma') \\ \mu = \text{std}(\mu')}} \sigma' \cdot \mu'. \quad (1.5)$$

Exemple 1.29. Dans l'exemple suivant, nous représentons les lettres du second mot en bleu pour mieux visualiser le résultatat de l'opération.

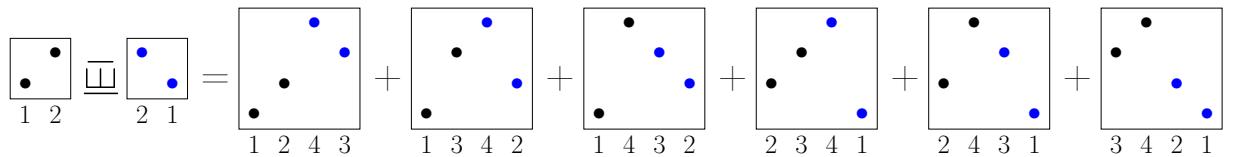


FIGURE 1.8 – Diagrammes de boîtes : produit de mélange des valeurs (permutations).

$$12 \boxplus 21 = 1243 + 1342 + 1432 + 2314 + 2413 + 3412.$$

On peut remarquer que dans tous les mots w de cette somme formelle on a cette fois les facteurs standardisés $\text{std}(w|_{\leq 2}) = 12$ et $\text{std}(w|_{> 2}) = 21$ qui correspondent aux deux mots initiaux. Cette remarque prendra tout son sens avec la notion de coproduit dual ([définition 3.7](#)) que nous traiterons dans le [chapitre 3](#) sur les algèbres de Hopf.

Dans les deux demi-produits des [définitions 1.12 et 1.26](#), on a séparé en fonction du mot dont provenait la dernière lettre. Cette fois, comme nous mélangeons les valeurs, nous allons séparer en fonction de la provenance de la lettre de valeur maximale.

Définition 1.30. Les deux **demi-produits de mélange des valeurs** de deux permutations σ et μ se définissent par :

$$\sigma \preceq \mu := \sum_{\substack{\sigma' \cdot \mu' \in \mathfrak{S} \\ \sigma = \text{std}(\sigma') \\ \mu = \text{std}(\mu') \\ \max(\sigma') > \max(\mu')}} \sigma' \cdot \mu' \quad \text{and} \quad \sigma \succeq \mu := \sum_{\substack{\sigma' \cdot \mu' \in \mathfrak{S} \\ \sigma = \text{std}(\sigma') \\ \mu = \text{std}(\mu') \\ \max(\sigma') < \max(\mu')}} \sigma' \cdot \mu'. \quad (1.6)$$

Exemple 1.31. $12 \preceq 21 = 1432 + 2431 + 3421$ et $12 \succeq 21 = 1243 + 1342 + 2341$.

Définition 1.32. Une permutation σ admet un **motif** μ s'il existe un sous-mot de σ dont le standardisé vaut μ . Au contraire, on dit que σ **évite le motif** μ s'il n'en existe pas. Un motif μ est nécessairement une permutation.

Exemple 1.33. Voici toutes les permutations de taille 4 évitant le motif 321 :

$$\begin{array}{cccccccc} 1234 & 1243 & 1324 & 1342 & 1423 & 2134 & 2143 \\ 2314 & 2341 & 2413 & 3124 & 3142 & 4123 & 4132. \end{array}$$

1.1.3 Mots tassés

La structure de cette section sur les mots tassés sera identique à celle sur les permutations. En effet, toutes les permutations sont des mots tassés, toutefois la réciproque n'est pas vraie. Pour éviter les redondances de lecture, le tableau de synthèse suivant présente les similitudes et quelques différences entre les permutations et les mots tassés. Néanmoins, comme ces notions sont fondamentales sur les mots tassés pour cette thèse, nous redonnons formellement les définitions complètes.

définition 1.16	définition 1.39	La standardisation et le tassemement.
définition 1.18	définition 1.41	Les descentes globales.
définition 1.20	définition 1.43	Les permutations et mots tassés irréductibles.
lemme 1.22	lemme 1.45	La factorisation en descentes globales.
définition 1.24	définition 1.47	Le produit de mélange décalé.
définition 1.26	définition 1.49	Les demi-produits de mélange décalé.
définition 1.28	définition 1.51	Le produit de mélange des valeurs, pour les mots tassés on définit aussi le produit de mélange sans chevauchement.
définition 1.30	définition 1.54	Les demi-produits de mélange des valeurs, pour les mots tassés on définit aussi les demi-produits de mélange sans chevauchement.
	définition 1.55	

Définition 1.34. Un **mot tassé** (*packed word*) w de taille n et de maximum $m \leq n$ est un mot où chaque lettre entre 1 et m apparaît au moins une fois. Par convention le mot vide ϵ est un mot tassé.

Pour $n \in \mathbb{N}$, on note \mathbf{PW}_n l'ensemble des mots tassés de taille n .

Enfin $\mathbf{PW} := \bigsqcup_{n \in \mathbb{N}} \mathbf{PW}_n$ désigne l'ensemble de tous les mots tassés.

n	1	2	3	4	5	6	7	8	9	OEIS [SI20]
\mathbf{PW}_n	1	3	13	75	541	4 683	47 293	545 835	7 087 261	A000670

TABLE 1.3 – Nombre de mots tassés de taille inférieure à 9.

Exemple 1.35. Le mot 4152142 n'est pas tassé car la lettre 3 n'apparaît pas alors que la lettre maximale est $5 > 3$. Voici tous les mots tassés de tailles 1, 2 et 3 dans l'ordre lexicographique :

$$\begin{array}{ccccccccc} 1, & & 11 & 12 & 21, \\ 111 & 112 & 121 & 122 & 123 & 132 & 211 & 212 & 213 & 221 & 231 & 312 & 321 \end{array}$$

Pour le lecteur plus habitué aux partitions d'ensembles ordonnées, il existe une bijection classique entre les mots tassés et les partitions d'ensembles ordonnées. La partition correspondant à un mot tassé $w = w_1 w_2 \cdots w_n$ est obtenue en plaçant l'indice i dans le bloc w_i .

Exemple 1.36. Le mot tassé 121 est associé à $\{\{13\}, \{2\}\}$ et le mot 113223 à $\{\{12\}, \{45\}, \{36\}\}$.

Définition 1.37. Soit un mot tassé w de taille n et de maximum m . La **composition** $\text{comp}(w)$ est le mot de taille m dont la première lettre est égale au nombre de 1 dans w , la deuxième lettre est égale au nombre de 2 dans w et ainsi de suite. Le **poids** de la composition est défini comme la somme de ses lettres et vaut donc n .

Exemple 1.38. $\text{comp}(134234133) = 2142$.

Définition 1.39. Le **tassement** $\text{pack}(w)$ d'un mot w est un analogue de la standardisation des permutations. Cela consiste à numérotier les lettres des plus petites aux plus grandes en conservant les égalités de lettres. On obtient alors un mot tassé.

Exemple 1.40. Soit $w = 429727$, nous numérotions d'abord les 2 qui sont les plus petites lettres par 1. Ensuite comme il n'y a pas de 3 dans le mot, on numérote le 4 par un 2 et ainsi de suite comme montré ci-dessous jusqu'à obtenir $\text{pack}(w) = 214313$.

4	2	9	7	2	7	
.	1	.	.	1	.	
2	1	.	.	1	.	
2	1	.	3	1	3	
2	1	4	3	1	3	

Un mot w est un tassé si et seulement si $\text{pack}(w) = w$. Dans la représentation en diagramme de boîtes, tasser un mot revient à retirer les lignes vides, un mot tassé a donc exactement un point par colonne et au moins un par ligne, comme on peut le voir ci-dessous.



Définition 1.41. Une **descente globale** d'un mot tassé w est une position c telle que toutes les lettres avant et à la position c sont strictement plus grandes que toutes les lettres après la position c .

Exemple 1.42. Les descentes globales du mot tassé $\sigma = 54664312$ sont les positions 5 et 6. En effet, toutes les lettres du préfixe 54664 sont plus grandes que le suffixe 312 et cela est aussi vrai pour 546643 et 12. Cet exemple est illustré [figure 1.9](#).

Définition 1.43. Un mot tassé w est **irréductible** s'il est non vide et qu'il n'a pas de descente globale.

n	1	2	3	4	5	6	7	8	9	OEIS [SI20]
p_n	1	2	8	48	368	3 376	35 824	430 512	5 773 936	A095989

TABLE 1.4 – Nombre de mots tassés irréductibles de taille inférieure à 9.

Exemple 1.44. Voici tous les mots tassés irréductibles de tailles 1, 2 et 3 :

$$1, \quad 11 \quad 12, \quad 111 \quad 112 \quad 121 \quad 122 \quad 123 \quad 132 \quad 212 \quad 213.$$

Lemme 1.45. *Tout mot tassé w admet une unique factorisation en descentes globales notée $w = w_{(1)}/w_{(2)}/\dots/w_{(k)}$ tel que $w_{(i)}$ est irréductible pour tout i .*

Exemple 1.46. La décomposition en descentes globales du mot tassé $w = 54664312$ est donc $w = 21331/1/12$ comme on peut le voir sur la figure 1.9.

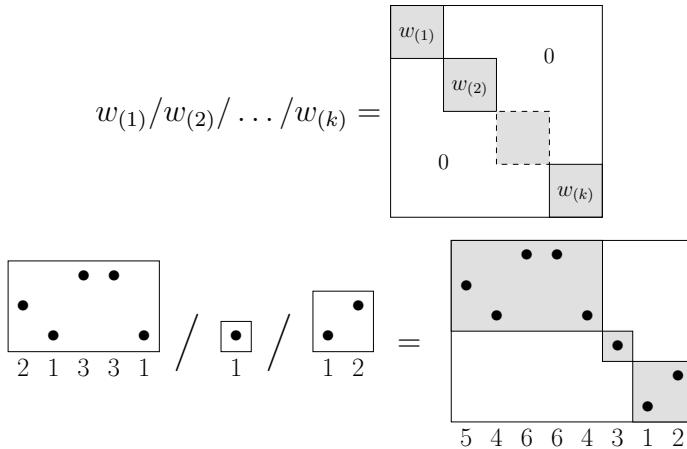


FIGURE 1.9 – Diagrammes de boîtes : décomposition en descentes globales (mot tassé).

Définition 1.47. Le produit de mélange décalé à droite sur deux mots tassés se note $u \bar{\sqcup} v$ et se définit par $u \bar{\sqcup} v := u \sqcup v^{[\max(u)]}$. Toutes les lettres du mot tassé de droite v sont augmentées de la valeur maximale du mot tassé de gauche u .

Exemple 1.48. $12 \bar{\sqcup} 11 = 12 \sqcup 33 = 1233 + 1323 + 1332 + 3123 + 3132 + 3312$.

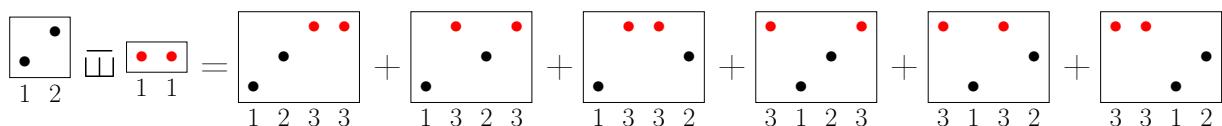


FIGURE 1.10 – Diagrammes de boîtes : produit de mélange décalé (mots tassés).

Définition 1.49. On peut également définir les deux **demi-produits de mélange décalés** à droite sur les mots tassés comme $u \overline{\prec} v := u \prec v^{[\max(u)]}$ et $u \overline{\succ} v := u \succ v^{[\max(u)]}$ où toutes les lettres du mot tassé de droite v sont augmentées de la valeur maximale du mot tassé de gauche u .

Exemple 1.50. $12 \overline{\prec} 33 = 1332 + 3132 + 3312$ et $12 \overline{\succ} 33 = 1233 + 1323 + 3123$.

Comme le tassemement (pack) est l'analogue de la standardisation (std), le produit de mélange sur les valeurs, qui est défini à partir de la standardisation pour les permutations ([définition 1.28](#)), est défini avec le tassemement sur les mots tassés. La différence principale est, qu'avec le tassemement, des lettres provenant des deux mots peuvent avoir la même valeur finale. C'est pourquoi nous allons définir un deuxième produit de mélange sur les valeurs pour les mots tassés, le produit de mélange sans chevauchement.

Définition 1.51. Le **produit de mélange des valeurs** (ou **produit de convolution**) de deux mots tassés u et v se note $u \underline{\boxplus} v$ et se définit par :

$$u \underline{\boxplus} v := \sum_{\substack{u' \cdot v' \in \mathbf{PW} \\ u = \text{pack}(u') \\ v = \text{pack}(v')}} u' \cdot v'. \quad (1.7)$$

Définition 1.52. Le **produit de mélange sans chevauchement** de deux mots tassés u et v se note $u \overline{\boxplus} v$ et se définit en ajoutant la contrainte que les lettres des deux parties sont distinctes :

$$u \overline{\boxplus} v := \sum_{\substack{u' \cdot v' \in \mathbf{PW} \\ u = \text{pack}(u') \\ v = \text{pack}(v')} \\ \forall i, j, u'_i \neq v'_j} u' \cdot v'. \quad (1.8)$$

Exemple 1.53. $12 \underline{\boxplus} 11 = 1211 + 1222 + 1233 + 1322 + 2311$, $12 \overline{\boxplus} 11 = 1233 + 1322 + 2311$.

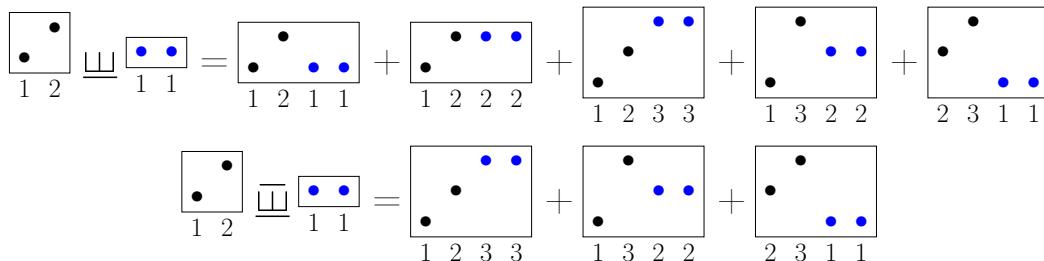


FIGURE 1.11 – Diagrammes de boîtes : produit de mélange des valeurs (mots tassés).

Définition 1.54. Les deux **demi-produits de mélange des valeurs** de deux mots tassés u et v se définissent par :

$$u \preceq v := \sum_{\substack{u' \cdot v' \in \mathbf{PW} \\ u = \text{pack}(u') \\ v = \text{pack}(v') \\ \max(u') > \max(v')}} u' \cdot v' \quad \text{et} \quad u \succeq v := \sum_{\substack{u' \cdot v' \in \mathbf{PW} \\ u = \text{pack}(u') \\ v = \text{pack}(v') \\ \max(u') \leq \max(v')}} u' \cdot v'. \quad (1.9)$$

Définition 1.55. Les deux **demi-produits de mélange sans chevauchement** de deux mots tassés u et v se définissent par :

$$u \overline{\preceq} v := \sum_{\substack{u' \cdot v' \in \mathbf{PW} \\ u = \text{pack}(u') \\ v = \text{pack}(v') \\ \max(u') > \max(v') \\ \forall i, j, u'_i \neq v'_j}} u' \cdot v' \quad \text{et} \quad u \overline{\succeq} v := \sum_{\substack{u' \cdot v' \in \mathbf{PW} \\ u = \text{pack}(u') \\ v = \text{pack}(v') \\ \max(u') < \max(v') \\ \forall i, j, u'_i \neq v'_j}} u' \cdot v'. \quad (1.10)$$

Exemple 1.56.

$$\begin{aligned} 12 \preceq 11 &= 1211 + 1322 + 2311 & \text{et} & 12 \succeq 11 = 1222 + 1233. \\ 12 \overline{\preceq} 11 &= 1322 + 2311 & \text{et} & 12 \overline{\succeq} 11 = 1233. \end{aligned}$$

1.2 Forêts et arbres

Avant de définir la notion d’arbre en combinatoire, c’est la notion de **forêts** qui se rapproche le plus de tout ce que nous avons vu jusqu’ici. En effet, les forêts sont des suites ordonnées d’arbres et donc des mots dont les lettres sont des arbres.

Nous allons commencer par définir des arbres binaires, puis des arbres plans (ou planaires ou ordonnés). Nous verrons dans le [chapitre 4](#) une autre famille d’arbres, les arbres biplans, émanant, comme leur nom l’indique, des deux notions précédentes.

1.2.1 Arbres binaires et arbres plans

Définition 1.57. Un **arbre binaire** (fini) se définit récursivement par le cas de base de l’arbre vide noté \emptyset et par le cas général d’un nœud $\text{Node}([t_\ell, t_r])$ avec exactement deux enfants, un arbre gauche t_ℓ et un arbre droit t_r . Nous utiliserons également la représentation suivante :

$$t = \begin{array}{c} \diagup \diagdown \\ t_\ell \quad t_r \end{array}.$$

L’ensemble des arbres binaires T_B satisfait les axiomes d’induction structurelle, plus précisément :

- T_B est un ensemble ayant \emptyset pour élément,
- Node est une fonction de $T_B \times T_B$ dans T_B ,

- \emptyset n'est pas dans l'image de la fonction Node,
- Node est injective,
- toute partie E de T_B contenant \emptyset et stable par la fonction Node est égale à T_B .

Plus tard dans cette thèse, plusieurs raisonnements de preuves seront fondés sur la construction récursive d'arbres satisfaisant également les axiomes d'induction structurelle. Ces axioms sont en des généralisations des axiomes de Peano [Wik22] et permettent de généraliser le raisonnement par récurrence. En effet, dire de (E, c, f) qu'il satisfait les axiomes d'induction structurelle, c'est dire qu'il satisfait les propriétés suivantes :

- E est un ensemble ayant c pour élément,
- f est une fonction de E dans E (dans le cas particulier des arbres binaires l'ensemble de départ est $E \times E$),
- c n'est pas dans l'image de la fonction f ,
- f est injective,
- toute partie F de E contenant c et stable par la fonction f est égale à E .

Définition 1.58. Un **arbre plan** (fini) est un noeud $\text{Node}(f)$ avec pour enfants une forêt $f = [t_1, t_2, \dots, t_k]$, potentiellement vide, d'arbres plans. Les arbres de cette forêt, qui sont les enfants d'un même noeud, seront appelés des frères. Nous utiliserons également la représentation suivante :

$$t = \begin{array}{c} \text{---} \\ | \\ t_1 \quad t_2 \quad \dots \quad t_k \end{array} .$$

Ces deux familles d'arbres sont des arbres enracinés. Nous appelons la **racine** d'un arbre le noeud initial et nous le positionnons en haut comme nous le voyons sur les représentations. Nous appelons **nœuds internes** les nœuds ayant au moins un fils.

Définition 1.59. La **taille** d'un arbre ou d'une forêt est son nombre de nœuds.

Nous pouvons remarquer qu'avec cette définition, l'arbre vide n'est pas un arbre plan, donc il n'existe pas d'arbre plan de taille 0 contrairement aux arbres binaires. En effet, le cas de base dans la définition d'un arbre plan est un nœud avec pour enfant une forêt vide.

n	1	2	3	4	5	6	7	8	9	OEIS [SI20]
Arbres binaires _{n}	1	2	5	14	42	132	429	1430	4862	A000108
Arbres plans _{n}	1	1	2	5	14	42	132	429	1430	A000108

TABLE 1.5 – Nombre d'arbres binaires et plans de taille inférieure à 9.

Les arbres binaires et les arbres plans sont ce qu'on appelle des objets de Catalan car ils sont comptés par les nombres de Catalan. Il y a cependant un décalage. Toutefois, selon la **définition 1.58**, pour construire un arbre plan de taille n il suffit d'avoir une forêt de taille $n - 1$. Nous allons donc présenter une **bijection** entre les arbres binaires et les forêts d'arbres plans de même taille.

Exemple 1.60. Voici tous les arbres binaires de tailles 1, 2 et 3 :

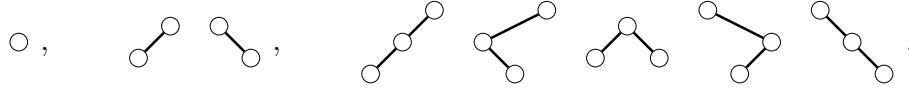


FIGURE 1.12 – Arbres binaires de tailles 1, 2 et 3.

Exemple 1.61. Voici tous les arbres plans de tailles 1, 2, 3 et 4 :

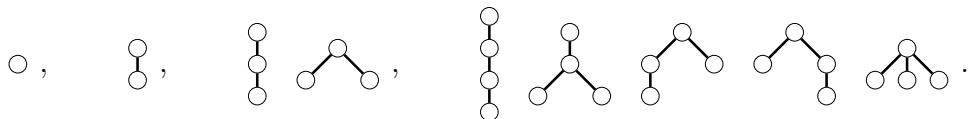


FIGURE 1.13 – Arbres plans de tailles 1, 2, 3 et 4.

Proposition 1.62. [FS09, I.5.3] La construction suivante décrit une bijection entre les arbres binaires et les forêts d’arbres plans de même taille :

Dans une forêt d’arbres plans, pour chaque nœud, le premier fils (resp. frère à sa droite) correspond au fils gauche (resp. droit) de l’arbre binaire associé.

Exemple 1.63. Pour les arbres binaires de tailles inférieures à 3 que l’on peut voir dans l’exemple 1.60, les forêts planes associées peuvent se voir dans l’exemple 1.61 en retirant la racine de l’arbre plan en suivant le même ordre. Par exemple, pour l’avant-dernier arbre binaire de taille 3 , la forêt plane associée est la suivante .

Ci-dessous un plus grand exemple où la bijection est montrée en deux étapes. Pour la première étape les noeuds ne sont pas déplacés, seules les arêtes sont différentes. La seconde étape consiste à “tourner” l’arbre pour qu’il soit dans le sens conventionnel.

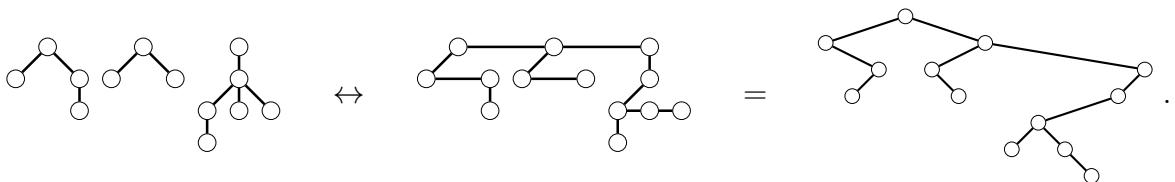


FIGURE 1.14 – Exemple de bijection entre une forêt plane et un arbre binaire.

1.2.2 Décorations des arbres

Dans la suite de cette section, nous nous intéresserons uniquement aux arbres plans. Nous décorerons ces arbres avec des étiquettes dans les nœuds et des couleurs sur les arêtes. Les notions de cette section se généralisent de façon évidente aux forêts planes.

Définition 1.64. Pour étiqueter les nœuds d'un arbre avec les éléments d'un ensemble X de décos, nous noterons $\text{Node}(x, f)$ l'arbre plan dont la racine est étiquetée par $x \in X$ et nous le représenterons comme ci-dessous :

$$t = \begin{array}{c} x \\ t_1 \quad t_2 \quad \dots \quad t_k \end{array} .$$

Exemple 1.65. Voici tous les arbres plan de tailles 1, 2 et 3 étiquetés par $X = \{1, 2\}$:

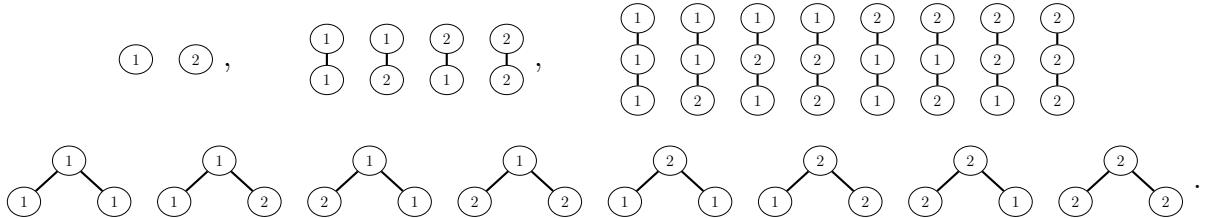


FIGURE 1.15 – Arbres plans décorés.

Si on étiquette les nœuds d'un arbre par des éléments tous différents il devient facile d'identifier un noeud grâce à son étiquette.

Définition 1.66. Soit $t = \text{Node}([t_1, \dots, t_k])$ un arbre plan. L'**ordre préfixe** sur les nœuds de t est tel que la racine de t est plus petite que les nœuds de t_1 qui sont eux-même plus petits que les nœuds de t_2 et ainsi de suite jusqu'à t_k .

Définition 1.67. Un arbre t de taille n est **numéroté selon l'ordre préfixe** si l'ensemble des décos est l'ensemble des nombres de 1 à n et que les étiquettes respectent l'ordre préfixe.

Exemple 1.68.

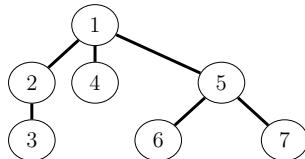


FIGURE 1.16 – Arbre plan numéroté selon l'ordre préfixe.

Définition 1.69. Pour **colorier** les arêtes d'un arbre plan avec des couleurs d'un ensemble C , nous noterons $\text{Node}_{[c_1, \dots, c_k]}([t_1, \dots, t_k])$ l'arbre plan dont l'arête arrivant à la racine depuis l'arbre t_i est coloriée par la couleur $c_i \in C$. Si toutes les couleurs sont égales à une couleur c , nous noterons simplement $\text{Node}_c([t_1, \dots, t_k])$.

On peut voir ces couleurs comme un code pour différencier les arbres, il ne s'agit pas d'une coloration de graphe au sens classique.

Nous rencontrerons plus tard, à plusieurs occasions, des **arbres étiquetés et coloriés**. Ces arbres nous permettront d'avoir une compréhension visuelle de propriétés fines d'objets combinatoires complexes.

1.2.3 Greffes d'arbres

Dans cette section nous numérotterons les arbres selon l'ordre préfixe. Nous nommerons également les noeuds par leurs étiquettes car elles sont toutes différentes.

Kontsevich a défini la notion d'angle d'un arbre enraciné dans [KS00]. Cette notion a été utilisée par Chapoton dans [Cha02] et par Foissy dans [Foi02b]. Leurs définitions utilisent la représentation d'un arbre. Nous redéfinissons la notion d'angle en accord avec nos notations.

Définition 1.70. Soit $t = \text{Node}(1, [t_1, \dots, t_k])$ un arbre plan (dont la racine est décorée par 1 avec la numérotation préfixe). On appelle **angle** de t un couple (n, α) où n est un noeud de t et α un espace entre ou autour des enfants du noeud n . On définit $\alpha = 0$ pour l'espace à gauche de la forêt, et $\alpha = i$ pour l'espace à droite de l'arbre i pour tout i . On note $\text{Angles}(t)$ l'ensemble des angles de t .

Exemple 1.71. Nous pouvons voir les angles en rouge pointillé sur cet exemple.

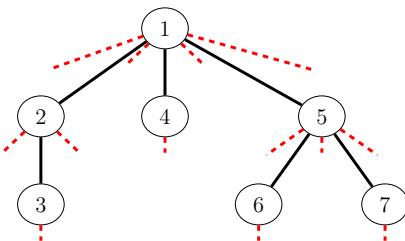


FIGURE 1.17 – Angles d'un arbre plan.

Les angles sont en fait tous les endroits où l'on peut ajouter un noeud dans l'arbre. Ces angles vont justement être les points d'accroche pour les greffes, mais pour définir les greffes, nous avons besoin de munir les angles d'une relation d'ordre totale.

Définition 1.72. L’ordre préfixe sur les angles de $\text{Angles}(t)$ est une relation d’ordre totale, de gauche à droite en considérant l’écriture sous la forme $t = \text{Node}(1, [(1, 0)t_1, (1, 1)t_2, \dots, (1, k-1)t_k, (1, k)])$. Nous avons alors que l’angle $(1, 0)$ est inférieur aux angles de t_1 qui sont eux-mêmes inférieurs à l’angle $(1, 1)$ et ainsi de suite jusqu’à $(1, k)$.

Exemple 1.73. Les angles sont numérotés sur cet exemple.

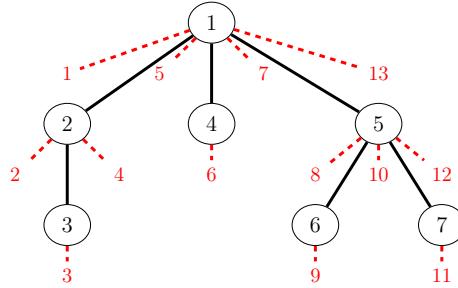


FIGURE 1.18 – Ordre sur les angles d’un arbre plan.

Définition 1.74. Soit $f = t_1 \dots t_k$ une forêt plane et t un arbre plan. Une **greffe** g de f sur t est une suite croissante de k angles de t et se note $g = [(n_1, \alpha_1), \dots, (n_m, \alpha_m)]$. Le **résultat d’une greffe** g de f sur t est l’arbre noté $R_g(f, t)$. Cet arbre est obtenu en greffant t_i sur le nœud n_i dans l’espace α_i , avec la condition suivante : si deux arbres t_i et t_{i+1} sont greffés au même nœud n_i et dans le même espace α_i alors t_i est greffé à gauche de t_{i+1} pour conserver le même ordre. On note $G_{f,t}$ l’ensemble des greffes possibles de f sur t .

Exemple 1.75. Pour plus de clarté, nous avons colorié les arbres dans cet exemple. Voici la greffe d’une forêt f , contenant 3 arbres, sur un arbre t en suivant la suite d’angles g :

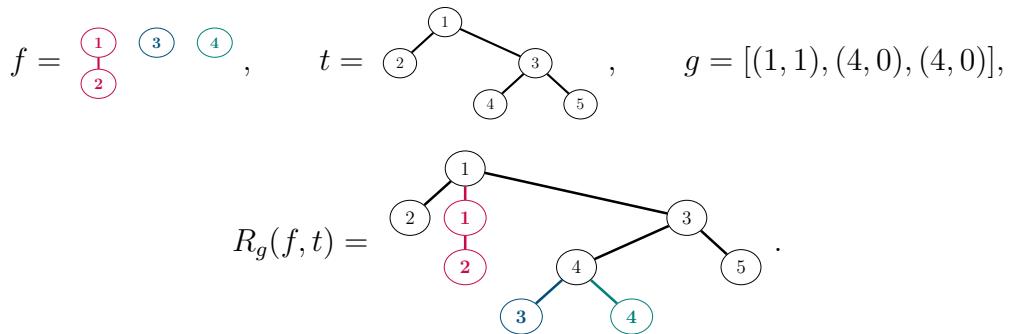


FIGURE 1.19 – Greffe d’une forêt plane sur un arbre plan.

Définition 1.76. Il est possible de généraliser la notion d’angle pour une forêt et donc de généraliser la notion de **greffe entre deux forêts** $f = t_1 \dots t_k$ et $f' = t'_1 \dots t'_{k'}$. Pour

cela, il suffit d'ajouter à f' un nœud racine n_0 imaginaire dont les enfants sont les racines des arbres t'_i de la forêt f' . Ainsi l'ensemble $\text{Angles}(f')$ correspond à l'union des angles des arbres t'_i de f' , auquel on adjoint les $k' + 1$ espaces (à gauche de t'_1 , entre les arbres et à droite de $t'_{k'}$) qui correspondent aux $k' + 1$ angles partant du nœud n_0 . L'ordre total sur les angles de $\text{Angles}(f')$ est immédiat, tout comme la notion de greffe entre deux forêts. On note $G_{f,f'}$ l'ensemble des greffes possibles de f sur f' .

Exemple 1.77. Pour plus de clarté, nous avons colorié les arbres dans cet exemple. Voici la greffe d'une forêt f , contenant 3 arbres, sur une autre forêt f' en suivant la suite d'angles g :

$$f = \begin{array}{c} 1 \\ 2 \end{array} \quad \begin{array}{c} 3 \\ \text{---} \\ 4 \end{array}, \quad f' = \begin{array}{c} 1 \\ \text{---} \\ 3 \\ \text{---} \\ 4 \end{array}, \quad g = [(0, 1), (3, 0), (0, 2)],$$

$$R_g(f, f') = \begin{array}{c} 1 \\ \text{---} \\ 2 \\ \text{---} \\ 3 \\ \text{---} \\ 4 \end{array}.$$

FIGURE 1.20 – Greffe d'une forêt plane sur une autre forêt plane.

Définition 1.78. Le **produit de greffe** \star entre deux forêts f et f' correspond à la somme formelle de toutes les greffes possibles de f sur f' ,

$$f \star f' := \sum_{g \in G_{f,f'}} R_g(f, f'). \quad (1.11)$$

Exemple 1.79. Pour cet exemple, en plus des couleurs, nous allons numérotter les nœuds de la deuxième forêt à la suite de la première forêt.

$$\begin{array}{c} 1 \\ 2 \end{array} \star \begin{array}{c} 4 \\ 5 \end{array} = \begin{array}{c} 1 \\ 2 \end{array} \begin{array}{c} 3 \\ \text{---} \\ 4 \\ \text{---} \\ 5 \end{array} + \begin{array}{c} 1 \\ 2 \\ \text{---} \\ 3 \\ \text{---} \\ 4 \\ \text{---} \\ 5 \end{array} + \begin{array}{c} 1 \\ 2 \\ \text{---} \\ 3 \\ \text{---} \\ 4 \\ \text{---} \\ 5 \end{array} + \begin{array}{c} 1 \\ 2 \\ \text{---} \\ 3 \\ \text{---} \\ 4 \\ \text{---} \\ 5 \end{array} + \begin{array}{c} 1 \\ 2 \\ \text{---} \\ 3 \\ \text{---} \\ 4 \\ \text{---} \\ 5 \end{array} + \begin{array}{c} 1 \\ 2 \\ \text{---} \\ 3 \\ \text{---} \\ 4 \\ \text{---} \\ 5 \end{array} +$$

$$\begin{array}{c} 1 \\ 2 \\ \text{---} \\ 4 \\ \text{---} \\ 3 \\ \text{---} \\ 5 \end{array} + \begin{array}{c} 1 \\ 2 \\ \text{---} \\ 4 \\ \text{---} \\ 5 \\ \text{---} \\ 3 \end{array} + \begin{array}{c} 1 \\ 2 \\ \text{---} \\ 4 \\ \text{---} \\ 5 \\ \text{---} \\ 3 \end{array} + \begin{array}{c} 1 \\ 2 \\ \text{---} \\ 4 \\ \text{---} \\ 5 \\ \text{---} \\ 3 \end{array} + \begin{array}{c} 1 \\ 2 \\ \text{---} \\ 4 \\ \text{---} \\ 5 \\ \text{---} \\ 3 \end{array} + \begin{array}{c} 1 \\ 2 \\ \text{---} \\ 4 \\ \text{---} \\ 5 \\ \text{---} \\ 3 \end{array} +$$

$$\begin{array}{c} 4 \\ \text{---} \\ 1 \\ \text{---} \\ 2 \\ \text{---} \\ 5 \\ \text{---} \\ 3 \end{array} + \begin{array}{c} 4 \\ \text{---} \\ 1 \\ \text{---} \\ 2 \\ \text{---} \\ 5 \\ \text{---} \\ 3 \end{array} + \begin{array}{c} 4 \\ \text{---} \\ 1 \\ \text{---} \\ 2 \\ \text{---} \\ 5 \\ \text{---} \\ 3 \end{array} + \begin{array}{c} 4 \\ \text{---} \\ 1 \\ \text{---} \\ 2 \\ \text{---} \\ 5 \\ \text{---} \\ 3 \end{array} + \begin{array}{c} 4 \\ \text{---} \\ 1 \\ \text{---} \\ 2 \\ \text{---} \\ 5 \\ \text{---} \\ 3 \end{array} + \begin{array}{c} 1 \\ 2 \\ \text{---} \\ 1 \\ \text{---} \\ 2 \\ \text{---} \\ 3 \end{array}.$$

FIGURE 1.21 – Produit de greffe entre deux forêts planes.

Il est également possible de définir des demi-produits de greffe sur les forêts comme dans [Foi02b, Théorème 29]. Nous en donnons les expressions.

Définition 1.80. Les deux **demi-produits de greffes** entre deux forêts $f = [t_1, \dots, t_k]$ et $f' = [t'_1, \dots, t'_{k'}]$ se définissent par la somme formelle de toutes les greffes possibles de f sur f' de façon à ce que le dernier arbre soit égal (resp. différent) au dernier arbre de f .

$$f \prec f' := \sum_{\substack{g \in G_{f,f'} \\ g_k = (0, k')}} R_g(f, f'), \quad (1.12)$$

$$f \succ f' := \sum_{\substack{g \in G_{f,f'} \\ g_k \neq (0, k')}} R_g(f, f'). \quad (1.13)$$

Exemple 1.81. Pour cet exemple, nous reprenons les même arbres que l'[exemple 1.79](#) et nous donnerons uniquement le résultat du produit de greffe gauche, le droit pouvant se deviner comme la somme des autres greffes possibles.

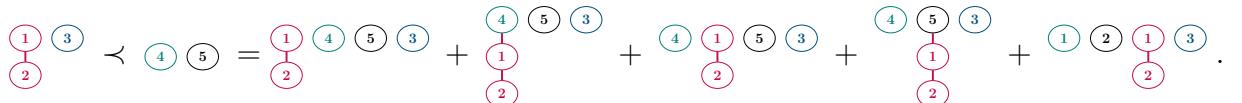


FIGURE 1.22 – Demi-produit de greffe entre deux forêts planes.

1.3 Grammaires et séries génératrices

Nous avons défini précédemment plusieurs classes combinatoires : permutations, permutations irréductibles, mots tassés, mots tassés irréductibles, arbres binaires, arbres plans, forêts d’arbres plans, arbres décorés. Nous avons donné des tables montrant les cardinalités de la plupart de ces objets par taille ([tables 1.1, 1.2, 1.3, 1.4 et 1.5](#)). Flajolet et Sedgewick font le lien entre les grammaires symboliques et les séries génératrices [[FS09](#), Figures I.18 et II.18]. Une grammaire est un formalisme permettant de définir une syntaxe et donc un langage formel, c’est-à-dire un ensemble de mots admissibles sur un alphabet donné. Une grammaire se caractérise par un ensemble fini de symboles, terminaux ou non, et un ensemble de règles de productions.

Définition 1.82. À toute classe combinatoire \mathcal{C} , on peut associer une série formelle, appelée **série génératrice** :

$$\mathcal{G}_{\mathcal{C}}(z) = \sum_{n \geq 0} \mathfrak{c}_n z^n, \quad (1.14)$$

où \mathfrak{c}_n est le nombre d’objets de taille n .

Il existe également un autre type de séries génératrices, les séries génératrices exponentielles, où le coefficient \mathfrak{c}_n est divisé par $n!$. Ces séries génératrices exponentielles sont pratiques pour parler des objets à permutations près.

Dans cette section, nous allons faire le lien entre les grammaires définissant tous les objets combinatoires que nous avons définis jusqu’ici avec leurs séries génératrices.

Arbres binaires : de la [définition 1.57](#) récursive sur les arbres binaires, on en déduit aisément la grammaire suivante :

$$T_B \rightarrow \emptyset \mid \text{Node}(T_B, T_B). \quad (1.15)$$

Cette grammaire nous permet, à son tour, de déduire aisément l'équation suivante :

$$\mathcal{T}_B(z) = 1 + z\mathcal{T}_B(z)^2. \quad (1.16)$$

En résolvant l'équation, on retrouve bien la série des nombres de Catalan ([OEIS \[SI20\] A000108 table 1.5](#)) :

$$\mathcal{T}_B(z) = \frac{1 - \sqrt{1 - 4z}}{2z} = 1 + z + 2z^2 + 5z^3 + 14z^4 + 42z^5 + 132z^6 + \dots \quad (1.17)$$

Forêts d'arbres plans : la [définition 1.58](#) est une récursion croisée entre les arbres et les forêts, il en découle la grammaire suivante :

$$F_P \rightarrow [] \mid T_P \cdot F_P, \quad (1.18)$$

$$T_P \rightarrow \text{Node}(F_P). \quad (1.19)$$

De cette grammaire découlent les équations suivantes :

$$\mathcal{F}_P(z) = 1 + \mathcal{T}_P(z)\mathcal{F}_P(z), \quad (1.20)$$

$$\mathcal{T}_P(z) = z\mathcal{F}_P(z). \quad (1.21)$$

En résolvant ce système d'équations on retombe naturellement sur la série des nombres de Catalan, nous avons bien $\mathcal{F}_P(z) = \mathcal{T}_B(z)$ (voir la [proposition 1.62](#)).

Forêts planes décorées : pour les décorer par un ensemble $X = \{x_1, x_2, \dots, x_n\}$, il suffit de modifier légèrement la seconde règle et d'ajouter une règle de décoration :

$$F_P^X \rightarrow [] \mid T_P^X \cdot F_P^X, \quad (1.22)$$

$$T_P^X \rightarrow \text{Node}(X, F_P^X), \quad (1.23)$$

$$X \rightarrow x_1 \mid x_2 \mid \dots \mid x_n. \quad (1.24)$$

En résolvant ce système on a :

$$\mathcal{F}_P^X(z) = \frac{1 - \sqrt{1 - 4nz}}{2nz} = 1 + nz + 2n^2z^2 + 5n^3z^3 + 14n^4z^4 + 42n^5z^5 + \dots \quad (1.25)$$

Il est également possible de décorer les forêts avec un ensemble gradué, qui a donc sa propre série génératrice. On obtient alors l'équation suivante :

$$\mathcal{F}_P^X(z) = 1 + \mathcal{X}(z)\mathcal{F}_P^{X^2}(z). \quad (1.26)$$

Nous abordons avec plus de détails les décorations graduées dans le [chapitre 7](#) lorsque nous étudierons les opérades.

Permutations : pour les permutations, nous aurions naturellement envie de travailler avec les séries exponentielles, mais pour les besoins que nous avons, nous allons rester avec les séries ordinaires.

Il est possible de décrire une permutation de taille $n > 0$ comme le standardisé de la concaténation d'une permutation de taille $n - 1$ avec une lettre entre 0 et $n - 1$.

Autrement dit, pour construire une permutation σ de taille $n > 0$, on choisit une permutation σ' de taille $n - 1$ à laquelle on ajoute une lettre i entre 1 et n , puis on augmente de 1 la valeur de toutes les lettres supérieures ou égales à i dans σ' .

Cette description se traduit par la grammaire suivante où les deuxième et troisième règles représentent en fait une règle pour chaque n :

$$P_0 \rightarrow \epsilon, \tag{1.27}$$

$$P_{n>0} \rightarrow \text{std}(P_{n-1} \cdot X_n), \tag{1.28}$$

$$X_n \rightarrow 1 \mid 2 \mid \dots \mid n. \tag{1.29}$$

Contrairement à précédemment où nous avions des équations sur les séries génératrices, nous avons ici un système d'équations sur les coefficients de la série :

$$\mathcal{P}_0 = 1, \tag{1.30}$$

$$\mathcal{P}_{n>0} = n * \mathcal{P}_{n-1}. \tag{1.31}$$

On retombe bien sur la série des factorielles :

$$\mathcal{P}(z) = \sum_{n \geq 0} n! z^n. \tag{1.32}$$

Permutations irréductibles : prenons un chemin un peu différent, revenons sur la description d'une permutation (dont on connaît déjà la série génératrice) comme la concaténation décalée d'une permutation irréductible avec une permutation :

$$P \rightarrow \epsilon \mid I/P, \tag{1.33}$$

$$I \rightarrow \{\text{une permutation irréductible}\}. \tag{1.34}$$

Nous en déduisons l'équation suivante :

$$\mathcal{P}(z) = 1 + \mathcal{I}(z)\mathcal{P}(z), \tag{1.35}$$

puis les deux équations équivalentes suivantes :

$$\mathcal{P}(z) = \frac{1}{1 - \mathcal{I}(z)}, \tag{1.36}$$

$$\mathcal{I}(z) = \frac{\mathcal{P}(z) - 1}{\mathcal{P}(z)}. \tag{1.37}$$

On retrouve bien la série OEIS [SI20] A003319 :

$$\mathcal{I}(z) = 1 - \frac{1}{\sum n! z^n} = z + z^2 + 3z^3 + 13z^4 + 71z^5 + \dots \quad (1.38)$$

Posons P^* comme l'ensemble des permutations non vides, l'expression de $\mathcal{I}(z)$ en fonction de $\mathcal{P}^*(z)$ est sous une forme que l'on peut généraliser (voir [équation \(1.43\)](#)). Nous avons alors avec le même raisonnement que précédemment les deux équations équivalentes suivantes :

$$\mathcal{P}^*(z) = \frac{\mathcal{I}(z)}{1 - \mathcal{I}(z)}, \quad (1.39)$$

$$\mathcal{I}(z) = \frac{\mathcal{P}^*(z)}{\mathcal{P}^*(z) + 1}. \quad (1.40)$$

Nous verrons plus tard un sous-ensemble de permutations T dont la série génératrice vérifie la relation suivante :

$$\mathcal{T}(z) = \frac{\mathcal{P}^*(z)}{(\mathcal{P}^*(z) + 1)^2}. \quad (1.41)$$

Cette relation est équivalente à :

$$\mathcal{P}^*(z) = \frac{1 - \sqrt{1 - 4\mathcal{T}(z)}}{2\mathcal{T}(z)}. \quad (1.42)$$

On reconnaît ici la formule de Catalan. Nous pouvons alors dire que les permutations non vides peuvent être décrites comme les arbres binaires (ou forêts planes) étiquetés par des éléments de T .

Il est possible de généraliser cette remarque, soit A et B deux ensembles tels que leurs séries génératrices vérifient la relation suivante :

$$\mathcal{A}(z) = \frac{\mathcal{B}(z)}{(\mathcal{B}(z) + 1)^n}. \quad (1.43)$$

Nous pouvons alors dire que les éléments de B peuvent être décrits comme les arbres n -aires (arbres dont tous les noeuds internes ont exactement n enfants) étiquetés par des éléments de A .

Mots tassés : encore une fois, il serait plus simple d'obtenir la série génératrice des mots tassés avec les séries exponentielles, mais pour les besoins de notre étude, nous utiliserons uniquement les séries ordinaires.

Il est possible de décrire un mot tassé de taille $n > 0$ d'une façon semblable aux permutations. Cependant en ajoutant une lettre à la fin, nous ferions intervenir une double graduation, par la taille et la lettre maximale. Nous allons plutôt décrire un mot tassé par

l'ajout de maximums. Un mot tassé de taille $n > 0$ est, pour tout $k < n$, un mot tassé de taille $n - k$ auquel on a ajouté k lettres (qui sont des nouveaux maximums) dans le mot. Avec cette description, comme il y a $\binom{n}{k}$ possibilité d'insérer k lettres dans le mot, nous avons le système d'équation suivant sur les coefficients de la série :

$$\mathcal{PW}_0 = 1, \quad (1.44)$$

$$\mathcal{PW}_{n>0} = \sum_{k=1}^n \binom{n}{k} \mathcal{PW}_{n-k}. \quad (1.45)$$

En utilisant cette formule récursivement on retombe bien sur la série des mots tassés (OEIS [SI20] A000670) :

$$\mathcal{PW}(z) = 1 + z + 3z^2 + 13z^3 + 75z^4 + 541z^5 + \dots \quad (1.46)$$

Mots tassés irréductibles : enfin, il est possible d'obtenir les mêmes équations qu'avec les permutations irréductibles en décomposant les mots tassés en mots tassés irréductibles selon les descentes globales (voir lemme 1.45) :

$$\mathcal{PW}(z) = \frac{1}{1 - \mathcal{I}_{\mathbf{PW}}(z)}, \quad (1.47)$$

$$\mathcal{I}_{\mathbf{PW}}(z) = \frac{\mathcal{PW}(z) - 1}{\mathcal{PW}(z)}. \quad (1.48)$$

On retrouve bien la série OEIS [SI20] A095989 :

$$\mathcal{I}_{\mathbf{PW}}(z) = 1 - \frac{1}{\sum n!z^n} = z + 2z^2 + 8z^3 + 48z^4 + 368z^5 + \dots \quad (1.49)$$

Chapitre 2

Ordres partiels

Les **ensembles partiellement ordonnés** sont couramment appelés **posets**, de l'anglais *Partially Ordered Set*. Les études d'ordres partiels sur des objets combinatoires sont très présentes dans la littérature et permettent souvent d'avoir de nouvelles visions sur ces objets. Les posets ont même été étudiés en tant qu'objets combinatoires et des ordres partiels sur les ordres partiels ont été étudiés dans [CPP17].

Une exposition claire et condensée des définitions les plus courantes sur les posets est donnée dans [Pon13]. Nous reprenons ici avec l'accord de l'autrice une partie de ces définitions et exemples.

Quelques exemples parmi les plus classiques en combinatoire sont l'ordre d'inclusion ensembliste, l'ordre faible sur les permutations ou encore le treillis de Tamari sur les arbres binaires.

Nous présentons dans ce chapitre les deux ordres faibles, droit et gauche, que nous généralisons ensuite aux mots tassés. Nous présentons également l'ordre de raffinement sur les mots tassés dont les composantes sont en bijection avec l'ordre d'inclusion d'ensemble.

Les définitions de ce chapitre nous serviront lors de la construction de certaines bases de **WQSym** dans la [section 3.2.2](#).

2.1 Définitions

Définition 2.1. Un **ordre partiel** sur un ensemble P (*poset*) est une relation d'ordre \leq vérifiant les conditions suivantes :

- réflexivité : $\forall x \in P, x \leq x$,
- transitivité : $\forall x, y, z \in P$, si $x \leq y$ et $y \leq z$ alors $x \leq z$,
- antisymétrie : $\forall x, y \in P$, si $x \leq y$ et $y \leq x$ alors $x = y$.

Si de plus tous les éléments de P sont comparables on dit alors que l'ordre est **total** ou **linéaire**. En d'autres termes, si pour tous $x, y \in P$ nous avons soit $x \leq y$ soit $y \leq x$, alors l'ordre est total.

Dans cette thèse nous n'étudierons que des ordres partiels sur des **ensembles finis**.

Définition 2.2. On dit que y **couvre** x et on écrit $x \lessdot y$ si on a $x < y$ et qu'il n'existe pas de $z \in P$ tel que $x < z < y$. On appelle l'ensemble de ces relations les **relations de couverture**.

Définition 2.3. On dit qu'un élément $x \in P$ est **minimal** (resp. **maximal**) s'il n'existe pas d'élément $y \in P$, $y \neq x$, tel que $y \leq x$ (resp. $y \geq x$).

Les relations de couverture suffisent à définir le poset (les autres relations étant obtenues par transitivité).

Définition 2.4. Un **diagramme de Hasse** est la représentation standard d'un poset dans lequel les éléments sont reliés selon leurs relations de couverture.

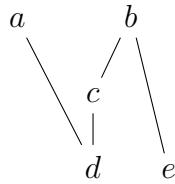


FIGURE 2.1 – Exemple de diagramme de Hasse d'un poset quelconque.

Les relations de couverture sont $d \lessdot c \lessdot b$, $d \lessdot a$ et $e \lessdot b$. Les éléments d et e sont minimaux et les éléments a et b sont maximaux. L'ordre est partiel, en effet a et b ne sont pas comparables.

Définition 2.5. Une **chaîne** d'un poset P est un ensemble d'éléments $\{x_1, \dots, x_n\}$ tel que :

$$x_1 \leq x_2 \leq \dots \leq x_n. \quad (2.1)$$

Si pour tout $i \in [1 \dots n]$, on a $x_i \lessdot x_{i+1}$, alors la chaîne est dite **saturée**.

Par exemple, $d \lessdot c \lessdot b$ est une chaîne saturée du poset donné en figure 2.1.

Définition 2.6. On dit qu'un poset est **gradué** s'il existe une application Φ bien définie telle que :

1. $\Phi(x) = 0$ si x est minimal,
2. $\Phi(y) = \Phi(x) + 1$ si $x \lessdot y$.

De façon équivalente, un poset est **gradué** si la longueur d'une chaîne saturée entre un élément $y \in P$ et un élément minimal x du poset ne dépend ni de x , ni de la chaîne choisie. Ainsi, le poset donné figure 2.1 n'est pas gradué car $d \lessdot c \lessdot b$ et $e \lessdot b$ sont deux chaînes saturées de longueurs différentes d'éléments minimaux vers b .

Les définitions suivantes donnent des constructions de posets à partir d'un poset donné. Des exemples de toutes les constructions (sous-poset, intervalle, etc.) sont illustrés dans la figure 2.2.

Définition 2.7. Un poset P' est un **sous-poset** de P si en tant qu'ensemble, $P' \subset P$ et si la relation d'ordre de P' est la même que celle de P restreinte aux éléments de P' .

Si pour tout $x, y \in P'$, on a que $x \leq y$ implique que $\forall z \in P$ avec $x \leq z \leq y$ alors $z \in P'$, on dit que P' est **clos par intervalle**. Si de plus, P' a un unique élément minimal m et un unique élément maximal M , on dit que P' est l'**intervalle** $[m, M]$ de P .

Définition 2.8. Soit P un poset et $x \in P$, l'**idéal inférieur** (resp. **supérieur**) de P engendré par x est l'ensemble des éléments $y \leq x$ (resp. $y \geq x$).

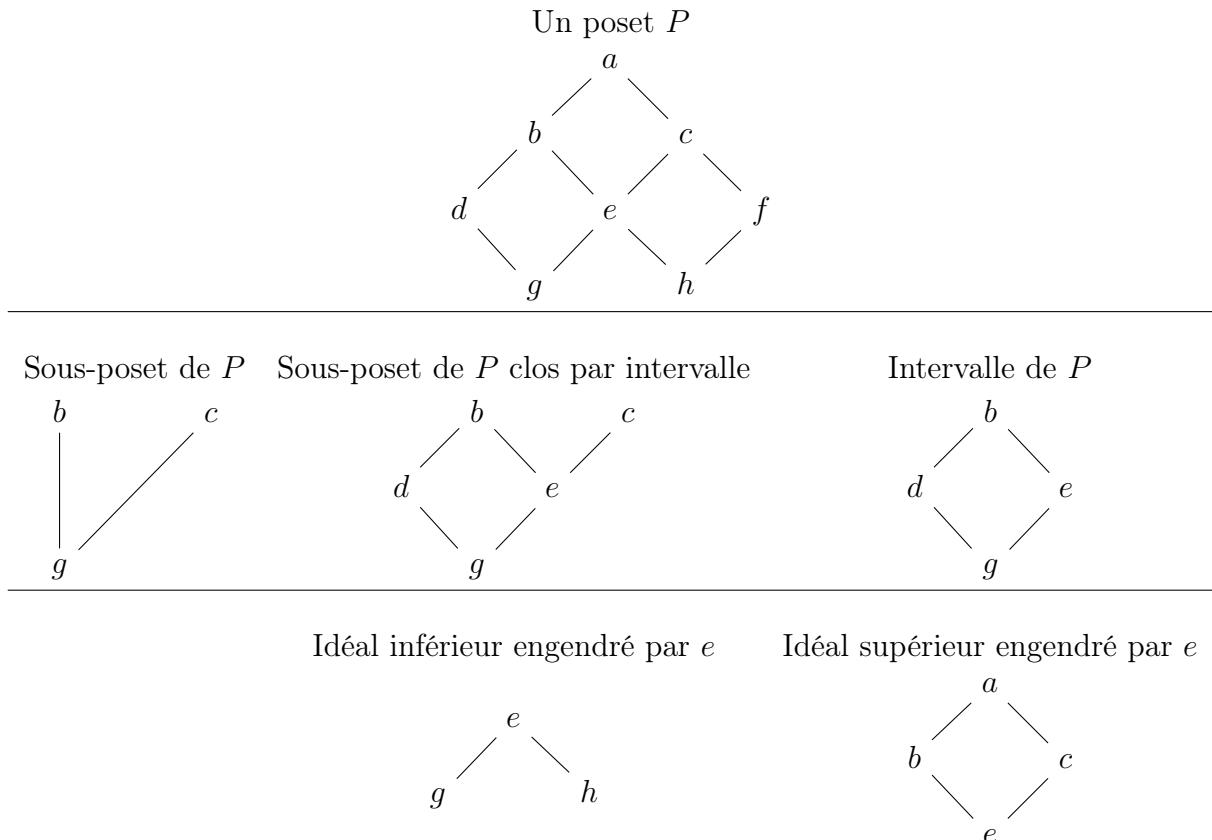


FIGURE 2.2 – Exemples de sous-posets.

2.2 Ordres faibles sur les permutations

2.2.1 Groupe symétrique

Les résultats que nous énonçons ici sont tous bien connus. Pour plus de détails nos références sont les suivantes [LS96, Pon13].

Nous avons décrit les permutations, dans la [définition 1.14](#), comme des mots sur l'alphabet des entiers naturels. On peut aussi les interpréter comme des bijections de $\{1, \dots, n\}$

vers $\{1, \dots, n\}$. En effet, chaque position est envoyée sur une valeur différente. Par exemple, la permutation $\sigma = 2314$ est l'application qui envoie 1 sur 2, 2 sur 3, 3 sur 1 et 4 sur 4.

Définition 2.9. L'ensemble des permutations de taille n forme un groupe non commutatif dont la loi est la composition des applications. On appelle ce groupe le **groupe symétrique**. L'élément neutre est la permutation identité, c'est-à-dire le mot $123\dots n$ et l'**inverse** correspond à l'échange des positions et des valeurs.

Exemple 2.10. Par exemple, si $\sigma = 2314$ et $\mu = 4213$, on a :

$$\sigma \circ \mu = 4321, \tag{2.2}$$

$$\mu \circ \sigma = 2143, \tag{2.3}$$

$$\sigma^{-1} = 3124. \tag{2.4}$$

Définition 2.11. Une **transposition** est une permutation où seuls deux points ne sont pas fixes, on dit qu'elle "échange" deux valeurs. Une **transposition simple** est une transposition où les valeurs échangées sont consécutives. On note $(i \ j)$ la transposition des points i et j . De plus, on note $s_i := (i \ i')$, avec $i' = i + 1$, la $i^{\text{ème}}$ transposition simple.

Exemple 2.12. Par exemple, 153426 est la transposition qui échange 2 et 5 et on la note $(2 \ 5)$.

Lorsqu'on multiplie par la **droite** une permutation σ par une transposition $(i \ j)$, cela revient à échanger les lettres en **positions** i et j dans σ . Si l'on multiplie par la **gauche**, alors on échange les **valeurs** i et j . Par exemple :

$$342615 \circ (2 \ 5) = 312645 \tag{2.5}$$

$$(2 \ 5) \circ 342615 = 345612. \tag{2.6}$$

Définition 2.13. L'ensemble des **inversions gauche** d'une permutation σ se note $\text{Inv}_L(\sigma)$ et se définit par l'ensemble des paires de valeurs placées dans l'ordre décroissant dans σ :

$$\text{Inv}_L(\sigma) := \{(\sigma_i, \sigma_j) \mid \sigma_i < \sigma_j \quad \text{et} \quad i > j\}. \tag{2.7}$$

De façon similaire, l'ensemble des **inversions droite** se note $\text{Inv}_R(\sigma)$ et se définit par l'ensemble des paires de positions dont les valeurs sont dans l'ordre décroissant dans σ :

$$\text{Inv}_R(\sigma) := \{(i, j) \mid i < j \quad \text{et} \quad \sigma_i > \sigma_j\}. \tag{2.8}$$

Exemple 2.14. Soit la permutation suivante $\sigma = 1423$.

L'ensemble de ces inversions gauches est $\text{Inv}_L(\sigma) = \{(2, 4), (3, 4)\}$, en effet, la valeur 2 est après la valeur 4 dans σ , de même la valeur 3 est après la valeur 4 dans σ .

L'ensemble de ces inversions droites est $\text{Inv}_R(\sigma) = \{(2, 3), (2, 4)\}$, en effet, en position 2 il y a une lettre plus grande qu'en position 3, de même pour les positions 3 et 4.

Remarque 2.15. Les inversions gauches (resp. droites) correspondent à des multiplications à gauche (resp. droite) par une transposition. Autrement dit, si σ est une permutation et (i, j) une inversion droite de σ , alors la permutation obtenue par $\sigma.(i\ j)$ a exactement une inversion droite de moins que σ .

Remarque 2.16. Les inversions droites et gauches peuvent se définir de la même façon sur des mots quelconques. De plus, le standardisé d'un mot w est l'unique permutation possédant le même ensemble d'inversions droites.

$$\sigma = \text{std}(w) \iff \text{Inv}_R(\sigma) = \text{Inv}_R(w).$$

2.2.2 Ordres faibles droit et gauche

Les ordres faibles sur les permutations constituent un exemple classique d'ordre partiel. Ce qui nous intéressera le plus seront les généralisations de ces ordres sur les mots tassés. Ces ordres sont appelés droit et gauche car leurs relations de couverture sont équivalentes à une multiplication à droite ou à gauche par une transposition simple. Néanmoins nous utiliserons les définitions en terme d'inclusion d'ensembles d'inversions, cette définition étant plus appropriée pour certaines preuves.

Définition 2.17. L'**ordre faible droit** \leq_R (resp. l'**ordre faible gauche** \leq_L) sur les permutations se définit pour toutes permutations σ et μ par l'inclusion des inversions gauches (resp. droites) :

$$\sigma \leq_R \mu \iff \text{Inv}_L(\sigma) \subseteq \text{Inv}_L(\mu), \quad (2.9)$$

$$(\text{resp. } \sigma \leq_L \mu \iff \text{Inv}_R(\sigma) \subseteq \text{Inv}_R(\mu)). \quad (2.10)$$

Nous pouvons remarquer que pour les définitions par inclusion d'ensemble, il y a un échange des côtés gauche et droit.

Les diagrammes de Hasse des ordres faibles droits et gauches pour les tailles 3 et 4 sont donnés [figures 2.3 et 2.4](#).

Les relations de couverture se déduisent facilement de la définition. Une permutation σ est couverte par une permutation μ (c'est-à-dire que μ est son successeur direct) si μ a les mêmes inversions que σ plus une. De façon équivalente, $\sigma <_R \mu$ (resp. $\sigma <_L \mu$) s'il existe une transposition simple s_i telle que $\mu = \sigma.s_i$ (resp. $\mu = s_i.\sigma$).

Ces deux ordres sont gradués par le nombre d'inversions. Ils sont également isomorphes l'un à l'autre par passage à l'inverse. Plongé dans \mathbb{R}^{n-1} , leur diagramme de Hasse est un polytope connu, le **permutoèdre**.

Enfin Loday-Ronco ont prouvé que les produits de mélanges se lisent sur ce poset.

Proposition 2.18. [[LR02](#), Théorème 4.1] Le produit de mélange décalé (resp. mélange des valeurs) de deux permutations σ et μ correspond à la somme des éléments d'un intervalle

de l'ordre faible droit (resp. gauche) selon les formules suivantes :

$$\sigma \overline{\sqcup} \mu = \sum_{\nu \in [\sigma \setminus \mu, \mu / \sigma]_R} \nu, \quad (2.11)$$

$$\sigma \underline{\sqcup} \mu = \sum_{\nu \in [\sigma \setminus \mu, \sigma / \mu]_L} \nu. \quad (2.12)$$

Dans le même article, Loday-Ronco [LR02, Proposition 4.6] prouvent que les demi-produits de mélange se lisent également comme des intervalles sur ce poset. Nous généraliserons ce résultat sur les mots tassés dans la [proposition 4.81](#).

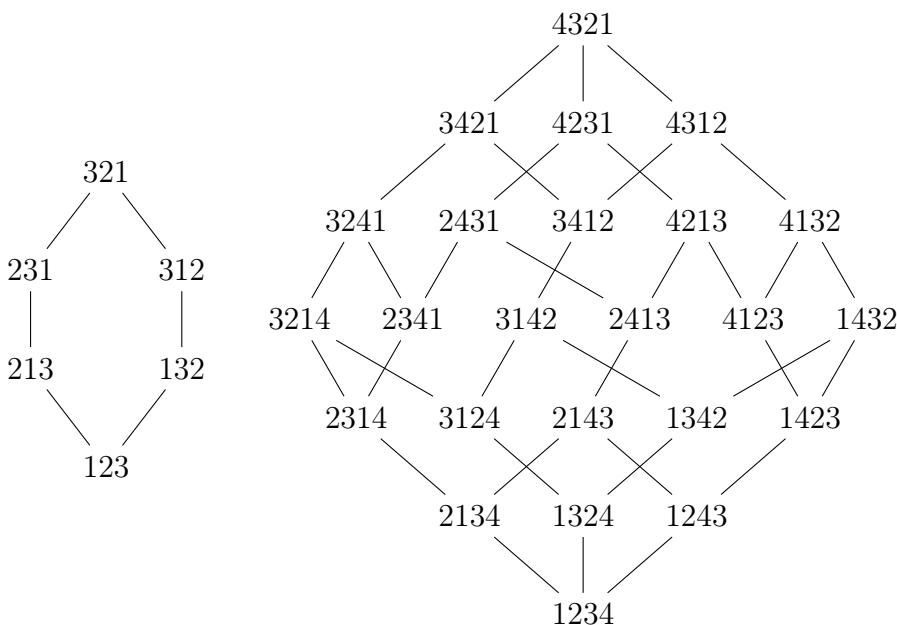


FIGURE 2.3 – Ordre faible droit \leq_R sur les permutations pour les tailles 3 et 4.

2.3 Ordres sur les mots tassés

Dans cette section nous allons nous intéresser à 4 ordres partiels sur les mots tassés. Pour 3 de ces ordres, il s'agit de généralisations des ordres faibles sur les permutations. Ce sont des généralisations dans le sens où la restriction aux permutations décrit bien l'ordre faible et que c'est un sous-poset. Plus de détails et de propriétés sur ces ordres sont présentés dans [Var19, KLP01].

Les produits de mélange sur les mots tassés sont visibles sur des intervalles de ces 3 ordres. Nous avons en effet bien défini 3 types de produits de mélange sur les mots tassés : mélange des positions, mélange des valeurs et mélange des valeurs sans chevauchement. Nous nommerons respectivement ces trois ordres : ordre faible droit, ordre faible gauche avec chevauchement et ordre faible gauche.

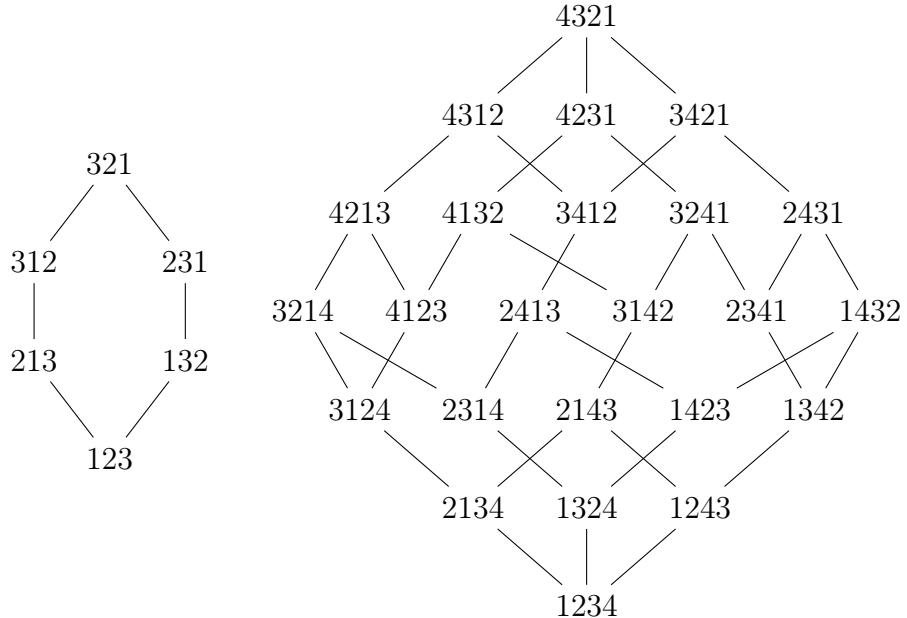


FIGURE 2.4 – Ordre faible gauche \leq_L sur les permutations pour les tailles 3 et 4.

Enfin, le quatrième et dernier ordre que nous présenterons, l’ordre de raffinement, permettra d’encoder le changement de base vers le produit de mélange sans chevauchement (équation (3.30)).

2.3.1 Généralisation de l’ordre faible droit

Définition 2.19. L’ordre faible droit \leq_R sur les mots tassés se définit pour tous mots tassés u et v par l’égalité des compositions et l’inclusion des inversions gauches :

$$u \leq_R v \iff \text{comp}(u) = \text{comp}(v) \quad \text{et} \quad \text{Inv}_L(u) \subseteq \text{Inv}_L(v). \quad (2.13)$$

Le diagramme de Hasse de l’ordre faible droit pour la taille 3 est donné figure 2.5. Une composante connexe de la taille 5 est donnée figure 2.6.

2.3.2 Deux généralisations de l’ordre faible gauche

Nous allons présenter deux généralisations de l’ordre faible gauche. Pour les définir, nous avons besoin de l’ensemble des paires de positions ayant la même valeur dans un mot tassé. Nous nommons ces paires des demi-inversions.

Définition 2.20. [KLN01] L’ensemble des **demi-inversions** d’un mot tassé w se note $\text{Inv}^*(w)$ et se définit par l’ensemble des paires de positions dont les valeurs sont égales dans w :

$$\text{Inv}^*(w) := \{(i, j) \mid i < j \quad \text{et} \quad w_i = w_j\}. \quad (2.14)$$

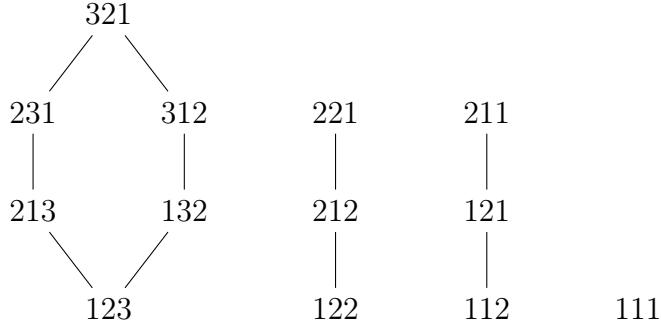


FIGURE 2.5 – Ordre faible droit \leq_R sur les mots tassés de taille 3.

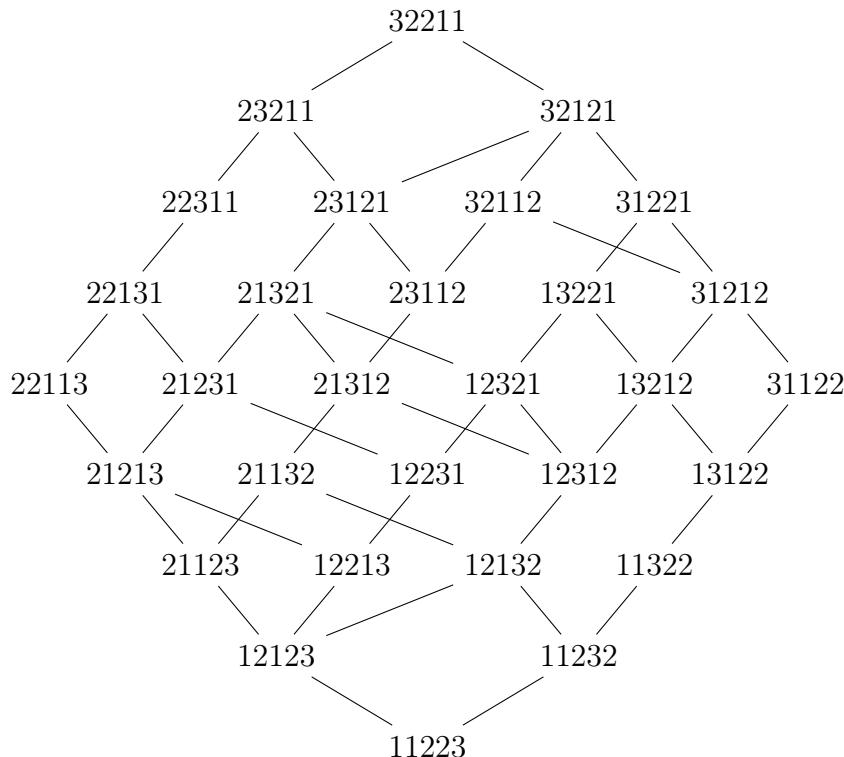


FIGURE 2.6 – Ordre faible droit \leq_R sur les mots tassés de composition 221.

Définition 2.21. [KLNP01] L’ensemble des **pseudo-inversions droites** d’un mot tassé w se note $\text{Inv}_R^*(w)$ et se définit par l’union des inversions droites et des demi-inversions coefficientées par $\frac{1}{2}$:

$$\text{Inv}_R^*(w) := \text{Inv}_R(w) \cup \left\{ \frac{1}{2}(i, j) \mid (i, j) \in \text{Inv}^* \right\}. \quad (2.15)$$

Définition 2.22. [KLNP01] L’ordre faible gauche avec chevauchement (ou ordre du pseudo-permutoèdre) \leq_L^* sur les mots tassés se définit pour tous mots tassés u et v

par l'inclusion des pseudo-inversions droites :

$$u \leq_L^* v \iff \text{Inv}_R^*(u) \subseteq \text{Inv}_R^*(v). \quad (2.16)$$

Dans cette définition, et comme nous pouvons le voir dans l'exemple suivant, la demi-inversion $\frac{1}{2}(i, j)$ est incluse dans l'inversion (i, j) , d'où le terme de demi-inversion.

Exemple 2.23. Soit les mots tassés suivants $u = 121$ et $v = 221$. Leurs ensembles de pseudo-inversions sont :

$$\text{Inv}_R^*(u) = \left\{ \frac{1}{2}(1, 3), (2, 3) \right\} \quad \text{et} \quad \text{Inv}_R^*(v) = \left\{ \frac{1}{2}(1, 2), (1, 3), (2, 3) \right\}.$$

Nous avons bien $u \leq_L^* v$ car l'inversion $(2, 3)$ est une inversion de u et de v et que la demi-inversion $\frac{1}{2}(1, 3)$ de u est incluse dans l'inversion $(1, 3)$ de v .

Cet ordre avec chevauchement n'est pas gradué et correspond à un ordre sur les faces du permutoèdre.

Le diagramme de Hasse de l'ordre faible gauche avec chevauchement pour la taille 3 est donné [figure 2.7](#).

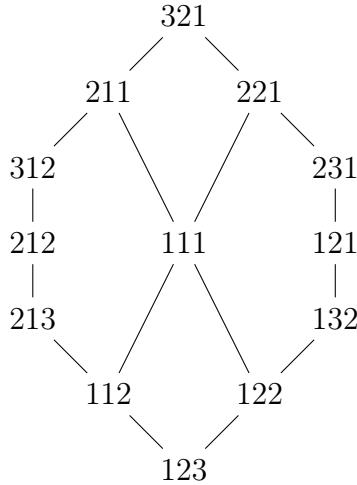


FIGURE 2.7 – Ordre faible gauche avec chevauchement \leq_L^* sur les mots tassés de taille 3.

Enfin, nous pouvons définir une deuxième généralisation de l'ordre faible gauche qui contient plusieurs composantes connexes comme la généralisation de l'ordre faible droit.

Définition 2.24. L'**ordre faible gauche** (ou **ordre faible gauche sans chevauchement**) \leq_L sur les mots tassés se définit pour tous mots tassés u et v par l'égalité des demi-inversions et l'inclusion des inversions droites :

$$u \leq_L v \iff \text{Inv}^*(u) = \text{Inv}^*(v) \quad \text{et} \quad \text{Inv}_R(u) \subseteq \text{Inv}_R(v). \quad (2.17)$$

Le diagramme de Hasse de l'ordre faible gauche pour la taille 3 est donné [figure 2.8](#). Une composante connexe de la taille 5 est donnée [figure 2.9](#).

L'ordre faibles droit et l'ordre faible gauche (sans chevauchement) sont composés de plusieurs composantes connexes. Une de ces composantes est l'ordre faible sur les permutations.

Proposition 2.25. *[BB94, Corollaire Théorème 8] Pour l'ordre faible droit, toute les composantes sont isomorphes à un intervalle de l'ordre faible droit sur les permutations.*

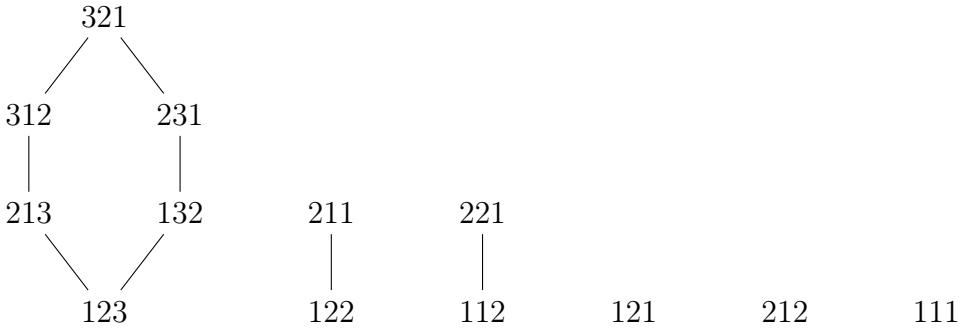


FIGURE 2.8 – Ordre faible gauche \leq_L sur les mots tassés de taille 3.

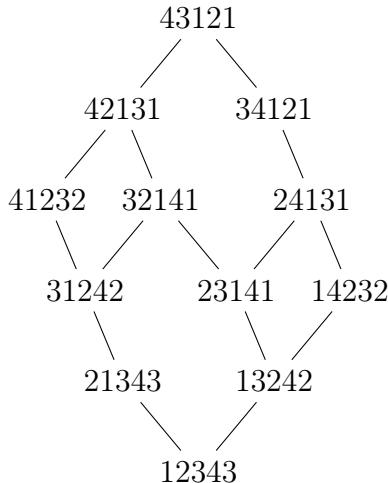


FIGURE 2.9 – Une composante de l'ordre faible gauche \leq_L sur les mots tassés de taille 5.

Nous généralisons la [proposition 2.18](#) de Loday et Ronco aux mots tassés. Cette généralisation est présentée dans [Var19] pour les [équations \(2.18\)](#) et [\(2.20\)](#) et dans [PR04, NT06] pour l'[équation \(2.19\)](#).

Proposition 2.26. *[Var19, PR04, NT06] Les 3 produits de mélange sur les mots tassés correspondent à la somme des éléments d'un intervalle d'un des 3 ordres faibles généralisés.*

Pour tous mots tassés u et v :

$$u \sqsupseteq v = \sum_{w \in [u \setminus v, v/u]_R} w, \quad (2.18)$$

$$u \sqcup v = \sum_{w \in [u \setminus v, u/v]_L^*} w, \quad (2.19)$$

$$u \sqsubseteq v = \sum_{w \in [u \setminus v, u/v]_L} w. \quad (2.20)$$

2.3.3 Ordre de raffinement

L'ordre de raffinement, que nous noterons \leq^* , est souvent défini sur les partitions d'ensembles ordonnées (comme dans [KLPN01, BZ05]) qui sont en bijection avec les mots tassés (voir [exemple 1.36](#)). Cet ordre est constitué de plusieurs composantes connexes qui sont toutes des hyper-cubes et qui sont des sous-posets du pseudo-permutoèdre [KLPN01]. Nous donnons ici la définition de cet ordre directement sur les mots tassés. Nous donnons également deux lemmes dont les preuves sont immédiates. Ils nous serviront dans le [chapitre 3](#), lorsque nous étudierons le changement de base fondé sur cet ordre partiel pour avoir un produit de mélange sans chevauchement.

Définition 2.27. L'**ordre de raffinement** \leq^* sur les mots tassés se définit pour tous mots tassés u et v par l'égalité des inversions droites et l'inclusion des demi-inversions :

$$u \leq^* v \iff \text{Inv}_R(u) = \text{Inv}_R(v) \quad \text{et} \quad \text{Inv}^*(u) \subseteq \text{Inv}^*(v). \quad (2.21)$$

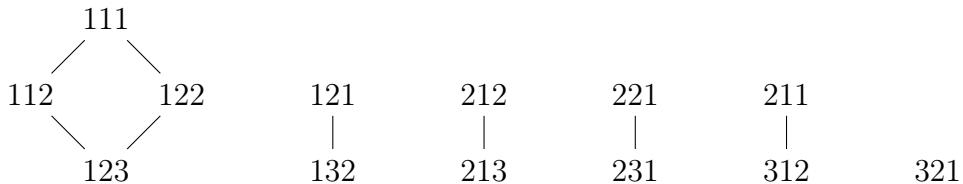


FIGURE 2.10 – Ordre de raffinement \leq^* sur les mots tassés de taille 3.

Nous rappelons la [remarque 2.16](#) qui précise que l'égalité des l'ensembles d'inversions droites est équivalent à l'égalité des standardisés.

Lemme 2.28. Pour $u \leq^* v$, soit m_u (resp. m_v) l'ensemble des positions des occurrences de la valeur maximale dans u (resp. v). Alors m_u est inclus dans m_v et toutes les positions dans m_v qui ne sont pas dans m_u sont plus petites que le minimum de m_u .

Démonstration. La preuve est immédiate d'après la définition de l'ordre de raffinement. \square

Lemme 2.29. Pour $u \leq^* v$, soit i et i' tels que $u|^{\leq i}$ et $v|^{\leq i'}$ soient de même taille, alors $u|^{\leq i} \leq^* v|^{\leq i'}$ et $u|^{>i} \leq^* v|^{>i'}$.

Démonstration. La preuve est immédiate d'après la définition de l'ordre de raffinement. \square

Chapitre 3

Algèbres de Hopf combinatoires

À ce stade, nous avons les outils combinatoires nécessaires pour étudier des algèbres de Hopf combinatoires. Ce chapitre est consacré à des rappels fondamentaux sur les algèbres de Hopf combinatoires. Nous étudierons 3 algèbres **FQSym**, **WQSym** et $\mathcal{H}_{P,R}^{\mathcal{D}}$ qui sont respectivement indexées par les permutations, les mots tassés et les forêts d'arbres plans décorés.

La suite de cette thèse étant fondée sur des théorèmes de rigidité sur les bigèbres bidendriformes définies par Foissy [Foi07], nous terminerons ce chapitre en rappelant les éléments principaux de sa théorie.

3.1 Définitions générales

Définition 3.1. Une **algèbre de Hopf** est une famille $(V, \mu, \iota, \Delta, \varepsilon, S)$ où

- V est un **K-espace vectoriel** avec \mathbb{K} un corps,
- le **produit** $\mu : V \otimes V \rightarrow V$ est une application associative :

$$\mu(\mu(x \otimes y) \otimes z) = \mu(x \otimes \mu(y \otimes z)), \quad (3.1)$$

- pour tout $x, y, z \in V$,
- l'**unité** $\iota : \mathbb{K} \rightarrow V$:

$$\mu(\iota(\alpha) \otimes x) = \alpha x = \mu(x \otimes \iota(\alpha)), \quad (3.2)$$

- pour tout $x \in V$ et tout $\alpha \in \mathbb{K}$,
- le **coproduit** $\Delta : V \rightarrow V \otimes V$ est une application coassociative :

$$(\text{Id} \otimes \Delta) \circ \Delta(x) = (\Delta \otimes \text{Id}) \circ \Delta(x), \quad (3.3)$$

- pour tout $x \in V$,
- la **counité** $\varepsilon : V \rightarrow \mathbb{K}$:

$$(\text{Id} \otimes \varepsilon) \circ \Delta(x) = x = (\varepsilon \otimes \text{Id}) \circ \Delta(x), \quad (3.4)$$

- pour tout $x \in V$,

— l'**antipode** $S : V \rightarrow V$ telle que :

$$(S * \text{Id})(x) = \iota(\varepsilon(x)) = (\text{Id} * S)(x), \quad (3.5)$$

pour tout $x \in V$, où $*$ est le **produit de convolution** :

$$f * g := \mu \circ (f \otimes g) \circ \Delta. \quad (3.6)$$

Les relations suivantes doivent être vérifiées :

$$\Delta(\iota) = \iota \otimes \iota, \quad (3.7)$$

$$\varepsilon(\iota) = 1, \quad (3.8)$$

$$\varepsilon(\mu(x \otimes y)) = \mu(\varepsilon(x) \otimes \varepsilon(y)), \quad (3.9)$$

pour tout $x, y \in V$.

Enfin, le produit et le coproduit sont **compatibles**, c'est-à-dire qu'ils sont des morphismes l'un pour l'autre :

$$\mu(\Delta(x) \otimes \Delta(y)) = \Delta(\mu(x \otimes y)), \quad (3.10)$$

pour tout $x, y \in V$.

La sous-structure (V, μ, ι) est une **algèbre**, la sous-structure (V, Δ, ε) est une **cogèbre**, la sous-structure $(V, \mu, \iota, \Delta, \varepsilon)$ est une **bigèbre**.

Définition 3.2. Une **algèbre de Hopf combinatoire** est une algèbre de Hopf où l'espace vectoriel sous-jacent V a pour base une classe combinatoire \mathcal{C} et que la graduation est compatible avec la structure d'algèbre de Hopf :

$$V = \text{Vect}_{\mathbb{K}}(\mathcal{C}) := \bigoplus_{n \geq 0} \text{Vect}_{\mathbb{K}}(\mathcal{C}_n), \quad (3.11)$$

et pour tout n ,

$$\Delta(\text{Vect}_{\mathbb{K}}(\mathcal{C}_n)) \subseteq \bigoplus_{i=0}^n (\text{Vect}_{\mathbb{K}}(\mathcal{C}_i) \otimes \text{Vect}_{\mathbb{K}}(\mathcal{C}_{n-i})), \quad (3.12)$$

$$\mu(\text{Vect}_{\mathbb{K}}(\mathcal{C}_i) \otimes \text{Vect}_{\mathbb{K}}(\mathcal{C}_j)) \subseteq \text{Vect}_{\mathbb{K}}(\mathcal{C}_{i+j}). \quad (3.13)$$

Si la classe combinatoire \mathcal{C} admet un unique objet de taille nulle, on dit que l'algèbre sur \mathcal{C} est **connexe**.

Proposition 3.3. [MM65, Proposition 8.2] Une bigèbre graduée et connexe est automatiquement munie d'une structure d'algèbre de Hopf. Avec la notation de Sweedler où $\Delta(x) = \sum x' \otimes x''$, nous donnons la définition récursive de l'antipode :

$$S(x) = \begin{cases} x & \text{si } \deg(x) = 0, \\ \sum_{x' \neq x} \mu(S(x') \otimes x'') & \text{sinon.} \end{cases} \quad (3.14)$$

Dans cette thèse nous étudions uniquement des algèbres de Hopf combinatoires connexes qui sont donc graduées. Nous ne nous intéresserons pas à l'étude de l'antipode. De plus l'unité et la counité se déduisent du produit et du coproduit. L'espace vectoriel, le produit et le coproduit caractérisent donc une algèbre de Hopf, nous les noterons donc par la famille suivante (V, μ, Δ) .

Définition 3.4. La connexité des algèbres nous permet de définir la notion de **coproduit réduit** $\tilde{\Delta}$ où les termes triviaux sont soustraits. Pour tout $x \in V$

$$\tilde{\Delta}(x) := \Delta(x) - (\epsilon \otimes x + x \otimes \epsilon). \quad (3.15)$$

Ici ϵ désigne l'unique objet de taille nulle de la classe combinatoire.

Définition 3.5. Pour une algèbre de Hopf combinatoire A , l'ensemble de ses éléments **primitifs** $\text{Prim}(A)$ sont les éléments dont le coproduit réduit est nul.

$$\text{Prim}(A) := \text{Ker}(\tilde{\Delta}). \quad (3.16)$$

Définition 3.6. Pour une algèbre de Hopf combinatoire A , les **séries de Hilbert** de l'algèbre $\mathcal{A}(z)$ et de l'espace des éléments primitifs $\mathcal{P}(z)$ sont les séries formelles des dimensions supérieures à 1.

$$\mathcal{A}(z) := \sum_{n=1}^{+\infty} \dim(A_n) z^n \quad (3.17)$$

$$\mathcal{P}(z) := \sum_{n=1}^{+\infty} \dim(\text{Prim}(A_n)) z^n. \quad (3.18)$$

Dans le cas d'une algèbre de Hopf combinatoire sur une classe \mathcal{C} , la série de Hilbert est égale à la série génératrice de la classe, $\mathcal{A}(z) = \mathcal{G}_{\mathcal{C}}(z)$.

Dualité

Les algèbres de Hopf combinatoires vont toujours par deux, une primale et une duale. Dans l'algèbre duale, les notions de produit et coproduits sont interverties.

Soit V un espace vectoriel de base B , son dual est l'espace, noté V^* , des formes linéaires sur V . Comme nous travaillons en dimension graduellement finie, V^* est isomorphe à V . Sa base est donnée par B^* tel que $b_i^*(b_j) = \delta_{i,j}$ pour $b_i^* \in B^*$ et $b_j \in B$. On utilise la notation du crochet de dualité $\langle \phi, x \rangle := \phi(x)$ pour $\phi \in V^*$ et $x \in V$. En particulier, si $\phi \in V^*$ et $b_i \in B$, alors $\langle \phi, b_i \rangle$ est le coefficient de b_i^* dans ϕ .

Définition 3.7. Soit (V, μ, Δ) , une algèbre de Hopf combinatoire, son **algèbre duale graduée** est (V^*, Δ^*, μ^*) où le produit Δ^* et le coproduit μ^* sont définis par :

$$\langle \mu^*(z), x \otimes y \rangle = \langle z, \mu(x, y) \rangle \quad \text{pour tout } x, y \in V \text{ et } z \in V^*, \quad (3.19)$$

$$\langle \Delta^*(x, y), z \rangle = \langle x \otimes y, \Delta(z) \rangle \quad \text{pour tout } z \in V \text{ et } x, y \in V^*. \quad (3.20)$$

Autrement dit, si on fixe une base $(e_i)_{i \in V}$ de V et si on considère la famille duale $(e_i^*)_{i \in V}$ qui est une base de V^* (car V est gradué), alors pour tout élément e_k^* de V^* , l'élément e_k apparaît dans le produit $\mu(e_i, e_j)$ si et seulement si la paire $e_i^* \otimes e_j^*$ apparaît dans le coproduit $\mu^*(e_k^*)$. De même pour le produit Δ^* et le coproduit Δ .

3.2 Exemples d'algèbres de Hopf combinatoires

Dans le [chapitre 1](#) nous avons vu plusieurs classes d'objets combinatoires et plusieurs opérations pour chacune d'entre elles. De plus, comme nous venons de le dire, une algèbre de Hopf combinatoire se caractérise par un espace vectoriel, un produit et un coproduit. À ce stade, nous pouvons présenter les algèbres de Hopf combinatoires dont nous aurons besoin dans la suite de cette thèse.

3.2.1 L'algèbre \mathbf{FQSym}

L'algèbre des fonctions quasi-symétriques libres (*Free Quasi-Symmetric functions*) \mathbf{FQSym} , également appelée l'algèbre de Malvenuto Reutenauer [[MR95](#)], est une algèbre de Hopf indexée par les **permutations**. Nous invitons le lecteur à lire [[DHT01](#), [DHNT08](#)] pour la définition de la réalisation polynomiale de l'algèbre. Nous donnons ici les définitions des bases \mathbb{G} et \mathbb{F} .

La base fondamentale duale \mathbb{G} : somme formelle de tous les mots ayant le même standardisé (voir [définition 1.16](#)). Pour toute permutation σ :

$$\mathbb{G}_\sigma := \sum_{\substack{w \in \mathcal{A}^*, \\ \text{std}(w)=\sigma}} w. \quad (3.21)$$

Produit : mélange des valeurs (voir [définition 1.28](#)) ou intervalle de l'ordre faible gauche (voir [proposition 2.18](#)). Pour toutes permutations μ et ν :

$$\mathbb{G}_\mu \mathbb{G}_\nu := \sum_{\sigma \in \mu \sqcup \nu} \mathbb{G}_\sigma, \quad \mathbb{G}_\mu \mathbb{G}_\nu := \sum_{\sigma \in [\mu \setminus \nu, \mu / \nu]_L} \mathbb{G}_\sigma. \quad (3.22)$$

Coproduit : somme formelle des paires des sous-mots des petites lettres et des sous-mots standardisés des grandes lettres (voir [exemple 1.2](#)). Pour toute permutation σ :

$$\Delta(\mathbb{G}_\sigma) := \sum_{i=0}^{|\sigma|} \mathbb{G}_{\sigma| \leq i} \otimes \mathbb{G}_{\text{std}(\sigma| > i)}. \quad (3.23)$$

Exemple 3.8.

$$\begin{aligned}\mathbb{G}_{132}\mathbb{G}_{21} &= \mathbb{G}_{132\mathbf{54}} + \mathbb{G}_{142\mathbf{53}} + \mathbb{G}_{152\mathbf{43}} + \mathbb{G}_{143\mathbf{52}} + \mathbb{G}_{153\mathbf{42}} + \\ &\quad \mathbb{G}_{154\mathbf{32}} + \mathbb{G}_{243\mathbf{51}} + \mathbb{G}_{253\mathbf{41}} + \mathbb{G}_{254\mathbf{31}} + \mathbb{G}_{354\mathbf{21}}, \\ \Delta(\mathbb{G}_{215364}) &= \mathbb{G}_\epsilon \otimes \mathbb{G}_{215364} + \mathbb{G}_1 \otimes \mathbb{G}_{14253} + \mathbb{G}_{21} \otimes \mathbb{G}_{3142} + \mathbb{G}_{213} \otimes \mathbb{G}_{231} + \\ &\quad \mathbb{G}_{2134} \otimes \mathbb{G}_{12} + \mathbb{G}_{21534} \otimes \mathbb{G}_1 + \mathbb{G}_{215364} \otimes \mathbb{G}_\epsilon.\end{aligned}$$

La base fondamentale \mathbb{F} : par dualité de la base \mathbb{G} .

Produit : mélange des positions (voir [définition 1.24](#)) ou intervalle de l'ordre faible droit (voir [proposition 2.18](#)). Pour toutes permutations μ et ν :

$$\mathbb{F}_\mu \mathbb{F}_\nu := \sum_{\sigma \in \mu \sqcup \nu} \mathbb{F}_\sigma, \quad \mathbb{F}_\mu \mathbb{F}_\nu := \sum_{\sigma \in [\mu \setminus \nu, \nu / \mu]_R} \mathbb{F}_\sigma. \quad (3.24)$$

Coproduit : déconcaténation standardisée, c'est-à-dire la somme formelle des paires des standardisés des préfixes et des suffixes (voir [exemple 1.2](#)). Pour toute permutation σ :

$$\Delta(\mathbb{F}_\sigma) := \sum_{i=0}^{|\sigma|} \mathbb{F}_{\text{std}(\sigma|_{\leq i})} \otimes \mathbb{F}_{\text{std}(\sigma|_{> i})}. \quad (3.25)$$

Exemple 3.9.

$$\begin{aligned}\mathbb{F}_{132}\mathbb{F}_{21} &= \mathbb{F}_{132\mathbf{54}} + \mathbb{F}_{13\mathbf{524}} + \mathbb{F}_{13\mathbf{542}} + \mathbb{F}_{1\mathbf{5324}} + \mathbb{F}_{1\mathbf{5342}} + \\ &\quad \mathbb{F}_{1\mathbf{5432}} + \mathbb{F}_{\mathbf{51324}} + \mathbb{F}_{\mathbf{51342}} + \mathbb{F}_{\mathbf{51432}} + \mathbb{F}_{\mathbf{54132}}, \\ \Delta(\mathbb{F}_{215364}) &= \mathbb{F}_\epsilon \otimes \mathbb{F}_{215364} + \mathbb{F}_1 \otimes \mathbb{F}_{14253} + \mathbb{F}_{21} \otimes \mathbb{F}_{3142} + \mathbb{F}_{213} \otimes \mathbb{F}_{132} + \\ &\quad \mathbb{F}_{2143} \otimes \mathbb{F}_{21} + \mathbb{F}_{21435} \otimes \mathbb{F}_1 + \mathbb{F}_{215364} \otimes \mathbb{F}_\epsilon.\end{aligned}$$

Isomorphisme de Hopf : l'inverse des permutations décrit un isomorphisme de Hopf entre les bases \mathbb{G} et \mathbb{F} . Pour toute permutation σ :

$$\mathbb{F}_\sigma \leftrightarrow \mathbb{G}_{\sigma^{-1}}. \quad (3.26)$$

3.2.2 L'algèbre **WQSym**

L'algèbre des fonctions quasi-symétriques en mots (*Word Quasi-Symmetric functions*) **WQSym** est l'algèbre de Hopf indexée par les **mots tassés**. Cette algèbre a été définie par Hivert dans [\[Hiv99\]](#). Nous verrons plus bas ([théorème 3.21](#)) que Novelli-Thibon ont prouvé que **WQSym** et sa duale sont des bigèbres bidendrififormes [\[NT06,](#)

Théorèmes 2.5 et 2.6]. Ils démontrent ce théorème à partir de la base monomiale $(\mathbb{M}_w)_{w \in \mathbf{PW}}$ dans laquelle les valeurs se chevauchent dans le produit de mélange (voir [équation \(3.28\)](#)). Cependant, pour notre étude de la structure bidendriforme, il sera plus simple de choisir, parmi toutes les bases connues dans la littérature [Hiv99, BZ05, NT06, Var19], une base dans laquelle le produit de mélange ne se chevauche pas.

C'est pourquoi, pour \mathbf{WQSym}^* , nous utiliserons la base $(\mathbb{Q}_w)_{w \in \mathbf{PW}}$ définie par Bergeron-Zabrocki [BZ05, équation 23] en utilisant la bijection classique entre les partitions ordonnées et les mots tassés (see [exemple 1.36](#)). Pour la primale \mathbf{WQSym} , par dualité, nous définissons la base que nous notons $(\mathbb{R}_w)_{w \in \mathbf{PW}}$.

Nous repartons donc de la définition de la base monomiale \mathbb{M} de Novelli-Thibon puis nous recalculons le produit et le coproduit dans la base \mathbb{Q} . Enfin par dualité nous calculons les expressions du produit et du coproduit dans la base \mathbb{R} .

La base monomiale \mathbb{M} : sur un alphabet X , la somme formelle de tous les mots ayant le même tassemement (voir [définition 1.39](#)).

$$\mathbb{M}_u := \sum_{\substack{w \in X^*, \\ \text{pack}(w)=u}} w. \quad (3.27)$$

Produit : mélange des valeurs avec chevauchement (voir [définition 1.51](#)) ou inter-intervalle de l'ordre faible gauche avec chevauchement (voir [proposition 2.26](#)). Pour tous mots tassés u et v :

$$\mathbb{M}_u \mathbb{M}_v := \sum_{w \in u \sqcup \sqcup v} \mathbb{M}_w, \quad \mathbb{M}_u \mathbb{M}_v := \sum_{w \in [u \setminus v, u/v]_L^*} \mathbb{M}_w. \quad (3.28)$$

Coproduit : somme formelle des paires des sous-mots des petites lettres et des sous-mots tassés des grandes lettres (voir [exemple 1.2](#)). Pour tout mot tassé w :

$$\Delta(\mathbb{M}_w) := \sum_{i=0}^{\max(w)} \mathbb{M}_{w| \leq i} \otimes \mathbb{M}_{\text{pack}(w| > i)}. \quad (3.29)$$

Exemple 3.10.

$$\begin{aligned} \mathbb{M}_{112} \mathbb{M}_{12} &= \mathbb{M}_{11212} + \mathbb{M}_{11213} + \mathbb{M}_{11223} + \mathbb{M}_{11234} + \mathbb{M}_{11312} + \mathbb{M}_{11323} + \mathbb{M}_{11324} + \\ &\quad \mathbb{M}_{11423} + \mathbb{M}_{22312} + \mathbb{M}_{22313} + \mathbb{M}_{22314} + \mathbb{M}_{22413} + \mathbb{M}_{33412}, \\ \Delta(\mathbb{M}_{212536434}) &= \mathbb{M}_\epsilon \otimes \mathbb{M}_{212536434} + \mathbb{M}_1 \otimes \mathbb{M}_{11425323} + \mathbb{M}_{212} \otimes \mathbb{M}_{314212} + \mathbb{M}_{21233} \otimes \mathbb{M}_{2311} + \\ &\quad \mathbb{M}_{2123434} \otimes \mathbb{M}_{12} + \mathbb{M}_{21253434} \otimes \mathbb{M}_1 + \mathbb{M}_{212536434} \otimes \mathbb{M}_\epsilon. \end{aligned}$$

La base \mathbb{Q} : définie par Bergeron-Zabrocki [BZ05, Equation 23] comme la somme de l'idéal supérieur pour l'ordre de raffinement sur la base \mathbb{M} . Pour tout mot tassé u :

$$\mathbb{Q}_u := \sum_{u \leq^* v} \mathbb{M}_v. \quad (3.30)$$

Exemple 3.11.

$$\begin{aligned} \mathbb{Q}_{123} &= \mathbb{M}_{123} + \mathbb{M}_{122} + \mathbb{M}_{112} + \mathbb{M}_{111} & \mathbb{Q}_{43132} &= \mathbb{M}_{43132} + \mathbb{M}_{32121} \\ \mathbb{Q}_{412234} &= \mathbb{M}_{412234} + \mathbb{M}_{312223} + \mathbb{M}_{311123} + \mathbb{M}_{211112} & \mathbb{Q}_{2131} &= \mathbb{M}_{2131} + \mathbb{M}_{2121} \end{aligned}$$

Produit : en partant du produit dans la base \mathbb{M} , il est démontré dans [BZ05, Theorem 17] que le produit dans la base \mathbb{Q} est le mélange des valeurs sans chevauchement (voir [définition 1.52](#)) ou l'intervalle de l'ordre faible gauche (voir [proposition 2.26](#)). Pour tous mots tassés u et v :

$$\mathbb{Q}_u \mathbb{Q}_v := \sum_{w \in u \sqcup v} \mathbb{Q}_w, \quad \mathbb{Q}_u \mathbb{Q}_v := \sum_{w \in [u \setminus v, u/v]_L} \mathbb{Q}_w. \quad (3.31)$$

Coproduit : en partant du coproduit dans la base \mathbb{M} , nous prouvons que le coproduit est décrit comme la somme formelle des paires des sous-mots des petites lettres et des sous-mots tassés des grandes lettres (voir [exemple 1.2](#)). Pour tout mot tassé u :

$$\begin{aligned} \Delta(\mathbb{Q}_u) &= \sum_{u \leq^* v} \Delta(\mathbb{M}_v) \\ &= \sum_{u \leq^* v} \left(\sum_{i=0}^{\max(v)} \mathbb{M}_{v| \leq i} \otimes \mathbb{M}_{\text{pack}(v| > i)} \right) \\ &= \sum_{i=0}^{\max(u)} \left(\sum_{u| \leq i \leq^* v} \mathbb{M}_v \otimes \sum_{u| > i \leq^* v'} \mathbb{M}_{\text{pack}(v')} \right) \quad (\text{d'après le lemme 2.29}) \\ &= \sum_{i=0}^{\max(u)} \mathbb{Q}_{u| \leq i} \otimes \mathbb{Q}_{\text{pack}(u| > i)}. \end{aligned} \quad (3.32)$$

Exemple 3.12.

$$\begin{aligned} \mathbb{Q}_{112} \mathbb{Q}_{12} &= \mathbb{Q}_{11234} + \mathbb{Q}_{11324} + \mathbb{Q}_{11423} + \mathbb{Q}_{22314} + \mathbb{Q}_{22413} + \mathbb{Q}_{33412}, \\ \Delta(\mathbb{Q}_{212536434}) &= \mathbb{Q}_\epsilon \otimes \mathbb{Q}_{212536434} + \mathbb{Q}_1 \otimes \mathbb{Q}_{11425323} + \mathbb{Q}_{212} \otimes \mathbb{Q}_{314212} + \mathbb{Q}_{21233} \otimes \mathbb{Q}_{2311} + \\ &\quad \mathbb{Q}_{2123434} \otimes \mathbb{Q}_{12} + \mathbb{Q}_{21253434} \otimes \mathbb{Q}_1 + \mathbb{Q}_{212536434} \otimes \mathbb{Q}_\epsilon. \end{aligned}$$

La base \mathbb{R} : par dualité de la base \mathbb{Q} .

Produit : mélange des positions (voir [définition 1.47](#)) ou intervalle de l'ordre faible droit (voir [proposition 2.26](#)). Pour tous mots tassés u et v :

$$\mathbb{R}_u \mathbb{R}_v := \sum_{w \in u \sqcup v} \mathbb{R}_w, \quad \mathbb{R}_u \mathbb{R}_v := \sum_{w \in [u \setminus v, u/v]_R} \mathbb{R}_w. \quad (3.33)$$

Coproduit : déconcaténation réduite avec tassemement, c'est-à-dire les paires des tassemements des préfixes et des suffixes où deux lettres de même valeur ne peuvent être séparées. Pour tout mot tassé w :

$$\Delta(\mathbb{R}_w) := \sum_{\substack{i=0 \\ w|_{\leq i} \cap w|_{>i} = \emptyset}}^{|w|} \mathbb{R}_{\text{pack}(w|_{\leq i})} \otimes \mathbb{R}_{\text{pack}(w|_{>i})}. \quad (3.34)$$

Exemple 3.13.

$$\begin{aligned} \mathbb{R}_{122} \mathbb{R}_{21} &= \mathbb{R}_{12243} + \mathbb{R}_{12423} + \mathbb{R}_{12432} + \mathbb{R}_{14223} + \mathbb{R}_{14232} + \\ &\quad \mathbb{R}_{14322} + \mathbb{R}_{41223} + \mathbb{R}_{41232} + \mathbb{R}_{41322} + \mathbb{R}_{43122}, \\ \Delta(\mathbb{R}_{212536434}) &= \mathbb{R}_\epsilon \otimes \mathbb{R}_{212536434} + \mathbb{R}_{212} \otimes \mathbb{R}_{314212} + \mathbb{R}_{2123} \otimes \mathbb{R}_{13212} + \mathbb{R}_{212536434} \otimes \mathbb{R}_\epsilon. \end{aligned}$$

3.2.3 L'algèbre $\mathcal{H}_{P,R}^{\mathcal{D}}$

Dans [\[Kre97, CK98\]](#), Connes-Kreimer introduisent une algèbre d'arbres enracinés \mathcal{H}_R . Foissy généralise cette construction dans [\[Foi02a, Foi02b\]](#) en introduisant l'algèbre $\mathcal{H}_{P,R}^{\mathcal{D}}$ des arbres plans enracinés décorés. Il construit également une base (e_f) de l'algèbre duale qui est particulièrement bien adaptée au calcul du coproduit ; en particulier, elle permet de trouver facilement les éléments primitifs de $\mathcal{H}_{P,R}^{\mathcal{D}}$. C'est cette base (e_f) et uniquement celle-ci que nous allons présenter ici.

La base e : définie par Foissy [\[Foi02a, Foi02b\]](#) comme la base duale de l'algèbre $\mathcal{H}_{P,R}^{\mathcal{D}}$.

Produit : [\[Foi02b, Proposition 28\]](#) produit de greffe (voir [définition 1.78](#)). Pour toutes forêts d'arbres plans enracinés décorés f_1 et f_2 :

$$e_{f_1} e_{f_2} := \sum_{f \in f_1 \star f_2} e_f. \quad (3.35)$$

Coproduit : [Foi02a, Théorème 35.7.c] déconcaténation des forêts en arbres. Pour toute forêt d'arbres plans enracinés décorés $f = [t_1, \dots, t_k]$:

$$\Delta(e_f) := \sum_{i=0}^k e_{[t_1, \dots, t_i]} \otimes e_{[t_{i+1}, \dots, t_k]}. \quad (3.36)$$

Exemple 3.14.

$$e \begin{array}{c} 1 \\ \circ \\ 2 \end{array} e \begin{array}{c} 1 \\ \circ \\ 2 \end{array} \circ 3 = 2e \begin{array}{c} 1 \\ \circ \\ 2 \end{array} \begin{array}{c} 1 \\ \circ \\ 2 \end{array} \circ 3 + e \begin{array}{c} 1 \\ \circ \\ 2 \end{array} \begin{array}{c} 1 \\ \circ \\ 2 \end{array} \begin{array}{c} 3 \\ \circ \\ 2 \end{array} + e \begin{array}{c} 1 \\ \circ \\ 2 \end{array} \begin{array}{c} 3 \\ \circ \\ 2 \end{array} + e \begin{array}{c} 1 \\ \circ \\ 2 \end{array} \begin{array}{c} 1 \\ \circ \\ 2 \end{array} \begin{array}{c} 3 \\ \circ \\ 2 \end{array} + e \begin{array}{c} 1 \\ \circ \\ 2 \end{array} \begin{array}{c} 3 \\ \circ \\ 2 \end{array} \begin{array}{c} 1 \\ \circ \\ 2 \end{array} + e \begin{array}{c} 1 \\ \circ \\ 2 \end{array} \begin{array}{c} 1 \\ \circ \\ 2 \end{array} \begin{array}{c} 3 \\ \circ \\ 2 \end{array},$$

$$\Delta(e \begin{array}{c} 1 \\ \circ \\ 2 \end{array} \circ 3) = e \begin{array}{c} 1 \\ \circ \\ 2 \end{array} \otimes e \begin{array}{c} 1 \\ \circ \\ 2 \end{array} \circ 3 + e \begin{array}{c} 1 \\ \circ \\ 2 \end{array} \otimes e \begin{array}{c} 3 \\ \circ \\ 2 \end{array} + e \begin{array}{c} 1 \\ \circ \\ 2 \end{array} \circ 3 \otimes e \begin{array}{c} 1 \\ \circ \\ 2 \end{array} + e \begin{array}{c} 1 \\ \circ \\ 2 \end{array} \circ 3 \otimes e \begin{array}{c} 1 \\ \circ \\ 2 \end{array}$$

Dans la base primale ((e_f) étant la base duale) le produit est une concaténation d'arbres et le coproduit se définit par des coupes admissibles. Plus de détails sont disponibles dans [Foi02a, Foi02b]. Nous ne donnerons pas plus de détails sur la base primale. Nous ne nous servirons que de la base duale (e_f) dans la suite de cette thèse. En effet, cette base a été construite dans le but de distinguer facilement les éléments primitifs (éléments dont le coproduit réduit [définition 3.4](#) est nul), dont nous reparlerons dans la [section 3.3.3](#).

3.3 Bigèbres bidendriformes

Le but de cette section est de rappeler les éléments de la définition des bigèbres bidendriformes. Cette notion a été définie par Foissy [Foi07] et elle est primordiale dans cette thèse. Il s'agit de scinder le produit et le coproduit en demi-produits et de vérifier un certain nombre de relations.

Les algèbres de Hopf que nous avons citées jusqu'ici sont toutes des bigèbres bidendriformes. Nous détaillerons les définitions des demi-produits et demi-coproduits. Nous terminerons cette section en donnant les théorèmes principaux qui témoignent de la rigidité des bigèbres bidendriformes.

3.3.1 Définitions

Définition 3.15. Une **algèbre dendriforme** (voir [Lod01, LR98, Ron00, Ron02]) A est un \mathbb{K} -espace vectoriel, munie de deux opérations binaire et bilinéaire \prec, \succ satisfaisant les trois axiomes suivants, pour tous $a, b, c \in A$:

$$(a \prec b) \prec c = a \prec (b \prec c + b \succ c), \quad (3.37)$$

$$(a \succ b) \prec c = a \succ (b \prec c), \quad (3.38)$$

$$(a \prec b + a \succ b) \succ c = a \succ (b \succ c). \quad (3.39)$$

La somme des trois relations des équations (3.37) à (3.39) montre que le produit défini par $a \cdot b := a \prec b + a \succ b$ est associatif.

En ajoutant la composante homogène de degré nul cela définit une structure algébrique unitaire sur $\mathbb{K} \oplus A$.

Définition 3.16. En dualisant cette notion nous pouvons définir la notion de **cogèbre codendriforme** (voir [Foi07]), qui est un \mathbb{K} -espace vectoriel, muni de deux coopérations binaires (*i.e.*, applications linéaires $A \rightarrow A \otimes A$), notées Δ_\prec et Δ_\succ , satisfaisant les trois axiomes duals des équations (3.37) à (3.39) :

$$(\Delta_\prec \otimes \text{Id}) \circ \Delta_\prec(a) = (\text{Id} \otimes \Delta_\prec + \text{Id} \otimes \Delta_\succ) \circ \Delta_\prec(a), \quad (3.40)$$

$$(\Delta_\succ \otimes \text{Id}) \circ \Delta_\prec(a) = (\text{Id} \otimes \Delta_\prec) \circ \Delta_\succ(a), \quad (3.41)$$

$$(\Delta_\prec \otimes \text{Id} + \Delta_\succ \otimes \text{Id}) \circ \Delta_\succ(a) = (\text{Id} \otimes \Delta_\succ) \circ \Delta_\succ(a). \quad (3.42)$$

La somme des trois relations équations (3.40) à (3.42) montre que le coproduit réduit défini par $\tilde{\Delta}(a) := \Delta_\prec(a) + \Delta_\succ(a)$ est coassociatif.

Sur $\mathbb{K} \oplus A$, le coproduit défini par $\Delta(a) := 1 \otimes a + a \otimes 1 + \tilde{\Delta}(a)$ est coassociatif et counitaire.

Définition 3.17. Une **bigèbre bidendriforme** est un \mathbb{K} -espace vectoriel qui est à la fois une algèbre dendriforme et une cogèbre codendriforme et qui satisfait, de plus, quatre relations concernant respectivement \prec et \succ avec Δ_\prec et Δ_\succ . Dans ces équations nous utilisons la notation d'Einstein $\tilde{\Delta}(a) = a' \otimes a''$ et $\Delta_\alpha(b) = b'_\alpha \otimes b''_\alpha$ avec $\alpha \in \{\prec, \succ\}$.

$$\Delta_\succ(a \succ b) = a'b'_\succ \otimes a'' \succ b''_\succ + b'_\succ \otimes a \succ b''_\succ + ab'_\succ \otimes b''_\succ + a' \otimes a'' \succ b + a \otimes b, \quad (3.43)$$

$$\Delta_\succ(a \prec b) = a'b'_\succ \otimes a'' \prec b''_\prec + b'_\prec \otimes a \prec b''_\prec + a' \otimes a'' \prec b, \quad (3.44)$$

$$\Delta_\prec(a \succ b) = a'b'_\prec \otimes a'' \succ b''_\prec + b'_\prec \otimes a \succ b''_\prec + ab'_\prec \otimes b''_\prec, \quad (3.45)$$

$$\Delta_\prec(a \prec b) = a'b'_\prec \otimes a'' \prec b''_\prec + b'_\prec \otimes a \prec b''_\prec + a'b \otimes a'' + b \otimes a. \quad (3.46)$$

La somme de ces quatre relations équations (3.43) à (3.46) montre que \cdot et Δ (définis comme précédemment) donne bien la définition d'une bigèbre.

3.3.2 Reprise des exemples

Les algèbres **FQSym** et **WQSym** sont des bigèbres bidendriformes [Foi11, NT06]. Nous allons simplement décrire comment les produits et coproduits se scindent en deux. Nous donnerons ainsi les définitions des demi-produits et demi-coproduits pour toutes les bases que nous avons définies plus haut.

Nous pourrions noter \prec et \succ puis Δ_\prec et Δ_\succ pour chaque demi-produit et demi-coproduit et donner un sens différent à ces symboles en fonction de la base dans laquelle ils sont utilisés. Cependant, pour plus de cohérence et de clarté, nous utiliserons les mêmes notations que pour les différents demi-produits que nous avons présentés dans le chapitre 1. Pour les demi-coproduits dans une base donnée, nous utiliserons, en indice du delta, les mêmes symboles que pour les demi-produits de cette base.

L'algèbre $(\mathbf{FQSym})_+$:

$$(\mathbf{FQSym})_+ := Vect(\mathbb{G}_\sigma \mid \sigma \in \mathfrak{S}_n, n \geq 1).$$

La base fondamentale duale \mathbb{G}

Demi-produits : demi-produits de mélange des valeurs (voir [définition 1.30](#)). Pour toutes permutations non vides μ et ν :

$$\mathbb{G}_\mu \overline{\preceq} \mathbb{G}_\nu := \sum_{\sigma \in \mu \overline{\preceq} \nu} \mathbb{G}_\sigma, \quad \mathbb{G}_\mu \overline{\sum} \mathbb{G}_\nu := \sum_{\sigma \in \mu \overline{\sum} \nu} \mathbb{G}_\sigma. \quad (3.47)$$

Demi-coproduits : scindement du coproduit de la base \mathbb{G} en fonction de la dernière lettre. Pour toute permutation non vide σ :

$$\Delta_{\overline{\preceq}}(\mathbb{G}_\sigma) := \sum_{i=\sigma_{|\sigma|}}^{|\sigma|-1} \mathbb{G}_{\sigma| \leq i} \otimes \mathbb{G}_{\text{std}(\sigma| > i)}, \quad \Delta_{\overline{\sum}}(\mathbb{G}_\sigma) := \sum_{i=1}^{\sigma_{|\sigma|}-1} \mathbb{G}_{\sigma| \leq i} \otimes \mathbb{G}_{\text{std}(\sigma| > i)}. \quad (3.48)$$

Exemple 3.18.

$$\begin{aligned} \mathbb{G}_{132} \overline{\preceq} \mathbb{G}_{12} &= \mathbb{G}_{15234} + \mathbb{G}_{15324} + \mathbb{G}_{15423} + \mathbb{G}_{25314} + \mathbb{G}_{25413} + \mathbb{G}_{35412}, \\ \mathbb{G}_{132} \overline{\sum} \mathbb{G}_{12} &= \mathbb{G}_{13245} + \mathbb{G}_{14235} + \mathbb{G}_{14325} + \mathbb{G}_{24315}, \\ \Delta_{\overline{\preceq}}(\mathbb{G}_{215364}) &= \mathbb{G}_{2134} \otimes \mathbb{G}_{12} + \mathbb{G}_{21534} \otimes \mathbb{G}_1, \\ \Delta_{\overline{\sum}}(\mathbb{G}_{215364}) &= \mathbb{G}_1 \otimes \mathbb{G}_{14253} + \mathbb{G}_{21} \otimes \mathbb{G}_{3142} + \mathbb{G}_{213} \otimes \mathbb{G}_{231}. \end{aligned}$$

La base fondamentale \mathbb{F}

Demi-produits : demi-produits de mélange des positions (voir [définition 1.26](#)). Pour toutes permutations non vides μ et ν :

$$\mathbb{F}_\mu \overline{\prec} \mathbb{F}_\nu := \sum_{\sigma \in \mu \overline{\prec} \nu} \mathbb{F}_\sigma, \quad \mathbb{F}_\mu \overline{\succ} \mathbb{F}_\nu := \sum_{\sigma \in \mu \overline{\succ} \nu} \mathbb{F}_\sigma. \quad (3.49)$$

Coproduit : scindement du coproduit de la base \mathbb{F} en fonction de la position de la lettre maximale. Pour toute permutation non vide σ :

$$\Delta_{\overline{\prec}}(\mathbb{F}_\sigma) := \sum_{\substack{i=k \\ \sigma_k = \max(\sigma)}}^{|\sigma|-1} \mathbb{F}_{\text{std}(\sigma| \leq i)} \otimes \mathbb{F}_{\text{std}(\sigma| > i)}, \quad (3.50)$$

$$\Delta_{\overline{\succ}}(\mathbb{F}_\sigma) := \sum_{\substack{i=1 \\ \sigma_k = \max(\sigma)}}^{k-1} \mathbb{F}_{\text{std}(\sigma| \leq i)} \otimes \mathbb{F}_{\text{std}(\sigma| > i)}. \quad (3.51)$$

Exemple 3.19.

$$\begin{aligned}\mathbb{F}_{213} \setminus \mathbb{F}_{12} &= \mathbb{F}_{21453} + \mathbb{F}_{24153} + \mathbb{F}_{24513} + \mathbb{F}_{42153} + \mathbb{F}_{42513} + \mathbb{F}_{45213}, \\ \mathbb{F}_{221} \setminus \mathbb{F}_{12} &= \mathbb{F}_{21345} + \mathbb{F}_{21435} + \mathbb{F}_{24135} + \mathbb{F}_{42135}, \\ \Delta_{\setminus}(\mathbb{F}_{21534}) &= \mathbb{F}_{213} \otimes \mathbb{F}_{12} + \mathbb{F}_{2143} \otimes \mathbb{F}_1, \\ \Delta_{\setminus}(\mathbb{F}_{21534}) &= \mathbb{F}_1 \otimes \mathbb{F}_{1423} + \mathbb{F}_{21} \otimes \mathbb{F}_{312}.\end{aligned}$$

Isomorphisme bidendrifforme : l'inverse des permutations décrit également un isomorphisme bidendrifforme entre les bases \mathbb{G} et \mathbb{F} .

L'algèbre $(\mathbf{WQSym})_+$:

$$(\mathbf{WQSym})_+ := Vect(\mathbb{M}_u \mid u \in \mathbf{PW}_n, n \geq 1).$$

La base monomiale \mathbb{M}

Demi-produits : demi-produits de mélange des valeurs avec chevauchement (voir [définition 1.54](#)). Pour tous mots tassés non vides u et v :

$$\mathbb{M}_u \preceq \mathbb{M}_v := \sum_{w \in u \preceq v} \mathbb{M}_w, \quad \mathbb{M}_u \succeq \mathbb{M}_v := \sum_{w \in u \succeq v} \mathbb{M}_w. \quad (3.52)$$

Demi-coproduits : scindement du coproduit de la base \mathbb{M} en fonction de la dernière lettre. Pour tout mot tassé non vide w :

$$\Delta_{\preceq}(\mathbb{M}_w) := \sum_{i=w_{|w|}}^{\max(w)-1} \mathbb{M}_{w| \leq i} \otimes \mathbb{M}_{\text{pack}(w|^{>i})}, \quad (3.53)$$

$$\Delta_{\succeq}(\mathbb{M}_w) := \sum_{i=1}^{w_{|w|}-1} \mathbb{M}_{w| \leq i} \otimes \mathbb{M}_{\text{pack}(w|^{>i})}. \quad (3.54)$$

Exemple 3.20.

$$\begin{aligned}\mathbb{M}_{112} \preceq \mathbb{M}_{12} &= \mathbb{M}_{11312} + \mathbb{M}_{11423} + \mathbb{M}_{22312} + \mathbb{M}_{22413} + \mathbb{M}_{33412}, \\ \mathbb{M}_{112} \succeq \mathbb{M}_{12} &= \mathbb{M}_{11212} + \mathbb{M}_{11213} + \mathbb{M}_{11223} + \mathbb{M}_{11234} + \mathbb{M}_{11323} + \mathbb{M}_{11324} + \mathbb{M}_{22313} + \mathbb{M}_{22314}, \\ \Delta_{\preceq}(\mathbb{M}_{212536434}) &= \mathbb{M}_{2123434} \otimes \mathbb{M}_{12} + \mathbb{M}_{21253434} \otimes \mathbb{M}_1, \\ \Delta_{\succeq}(\mathbb{M}_{212536434}) &= \mathbb{M}_1 \otimes \mathbb{M}_{11425323} + \mathbb{M}_{212} \otimes \mathbb{M}_{314212} + \mathbb{M}_{21233} \otimes \mathbb{M}_{2311}.\end{aligned}$$

C'est sur cette base que le théorème suivant a été prouvé par Novelli-Thibon.

Théorème 3.21. [[NT06](#), Theorem 2.5] $((\mathbf{WQSym})_+, \preceq, \succeq, \Delta_{\preceq}, \Delta_{\succeq})$ est une bigèbre bidendrifforme.

La base \mathbb{Q}

Demi-produits : grâce au lemme 2.28 nous pouvons déduire que les demi-produits dans cette base sont les demi-produits de mélange de valeurs sans chevauchement (voir définition 1.55). Pour tous mots tassés non vides u et v :

$$\mathbb{Q}_u \overline{\preceq} \mathbb{Q}_v := \sum_{w \in u \overline{\preceq} v} \mathbb{Q}_w, \quad \mathbb{Q}_u \overline{\succeq} \mathbb{Q}_v := \sum_{w \in u \overline{\succeq} v} \mathbb{Q}_w. \quad (3.55)$$

Demi-coproduits : de même, grâce au lemme 2.28 nous pouvons déduire que les demi-coproduits dans cette base sont issus du scindement en fonction de la dernière lettre. Pour tout mot tassé non vide w :

$$\Delta_{\overline{\preceq}}(\mathbb{Q}_u) := \sum_{i=u_n}^{\max(u)-1} \mathbb{Q}_{u|_{\leq i}} \otimes \mathbb{Q}_{\text{pack}(u|_{>i})}, \quad (3.56)$$

$$\Delta_{\overline{\succeq}}(\mathbb{Q}_u) := \sum_{i=1}^{u_n-1} \mathbb{Q}_{u|_{\leq i}} \otimes \mathbb{Q}_{\text{pack}(u|_{>i})}. \quad (3.57)$$

Exemple 3.22.

$$\begin{aligned} \mathbb{Q}_{1312} \overline{\preceq} \mathbb{Q}_{12} &= \mathbb{Q}_{151234} + \mathbb{Q}_{151324} + \mathbb{Q}_{151423} + \mathbb{Q}_{252314} + \mathbb{Q}_{252413} + \mathbb{Q}_{353412}, \\ \mathbb{Q}_{1312} \overline{\succeq} \mathbb{Q}_{12} &= \mathbb{Q}_{131245} + \mathbb{Q}_{141235} + \mathbb{Q}_{141325} + \mathbb{Q}_{242315}, \\ \Delta_{\overline{\preceq}}(\mathbb{Q}_{212536434}) &= \mathbb{Q}_{2123434} \otimes \mathbb{Q}_{12} + \mathbb{Q}_{21253434} \otimes \mathbb{Q}_1, \\ \Delta_{\overline{\succeq}}(\mathbb{Q}_{212536434}) &= \mathbb{Q}_1 \otimes \mathbb{Q}_{11425323} + \mathbb{Q}_{212} \otimes \mathbb{Q}_{314212} + \mathbb{Q}_{21233} \otimes \mathbb{Q}_{2311}. \end{aligned}$$

La base \mathbb{R}

Demi-produits : demi-produits de mélange des positions (voir définition 1.49). Pour tous mots tassés non vides u et v :

$$\mathbb{R}_u \overline{\prec} \mathbb{R}_v := \sum_{w \in u \overline{\prec} v} \mathbb{R}_w, \quad \mathbb{R}_u \overline{\succ} \mathbb{R}_v := \sum_{w \in u \overline{\succ} v} \mathbb{R}_w. \quad (3.58)$$

Demi-coproduits : scindement du coproduit de la base \mathbb{R} en fonction des positions des lettres de valeurs maximales. Pour tout mot tassé non vide w :

$$\Delta_{\overline{\prec}}(\mathbb{R}_u) := \sum_{\substack{i=1 \\ w|_{\leq i} \cap w|_{>i} = \emptyset \\ w_k = \max(w)}}^{|w|-1} \mathbb{R}_{\text{pack}(w|_{\leq i})} \otimes \mathbb{R}_{\text{pack}(w|_{>i})}, \quad (3.59)$$

$$\Delta_{\overline{\succ}}(\mathbb{R}_u) := \sum_{\substack{i=1 \\ w|_{\leq i} \cap w|_{>i} = \emptyset \\ w_k = \max(w)}}^{k-1} \mathbb{R}_{\text{pack}(w|_{\leq i})} \otimes \mathbb{R}_{\text{pack}(w|_{>i})}. \quad (3.60)$$

Exemple 3.23.

$$\mathbb{R}_{211} \overline{\prec} \mathbb{R}_{12} = \mathbb{R}_{21341} + \mathbb{R}_{23141} + \mathbb{R}_{23411} + \mathbb{R}_{32141} + \mathbb{R}_{32411} + \mathbb{R}_{34211},$$

$$\mathbb{R}_{221} \overline{\succ} \mathbb{R}_{12} = \mathbb{R}_{21134} + \mathbb{R}_{21314} + \mathbb{R}_{23114} + \mathbb{R}_{32114},$$

$$\Delta_{\overline{\prec}}(\mathbb{R}_{2125334}) = \mathbb{R}_{2123} \otimes \mathbb{R}_{112} + \mathbb{R}_{212433} \otimes \mathbb{R}_1,$$

$$\Delta_{\overline{\succ}}(\mathbb{R}_{2125334}) = \mathbb{R}_{212} \otimes \mathbb{R}_{3112}.$$

Avec les demi-produits et demi-coproduits ainsi calculés, nous déduisons à partir du [théorème 3.21](#) le théorème suivant.

Théorème 3.24. $((\mathbf{WQSym})_+, \overline{\preceq}, \overline{\succeq}, \Delta_{\overline{\preceq}}, \Delta_{\overline{\succeq}})$ et $((\mathbf{WQSym}^*)_+, \overline{\prec}, \overline{\succ}, \Delta_{\overline{\prec}}, \Delta_{\overline{\succ}})$ sont deux bigèbres bidendrifomes duals l'une de l'autre.

L'algèbre $(\mathcal{H}_{P,R}^{\mathcal{D}})_+$:

$$(\mathcal{H}_{P,R}^{\mathcal{D}})_+ := Vect(e_f \mid f \in F_P, f \neq []).$$

La base e

Demi-produits : demi-produits de greffe (voir [définition 1.80](#)). Pour toutes forêts planes non vides f_1 et f_2 :

$$e_{f_1} \prec e_{f_2} := \sum_{f \in f_1 \prec f_2} e_f, \quad e_{f_1} \succ e_{f_2} := \sum_{f \in f_1 \succ f_2} e_f. \quad (3.61)$$

Se reporter à l'[exemple 1.81](#) pour un exemple de demi-produit de greffe.

Seul l'aspect dendriforme de $\mathcal{H}_{P,R}^{\mathcal{D}}$ nous intéresse, nous ne donnerons pas d'expression des demi-coproduits. Par contre, des relations sur ces demi-produits vont être primordiales. En effet, ces deux demi-produits vérifient deux relations que nous allons présenter dans la section suivante (voir [proposition 3.25](#), [théorème 3.27](#), et [corollaire 3.29](#)).

3.3.3 Théorème de Cartier-Milnor-Moore pour les bigèbres bidendriformes

Nous terminons cette partie en rappelant les résultats principaux de Foissy [Foi07, Foi11] sur la rigidité des bigèbres bidendriformes basés sur le travail de Chapoton et Ronco [Cha02, Ron00]. Ce travail est fondé sur deux résultats analogues au théorème de Cartier-Milnor-Moore.

Cette première proposition est extraite de la preuve de [Foi11, Proposition 6].

Proposition 3.25. *Soit A une bigèbre bidendriforme et soit $p_1 \dots p_n \in \text{Prim}(A)$ des éléments primitifs de A . L'application suivante*

$$p_1 \otimes p_2 \otimes \dots \otimes p_n \mapsto p_1 \prec (p_2 \prec (\dots \prec p_n) \dots). \quad (3.62)$$

est un isomorphisme de cogèbres de $T^+(\text{Prim}(A))$ (la partie non triviale du produit tensoriel d'algèbre munie de la déconcaténation comme coproduit) vers A . Ainsi, si $(p_i)_{i \in I}$ est une base de $\text{Prim}(A)$, alors la famille $(p_{w_1} \prec (p_{w_2} \prec (\dots \prec p_{w_n}) \dots))_w$, avec $w = w_1 \dots w_n$ désignant un mot non vide sur I , définit une base de A . Cela implique la relation suivante sur les séries de Hilbert :

$$\mathcal{A} = \frac{\mathcal{P}}{(1 - \mathcal{P})}. \quad (3.63)$$

Il est possible d'approfondir l'analyse de $\text{Prim}(A)$ avec l'espace des éléments totalement primitifs d'une algèbre A .

Définition 3.26. Pour une bigèbre bidendriforme A , l'ensemble de ses éléments **totallement primitifs** $\text{TPrim}(A)$ sont les éléments dont les demi-coproduits sont nuls.

$$\text{TPrim}(A) := \text{Ker}(\Delta_\prec) \cap \text{Ker}(\Delta_\succ). \quad (3.64)$$

La série de Hilbert $\mathcal{T}(z)$, associée à $\text{TPrim}(A)$, est définie de la manière suivante :

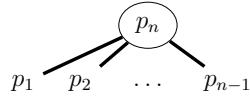
$$\mathcal{T}(z) := \sum_{n=1}^{+\infty} \dim(\text{TPrim}(A_n)) z^n \quad (3.65)$$

Pour le théorème suivant, nous rappelons rapidement ce qu'est une **algèbre de brace** et nous invitons le lecteur à lire [Ron00] pour plus de détails. Une algèbre de brace est un \mathbb{K} -espace vectoriel muni d'une opération n -multilinéaire notée $\langle \dots \rangle$ pour tout $n \geq 2$ et qui satisfait certaines relations que nous ne détaillerons pas ici.

Théorème 3.27 ([Foi11, Théorème 4 et 5]). *Soit A une bigèbre bidendriforme. Alors $\text{Prim}(A)$ est généré librement en tant qu'algèbre de brace par $\text{TPrim}(A)$ avec l'opération de crochets définie comme ceci :*

$$\begin{aligned} \langle p_1, \dots, p_{n-1}; p_n \rangle &:= \sum_{i=0}^{n-1} (-1)^{n-1-i} \\ &(p_1 \prec (p_2 \prec (\dots \prec p_i) \dots)) \succ p_n \prec ((\dots (p_{i+1} \succ p_{i+2}) \succ \dots) \succ p_{n-1}). \end{aligned}$$

Une base de $\text{Prim}(A)$ est décrite par les arbres plans décorés par des éléments de $\text{TPrim}(A)$ (voir [Ron00, Ron02, Cha02, Foi11]), comme montré ci-dessous.



p_n , qui décore la racine, est un élément de $\text{TPrim}(A)$ et p_1, \dots, p_{n-1} , qui sont des arbres greffés sur la racine, sont des éléments de $\text{Prim}(A)$.

Cela se reflète par les relations suivantes sur les séries de Hilbert [Foi07, Corollary 37] :

$$\mathcal{T} = \mathcal{A}/(1 + \mathcal{A})^2 \quad \text{ou de façon équivalente} \quad \mathcal{P} = \mathcal{T}(1 + \mathcal{A}). \quad (3.66)$$

Grâce à la [proposition 3.25](#) et au [théorème 3.27](#) auxquels on ajoute l'argument des dimensions, on peut prouver les deux corollaires suivants :

Corollaire 3.28 ([Foi11, Théorème 2]). *Soit A une bigèbre bidendriforme. Alors A est générée librement en tant qu'algèbre dendriforme par $\text{TPrim}(A)$.*

Corollaire 3.29 ([Foi11, Foi02a, Foi02b]). *Si A est une bigèbre bidendriforme alors A est isomorphe à $\mathcal{H}_{P,R}^{\mathcal{D}}$ où la décoration \mathcal{D} est une base de $\text{TPrim}(A)$.*

Nous utiliserons la base (e_f) , décrite dans la [section 3.2.3](#), où le produit est le produit de greffe et le coproduit est la déconcaténation des forêts.

Ces deux corollaires signifient la chose suivante : si les séries de Hilbert des espaces totalements primitifs de deux algèbres sont égales, alors ces deux algèbres sont isomorphes. En particulier, si une algèbre est une bigèbre bidendriforme, alors elle est auto-duale. Enfin, d'après le [théorème 3.24](#), nous avons le corollaire suivant :

Corollaire 3.30. $((\mathbf{WQSym})_+, \overline{\sqcup}, \overline{\sum}, \Delta_{\overline{\sqcup}}, \Delta_{\overline{\sum}})$ et $((\mathbf{WQSym}^*)_+, \overline{\prec}, \overline{\succ}, \Delta_{\overline{\prec}}, \Delta_{\overline{\succ}})$ sont isomorphes.

Autrement dit, **WQSym** est auto-duale.

Deuxième partie

Automorphism of \mathbf{WQSym}

This part is coming from our article [Mlo22a], chapters 4, 5 and 6 correspond to sections 2, 3 and 4 of the article. The main theorem of this thesis is an explicite bidendriform isomorphism between **WQSym** and its dual.

In order to compute this isomorphism, we construct two basis, \mathbb{O} for **WQSym** and \mathbb{P} for **WQSym**^{*}, in which primitive elements and totally primitive elements are easy to distinguish. These two basis are indexed by so-called biplan forests that are in bijection with packed words.

From now on $\text{Prim}(\mathbf{WQSym})$ and $\text{TPrim}(\mathbf{WQSym})$ are respectively abbreviated to **Prim** and **TPrim**. Moreover, we denote homogeneous components using indices and dualization using a * in exponent as in Prim_n^* . We give the first values of the dimensions $a_n := \dim(\mathbf{WQSym}_n)$, $p_n := \dim(\mathbf{Prim}_n)$ and $t_n := \dim(\mathbf{TPrim}_n)$:

n	1	2	3	4	5	6	7	8	9	OEIS [SI20]
a_n	1	3	13	75	541	4 683	47 293	545 835	7 087 261	A000670
p_n	1	2	8	48	368	3 376	35 824	430 512	5 773 936	A095989
t_n	1	1	4	28	240	2 384	26 832	337 168	4 680 272	

TABLE 3.1 – Dimensions of homogeneous components for **WQSym**, **Prim** and **TPrim**.

Though the numbers $(t_n)_n$ are easy to obtain thanks to the relation of théorème 3.27 : $\mathcal{T} = \mathcal{A}/(1 + \mathcal{A})^2$, no combinatorial interpretation existed. The first results of this part are two different subsets of packed words that are counted by these dimensions (red-irreducible and blue-irreducible).

Chapitre 4

Combinatorics of biplane forests

In this part, we will generalize twice the construction of [Foi11], one for **WQSym** and one for its dual. This chapter is devoted to the combinatorial ingredient, that is a notion of biplane forests suitable for indexing the various bases of primitive elements. Each time, we start by decomposing packed words through global descents and removal of specific letters. We then perform those decompositions recursively, encoding the result in a forests. We hence obtain so called biplane forests, which are in bijection with packed words. Later, the recursive structure of forests will be understood as a chaining of brace and dendriform operations generating some elements of **WQSym** or its dual. This will allow us to construct two bases of respectively **TPrim** and its dual by characterizing a subfamily of biplane trees.

From now on, we associate the color **blue** to the primal (**WQSym**) and the color **red** to the dual (**WQSym**^{*}). We start by explaining the construction on **WQSym**^{*} (**red**) then we dualize the construction to the primal **WQSym** (**blue**).

4.1 Dual (**Red**)

For the red side **WQSym**^{*}, the decomposition of packed words is through global descents and removal of maximum values. One step of this decomposition is called the red-factorization.

4.1.1 Decomposition of packed words through maximums

In this section, we define two combinatorial operations on packed words (ϕ_I and \blacktriangleright) and the red-factorization that uses them. The unary operation ϕ_I inserts new maximums in a word in positions I . A word that cannot be factorized $u \blacktriangleright v$ in a non trivial way is called red-irreducible. Red-irreducible words will index our basis of **TPrim**^{*}.

Définition 4.1. Fix $n \in \mathbb{N}$ and $w \in \mathbf{PW}_n$. We write $m' := \max(w) + 1$. For any $p > 0$ and any subset $I \subseteq [1, \dots, n+p]$ of cardinality p , we define $\phi_I(w) := u_1 \dots u_{n+p}$ as the packed

word of length $n + p$ obtained by inserting p occurrences of the letter m' in w so that they end up in positions $i \in I$. In other words $u_i = m'$ if $i \in I$ and w is obtained from $\phi_I(w)$ by removing all occurrences of m' . Notice that $\phi_I(w)$ is only defined if $n + p \geq i_p$.

Exemple 4.2. $\phi_{2,4,7}(1232) = 1424324$ and $\phi_{1,2,3}(\epsilon) = 111$.

Notations 4.3. For the rest of this part, $I = [i_1, \dots, i_p]$ will always denote a non-empty ($p > 0$) list of increasing non-zero integers. For any integer k , $I' = I + k$ denote the list $I' = [i_1 + k, \dots, i_p + k]$. Let \mathbf{PW}_n^I denote the set of packed words of size n whose maximums are in positions $i \in I$. This way $\phi_I(w) \in \mathbf{PW}_{n+p}^{I'}$ for any $w \in \mathbf{PW}_n^I$.

Lemme 4.4. Let $n \in \mathbb{N}$ and $p > 0$, for any $I = [i_1, \dots, i_p] \subseteq [1, \dots, n+p]$ of size p , ϕ_I is a bijection from \mathbf{PW}_n to the \mathbf{PW}_{n+p}^I .

Moreover, for any $W \in \mathbf{PW}_\ell$ where $\ell > 0$ there exists a unique pair (I, w) where $I \subseteq [1 \dots \ell]$ and w is packed, such that $W = \phi_I(w)$.

The box diagram that pictures this lemma is $W = \boxed{w}^{\leftarrow I}$.

Démonstration. Let $W \in \mathbf{PW}_\ell$ with $\ell > 0$ and m the value of the maximum letter of W . Let $I = [i_1, \dots, i_p] \subseteq [1, \dots, \ell]$ be the list of the positions of m in W and let w be the word obtain by removing all occurrences of m in W , then $W = \phi_I(w)$. If $\phi_I(u) = \phi_J(v)$ then positions of maximum values are the same so $I = J$ and words obtain by removing these maximum values are also the same so $u = v$. \square

Définition 4.5. Let $u, v \in \mathbf{PW}$ with $v \neq \epsilon$. By lemme 4.4, there is a unique pair (I, v') such that $v = \phi_I(v')$. Let $I' = I + |u|$, we define $u \blacktriangleright v := \phi_{I'}(u/v')$. In other words, we remove the maximum letter of the right word, perform a left shifted concatenation and reinser the removed letters as new maximums.

Exemple 4.6. $2123 \blacktriangleright 322312 = 2123 \blacktriangleright \phi_{1,4}(2212) = \phi_{1+4,4+4}(43452212) = 4345622612$.

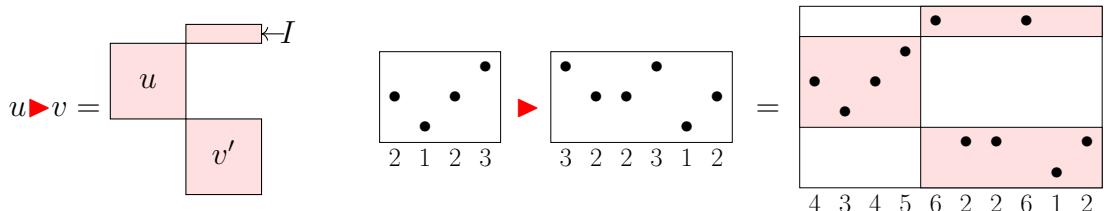


FIGURE 4.1 – Box diagrams : the operation \blacktriangleright

Lemme 4.7. Let w be an irreducible packed word. There exists a unique factorization of the form $w = u \blacktriangleright v$ which maximizes the size of u . In this factorization, let v' and I be such that $v = \phi_I(v')$, then

- either $v' = \epsilon$ and $I = [1, \dots, p]$ for some p ,
- or the global descent decomposition $v' = v_1 / \dots / v_r$ of v' satisfies the inequalities $1 \leq i_1 \leq |v_1|$, and $1 \leq (|I| + |v'|) + 1 - i_p \leq |v_r|$ with $I = [i_1, \dots, i_p]$.

We call it the **red-factorization** of a word.

Exemple 4.8. Here is a first detailed example of a red-factorization of an irreducible packed word :

Consider the irreducible packed word $w = 543462161$.

- The first step is to remove all the occurrences of the maximal value but keep in memory the positions in the initial word. We get $w' = 5434_21_1$ which is a packed word, but is not irreducible.
 - The second step is to decompose the new word w' in irreducible factors $w' = 1/212/_1/1_1$. We still keep in memory the positions of the removed value. (when we have the choice, we cut to the left of the removed value.)
 - We can distinguish two groups of factors, those strictly before the first maximum withdrawn and the others $w' = 1/212 / _1/1_1$.
 - Finally, by numbering the positions of the maximum removed value in the right factor (positions 1 and 4), we get the following decomposition of w (see [définition 4.1](#) for ϕ and [définition 4.5](#) for \blacktriangleright) :
- $$w = 543462161 = (1/212) \blacktriangleright \phi_{1,4}(1/1) = (3212) \blacktriangleright \phi_{1,4}(211) = 3212 \blacktriangleright 32131.$$

Exemple 4.9. Here are some other red-factorizations :

$$\begin{aligned} 21331 &= 1 \blacktriangleright \phi_{2,3}(11) = 1 \blacktriangleright 1221 & 1231 &= \epsilon \blacktriangleright \phi_3(121) = \epsilon \blacktriangleright 1231 \\ 1233 &= 12 \blacktriangleright \phi_{1,2}(\epsilon) = 12 \blacktriangleright 11 & 111 &= \epsilon \blacktriangleright \phi_{1,2,3}(\epsilon) = \epsilon \blacktriangleright 111 \\ 56434126 &= 1 \blacktriangleright \phi_{1,7}(212/12) = 1 \blacktriangleright \phi_{1,7}(43412) = 1 \blacktriangleright 5434125 \end{aligned}$$

Démonstration. Let w be irreducible and let (I, w') be the unique pair such that $w = \phi_I(w')$ according to [lemme 4.4](#). By [lemme 1.45](#), we write $w' = w'_1/w'_2/\dots/w'_k$, the unique decomposition into irreducibles. Let ℓ be such that w'_ℓ is the last factor which is entirely before the first removed maximum, it is the only choice to maximize the size of u . Then with $r = k - \ell$ we can rewrite w' as $(u_1/\dots/u_\ell)/(v_1/\dots/v_r)$. Now we get I' by subtracting $|u_1/\dots/u_\ell|$ to all parts of I ($I' = I - |u_1/\dots/u_\ell|$) and we obtain

$$w = u_1/\dots/u_\ell \blacktriangleright \phi_{I'}(v_1/\dots/v_r)$$

with $i'_1 \leq |v_1|$ or $r = 0$.

In the case of $v' \neq \epsilon$, the inequality $(|v'| + |I|) + 1 - i_p \leq |v_r|$ is always true otherwise w would not be irreducible. \square

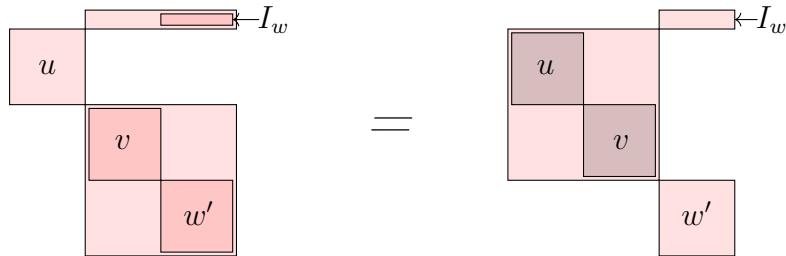
Définition 4.10. A packed word w is said to be **red-irreducible** if w is irreducible and the equality $w = u \blacktriangleright v$ implies that $u = \epsilon$ (and $w = v$).

Here are all red-irreducible packed words of size 1, 2, 3 and 4 in lexicographic order :

	1,	11,	111	121	132	212,
1111	1121	1132	1211	1212	1221	1231
1321	1322	1323	1332	1342	1423	1432
2132	2143	2212	2312	2413	3123	3132
						3213.

Here are some useful lemmas on the operation \blacktriangleright .

Lemme 4.11. *For any $u, v, w \in \mathbf{PW}$ with $w \neq \epsilon$, we have $u \blacktriangleright (v \blacktriangleright w) = (u/v) \blacktriangleright w$.*

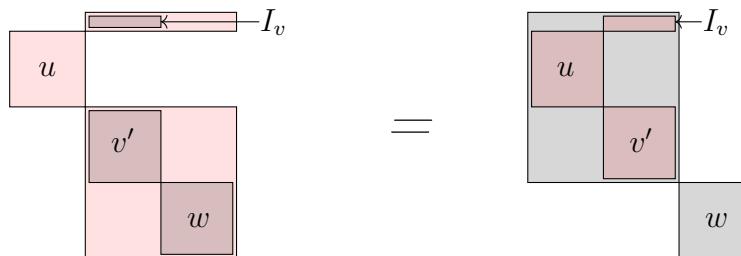


Démonstration. Let $u, v, w \in \mathbf{PW}$ with $w \neq \epsilon$ and let w' and I_w such that $w = \phi_{I_w}(w')$.

$$\begin{aligned}
u \blacktriangleright (v \blacktriangleright w) &= u \blacktriangleright (\phi_{I_w+|v|}(v/w')) \\
&= \phi_{I_w+|v|+|u|}(u/(v/w')) \\
&= \phi_{I_w+|v|+|u|}((u/v)/w') \\
&= (u/v) \blacktriangleright \phi_{I_w}(w') \\
&= (u/v) \blacktriangleright w.
\end{aligned}$$

□

Lemme 4.12. *For any $u, v, w \in \mathbf{PW}$ with $v \neq \epsilon$, we have $u \blacktriangleright (v/w) = (u \blacktriangleright v)/w$.*



Démonstration. Let $u, v, w \in \mathbf{PW}$ with $v \neq \epsilon$ and let v' and I_v such that $v = \phi_{I_v}(v')$.

$$\begin{aligned}
u \blacktriangleright (v/w) &= u \blacktriangleright \phi_{I_v}(v'/w) \\
&= \phi_{I_v+|u|}(u/(v'/w)) \\
&= \phi_{I_v+|u|}((u/v')/w) \\
&= \phi_{I_v+|u|}(u/v')/w \\
&= (u \blacktriangleright v)/w.
\end{aligned}$$

□

Remarque 4.13. Adding the associativity of shifted concatenation $u/(v/w) = (u/v)/w$, the two operations \blacktriangleright and $/$ verify relations of the *skew-duplicial operad* [BDO20].

Corollaire 4.14. For any $u, v \in \mathbf{PW}$, we have that $u \blacktriangleright v$ is irreducible if and only if v is irreducible.

Démonstration. By contradiction, if $v = v_1/v_2$ then by lemme 4.12 $u \blacktriangleright v = (u \blacktriangleright v_1)/v_2$. Now if $u \blacktriangleright v = w_1/w_2$, as the position of the first maximum of $u \blacktriangleright v$ is greater than $|u|$ we have that $w_1 = w'_1 \cdot w''_1$ such that $\text{pack}(w'_1) = u$. We also have that $\text{pack}(w''_1)/w_2 = v$. \square

Proposition 4.15. For any word w , $w = u \blacktriangleright v$ is the red-factorization of w if and only if v is red-irreducible.

Démonstration. Let $w \in \mathbf{PW}$ and let $u \blacktriangleright v$ be the red-factorization of w . Let v_1 and v_2 such that $v = v_1 \blacktriangleright v_2$, then $(u/v_1) \blacktriangleright v_2 = w$ by lemme 4.11, but in the red-factorization the size of u is maximized so $|(u/v_1)| \leq |u|$ and then we have that $v_1 = \epsilon$ so v is red-irreducible.

Let $w \in \mathbf{PW}$ and let u and v such that $w = u \blacktriangleright v$ and v is red-irreducible. By contradiction, suppose that there exists u', v' such that $w = u' \blacktriangleright v'$ with $|u| < |u'|$ and $v' \neq \epsilon$. Then necessarily u is a prefix of u' . Let u'' such that $u' = u \cdot u''$, then $\text{pack}(u'') \blacktriangleright v' = v$. But v is red-irreducible. So the size of u is maximal if v is red-irreducible. \square

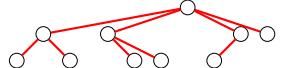
For the reader who is familiar with ordered set partitions, all the definitions in section 4.1 can be easily written with these. However in section 4.2 it is easier to do all the definitions on packed words and in chapitre 6 we must have the same object on both sides to explicit the isomorphism. So we decided to stick to packed words.

4.1.2 Red-forests from decomposed packed words using ϕ

We now apply recursively the red-factorization of the previous section to construct a bijection between packed words and a certain kind of trees that we now define.

Définition 4.16. An unlabeled **biplane tree** is an ordered tree (sometimes also called planar) whose children are organized in a pair of two (possibly empty) ordered forests, which we call the left and right forests, a forest being an ordered list of trees.

In the picture, we naturally draw the children of the left (resp. right) forest on the left (resp. right) of their father.

Exemple 4.17. The biplane trees  ,  and  are different. Indeed in the first case, the left forest contains two trees and the right forest is empty, in the second case both forests contain exactly one tree while in the third case the left forest is empty and the right contains two trees. Here is an example of a bigger biplane tree where the root has two trees in both left and right forests .

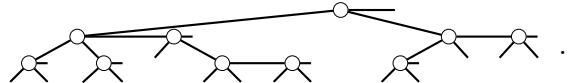
Définition 4.18. A **skeleton biplane tree** is a biplane tree where no node has a right forest.

These skeleton biplane trees can also be seen as planar trees. In [Foi11] we have planar trees recursively labeled by planar trees. Skeleton biplane trees are similar to these planar trees, we prefer to see them as biplane tree with no right forest in order to keep some consistency.

Définition 4.19. The **size** of a biplane tree is the number of node in the tree.

Remarque 4.20. Biplane forests \mathfrak{F} (*i.e.* ordered list of biplane trees \mathfrak{T}) are counted by the sequence $A001764$ in OEIS [SI20] whose explicit formula is $a(n) = \binom{3n}{n}/(2n+1)$. Biplane forests are in bijection with ternary trees. The bijection is the following, in a biplane forest a node has a first left child and a first right child and a right brother. A consequence is that unlabeled biplane trees are counted by the sequence $A006013$ in OEIS [SI20] whose explicit formula is $a(n) = \binom{3n+1}{n}/(n+1)$. Indeed, biplane trees are in bijection with pair of ternary trees. Here is an example of the ternary tree in bijection with the biplane forest constituted

of one tree, the big biplane tree in [exemple 4.17](#)



n	0	1	2	3	4	5	6	7	8	OEIS [SI20]
\mathfrak{F}_n	1	1	3	12	55	273	1 428	7 752	43 263	A001764
\mathfrak{T}_n	0	1	2	7	30	143	728	3 876	21 318	A006013

TABLE 4.1 – Number of biplane forests and biplane trees.

Remarque 4.21. As we can see on OEIS [SI20], sequence $A006013$ which counts unlabeled biplane trees is the dimensions of the free L-algebra on one generator (see [Ler07]). It would be interesting to investigate the link between L -algebras and bidendriform bialgebras using biplane trees.

In our construction we will deal with labeled biplane trees with colored edges. For a labeled biplane tree, we denote by $\text{Node}_{\text{R}}(x, f_\ell, f_r)$ the tree whose edges are colored in red, root is labeled by x and whose left (resp. right) forest is given by f_ℓ (resp. f_r). We also denote by $[t_1, \dots, t_k]$ a forest of k trees. The edge color (for now, only red) will play a role later in this part.

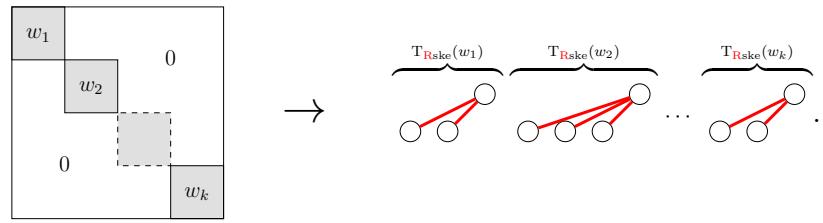
Exemple 4.22.

$$\text{Node}_{\text{R}}((1), [], []) = \textcircled{1}, \text{ and } \text{Node}_{\text{R}}((1, 3), [], [\text{Node}_{\text{R}}((1), [], [])]) = \textcircled{1} \text{---} \textcircled{1}^{(1, 3)}.$$

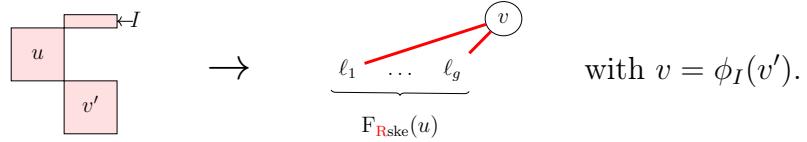
We now apply recursively the global descent decomposition and the red-factorization of [lemmes 1.45 et 4.7](#). We obtain an algorithm which takes a packed word and returns a biplane forest where nodes are decorated by red-irreducible packed words :

Définition 4.23. We now define two functions F_{Rske} and T_{Rske} . These functions transform respectively a packed word and an irreducible packed word into respectively a skeleton biplane forest and a skeleton biplane tree labeled by red-irreducible words. These functions are defined in a mutual recursive way as follows :

- $F_{Rske}(\epsilon) = []$ (empty forest),
- for any packed word w , let $w_1/w_2/\dots/w_k$ be the global descent decomposition of w , then $F_{Rske}(w) := [T_{Rske}(w_1), T_{Rske}(w_2), \dots, T_{Rske}(w_k)]$.

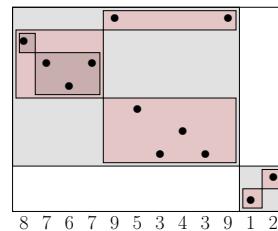


- for any irreducible packed word w , let $w = u \blacktriangleright v$ be the red-factorization of w . We define $T_{Rske}(w) := \text{Node}_R(v, F_{Rske}(u), [])$.



Exemple 4.24. Let $w = 876795343912$, the global descent decomposition of lemme 1.45 gives $w = w_1/w_2$ with $w_1 = 6545731217$ and $w_2 = 12$. Now, we have the red-factorization of w_1 and w_2 using lemme 4.7 as

$$w_1 = 3212 \blacktriangleright \phi_{1,6}(3121) = (1/212) \blacktriangleright 431214 \quad \text{and} \quad w_2 = 1 \blacktriangleright \phi_1(\epsilon) = 1 \blacktriangleright 1.$$



It gives the following forest :

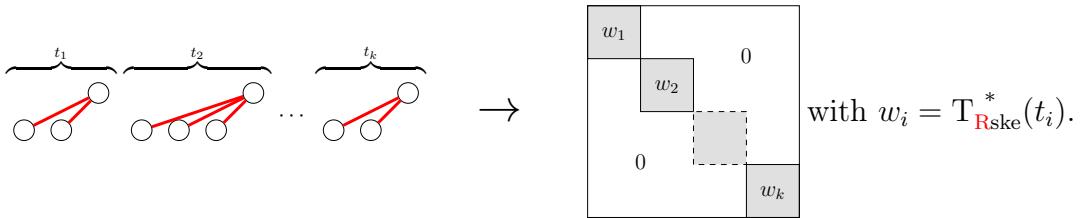
$$\begin{aligned} F_{Rske}(876795343912) &= [T_{Rske}(6545731217), T_{Rske}(12)] \\ &= F_{Rske}(3212) \quad 431214 \quad 1 \\ &= \text{Node}_R(431214, F_{Rske}(3212), \emptyset) \quad \text{Node}_R(1, F_{Rske}(1), \emptyset) \\ &= \text{Node}_R(431214, \text{Node}_R(212, F_{Rske}(1), \emptyset), \text{Node}_R(1, F_{Rske}(1), \emptyset)). \end{aligned}$$

Définition 4.25. A labeled biplane forest (resp. tree) is a **red-skeleton forest** (resp. **tree**) if it is labeled by red-irreducible words and no node has a right child.

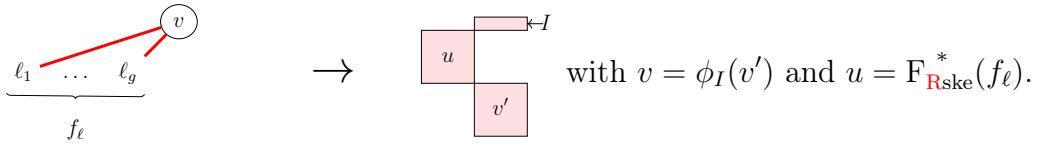
We want to prove that the functions F_{Rske} and T_{Rske} are bijections. To do that we first define two functions that are the inverses.

Définition 4.26. We now define two functions F_{Rske}^* and T_{Rske}^* that transform respectively a red-skeleton forest and tree into packed words. These functions are defined in a mutual recursive way as follows :

- $F_{\text{Rske}}^*([]) = \epsilon$,
- for any red-skeleton forest $f = [t_1, \dots, t_k]$, we define
 $F_{\text{Rske}}^*(f) := T_{\text{Rske}}^*(t_1) / \dots / T_{\text{Rske}}^*(t_k)$.



- for any red-skeleton tree $t = \text{Node}_{\text{R}}(v, f_\ell, []),$ we define
 $T_{\text{Rske}}^*(t) := F_{\text{Rske}}^*(f_\ell) \blacktriangleright v$.



Lemme 4.27. The functions F_{Rske} and F_{Rske}^* (resp. T_{Rske} and T_{Rske}^*) are two inverse bijections between packed words and red-skeleton forests (resp. irreducible packed words and red-skeleton trees). That is to say $F_{\text{Rske}}^{-1} = F_{\text{Rske}}^*$ and $T_{\text{Rske}}^{-1} = T_{\text{Rske}}^*$.

Démonstration. We start to prove that domain and codomain are as announced (see Items (a) and (b) bellow), then we prove that the functions F_{Rske} and F_{Rske}^* (resp. T_{Rske} and T_{Rske}^*) are inverse to each other (see Items (c) and (d) bellow).

(a) By [définition 4.23](#), a forest (resp. tree) obtain by F_{Rske} (resp. T_{Rske}) is a red-skeleton forest (resp. tree). Indeed, thanks to [proposition 4.15](#) nodes are labeled by red-irreducible words because a red-factorization is done and nodes have no right children.

(b) We prove by a mutual induction that F_{Rske}^* returns a packed word and that T_{Rske}^* returns an irreducible packed word. Indeed, we do an induction on the size of the forest or the tree. Here is our induction hypothesis for $n \in \mathbb{N}$:

$$\begin{aligned} \forall t, \text{red-skeleton tree of size } \leq n, T_{\text{Rske}}^*(t) & \text{ is an irreducible packed word,} \\ \forall f, \text{red-skeleton forest of size } \leq n, F_{\text{Rske}}^*(f) & \text{ is a packed word.} \end{aligned} \tag{4.1}$$

The base case ($n = 0$) is given by the first item of [définition 4.26](#) (*i.e.* $F_{\text{Rske}}^*(\text{[]}) = \epsilon$ the empty packed word).

Now, let us fix $n \geq 1$ and suppose that the hypothesis (4.1) holds. Let $f = [t_1, \dots, t_k]$ be a red-skeleton forest of size $n + 1$.

- If $k = 1$, then f is reduced to a single tree t . We need to prove that $T_{\text{Rske}}^*(t)$ is an irreducible packed word (which also gives that $F_{\text{Rske}}^*(f)$ is a packed word as in this case $F_{\text{Rske}}^*(f) = T_{\text{Rske}}^*(t)$). Let $t = \text{Node}_{\text{R}}(v, f_\ell, \text{[]})$ be a red-skeleton tree of size $n + 1$ (notice that the word v can be of any size). The size of f_ℓ is n , so by induction $F_{\text{Rske}}^*(f_\ell)$ is a packed word, as v is red-irreducible it is by definition irreducible so by [corollaire 4.14](#) $T_{\text{Rske}}^*(t) = F_{\text{Rske}}^*(f_\ell) \blacktriangleright v$ is irreducible.
- If $k \geq 2$, *i.e.*, the forest contains at least two trees, since all trees are of size at least one, t_1, \dots, t_k are at most of size n , so we have by induction that $T_{\text{Rske}}^*(t_1), \dots, T_{\text{Rske}}^*(t_k)$ are irreducible packed words. $F_{\text{Rske}}^*(f)$ is the shifted concatenation of $T_{\text{Rske}}^*(t_1), \dots, T_{\text{Rske}}^*(t_k)$ and thus it is a packed word.

(c) We now prove by a mutual induction on the size of the forest or the tree that, for $n \in \mathbb{N}$:

$$\begin{aligned} \forall t, \text{red-skeleton tree of size } \leq n, T_{\text{Rske}}(T_{\text{Rske}}^*(t)) &= t, \\ \forall f, \text{red-skeleton forest of size } \leq n, F_{\text{Rske}}(F_{\text{Rske}}^*(f)) &= f. \end{aligned} \quad (4.2)$$

The base case ($n = 0$) is given by the first item of [définitions 4.23 et 4.26](#) as $F_{\text{Rske}}(F_{\text{Rske}}^*(\text{[]})) = F_{\text{Rske}}(\epsilon) = \text{[]}$.

Now let us fix $n \geq 1$ and suppose that the hypothesis (4.2) holds. Let $f = [t_1, \dots, t_k]$ be a red-skeleton forest of size $n + 1$.

- If $k = 1$, then the forest f is reduced to a single tree t , then it is sufficient to prove $T_{\text{Rske}}(T_{\text{Rske}}^*(t)) = t$ (as in this case $F_{\text{Rske}}^*(f) = T_{\text{Rske}}^*(t)$). Let $t = \text{Node}_{\text{R}}(v, f_\ell, \text{[]})$ a red-skeleton tree of size $n + 1$. As the label v is a red-irreducible packed word, with the induction hypothesis on $F_{\text{Rske}}^*(f_\ell)$ and with [proposition 4.15](#), $F_{\text{Rske}}^*(f_\ell) \blacktriangleright v$ is the red-factorization so :

$$\begin{aligned} T_{\text{Rske}}(T_{\text{Rske}}^*(t)) &= T_{\text{Rske}}(F_{\text{Rske}}^*(f_\ell) \blacktriangleright v) \\ &= \text{Node}_{\text{R}}(v, F_{\text{Rske}}(F_{\text{Rske}}^*(f_\ell)), \text{[]}) \\ &= \text{Node}_{\text{R}}(v, f_\ell, \text{[]}) = t. \end{aligned}$$

- If $k \geq 2$, since all trees are of size at least one, they are at most of size n , so we have by induction that :

$$\begin{aligned} F_{\text{Rske}}(F_{\text{Rske}}^*(f)) &= F_{\text{Rske}}(T_{\text{Rske}}^*(t_1) / \dots / T_{\text{Rske}}^*(t_k)) \\ &\quad \text{as } T_{\text{Rske}}^*(t_i) \text{ are irreducible packed words} \\ &= [T_{\text{Rske}}(T_{\text{Rske}}^*(t_1)), \dots, T_{\text{Rske}}(T_{\text{Rske}}^*(t_k))] \\ &= [t_1, \dots, t_k] = f. \end{aligned}$$

(d) Finally we prove by a mutual induction on the size of the word w that, for $n \in \mathbb{N}$:

$$\begin{aligned} \forall v \in \mathbf{PW}, \text{irreducible packed word of size } \leq n, T_{\text{Rske}}^*(T_{\text{Rske}}(v)) &= v, \\ \forall w \in \mathbf{PW}, \text{packed word of size } \leq n, F_{\text{Rske}}^*(F_{\text{Rske}}(w)) &= w. \end{aligned} \quad (4.3)$$

The base case ($n = 0$) is given by the first item of [définitions 4.23 et 4.26](#) as $F_{Rske}^*(F_{Rske}(\epsilon)) = F_{Rske}^*([]) = \epsilon$.

Now let us fix $n \geq 1$ and suppose that the hypothesis (4.3) holds. Let $w \in PW_{n+1}$ a packed word of size $n + 1$. Let $w = w_1/w_2/\dots/w_k$ be the global descent decomposition of w .

- If $k = 1$, the packed word w is irreducible then $F_{Rske}(w) = [T_{Rske}(w)]$ so we need to prove that $T_{Rske}^*(T_{Rske}(w)) = w$. Let $w = u \blacktriangleright v$ be the red-factorization of w , then we can use the induction hypothesis on u , indeed as v is not empty the size of u is smaller than n :

$$\begin{aligned} T_{Rske}^*(T_{Rske}(w)) &= T_{Rske}^*(\text{Node}_R(v, F_{Rske}(u), [])) \\ &= F_{Rske}^*(F_{Rske}(u)) \blacktriangleright v \\ &= u \blacktriangleright v = w. \end{aligned}$$

- If $k \geq 2$, then we use the induction hypothesis on each factors, so we have :

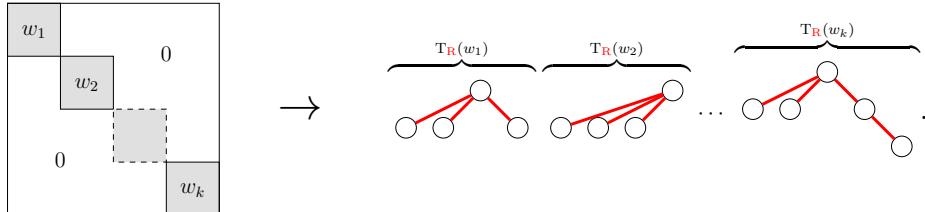
$$\begin{aligned} F_{Rske}^*(F_{Rske}(w)) &= F_{Rske}^*([T_{Rske}(w_1), \dots, T_{Rske}(w_k)]) \\ &= T_{Rske}^*(T_{Rske}(w_1)) / \dots / T_{Rske}^*(T_{Rske}(w_k)) \\ &= w_1 / \dots / w_k = w. \end{aligned} \quad \square$$

Now that we have the red-skeleton, we will add right forests to every nodes to obtain biplane trees. For every node, if v is the red-irreducible word in label, with $\phi_I(v') = v$, then I is the new label and $F_R(v')$ is the new right forest.

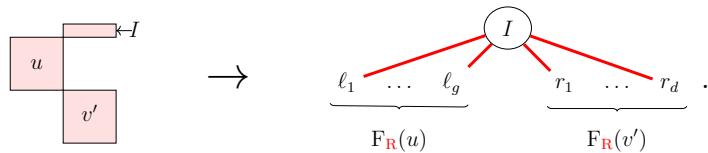
Here is the formal definition of $F_R(w)$ and $T_R(w)$ which are very similar to [définition 4.23](#), only the third item is different. The labels are now lists of integers.

Définition 4.28. The forest $F_R(w)$ (resp. tree $T_R(w)$) associated to a packed word (resp. irreducible packed word) w are defined in a mutual recursive way as follows :

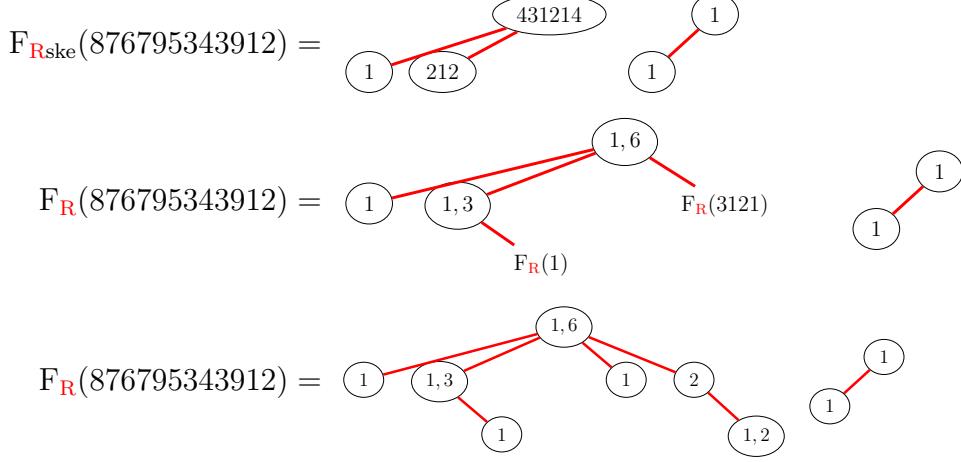
- $F_R(\epsilon) = []$ (empty forest),
- for any packed word w , let $w_1/w_2/\dots/w_k$ be the global descent decomposition of w , then $F_R(w) := [T_R(w_1), T_R(w_2), \dots, T_R(w_k)]$.



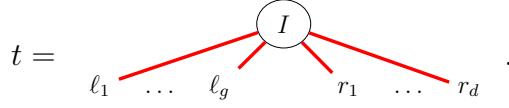
- for any irreducible packed word w , we define $T_R(w) := \text{Node}_R(I, F_R(u), F_R(v'))$ where $w = u \blacktriangleright \phi_I(v')$ is the red-factorization of w .



Exemple 4.29. Consider again $w = 876795343912$. We start from the red-skeleton forest from [exemple 4.24](#).



Définition 4.30. Let t be a labeled biplane tree. We write $t = \text{Node}_R(I, f_\ell, f_r)$ where $I = [i_1, \dots, i_p]$, $p > 0$, $(1 \leq i_1 < \dots < i_p)$, $f_\ell = [\ell_1, \dots, \ell_g]$ and $f_r = [r_1, \dots, r_d]$, which is depicted as follows :



The **weight** of t is recursively defined by $\omega(t) = p + \sum_{i=0}^g \omega(\ell_i) + \sum_{j=0}^d \omega(r_j)$. In particular, if t is a single node then $\omega(t) = p$. By extension, the **weight** of a forest is the sum of the weight of its trees.

Lemme 4.31. *The weight of a forest (resp. a tree) obtained by the functions F_R (resp. T_R) is equal to the size of the word, i.e. For all $w \in \mathbf{PW}$ then $\omega(F_R(w)) = |w|$ and for all $w \in \mathbf{PW}$ with w irreducible then $\omega(T_R(w)) = |w|$.*

Démonstration. We prove by induction with the following hypothesis, for $n \in \mathbb{N}$:

$$\begin{aligned} \forall w \in \mathbf{PW}_n, \quad \omega(F_R(w)) &= |w|, \\ \forall w \in \mathbf{PW}_n \text{ with } w \text{ irreducible}, \quad \omega(T_R(w)) &= |w|. \end{aligned} \tag{4.4}$$

The base case is given by the first item of [définition 4.28](#) as $F_R(\epsilon) = []$ and $\omega([]) = |\epsilon| = 0$. Let us fix $n \geq 1$ and suppose that the hypothesis (4.4) holds. Let $w \in \mathbf{PW}_{n+1}$ and $w = w_1/w_2/\dots/w_k$ be the global descent decomposition of w .

- If $k = 1$, we have $F_R(w) = [T_R(w)]$. Let $w = u \blacktriangleright \phi_I(v)$ with $I = [i_1, \dots, i_p]$, $p > 0$ be the red-factorization of w , then

$$\omega(T_R(w)) = \omega(\text{Node}_R(I, F_R(u), F_R(v))) = p + \omega(F_R(u)) + \omega(F_R(v)).$$

As $p > 0$, the sizes of u and v are at most n , by induction $\omega(T_R(w)) = p + |u| + |v| = |w|$.

- If $k \geq 2$, by induction on each factors, we have that

$$\omega(F_R(w)) = \omega(T_R(w_1)) + \dots + \omega(T_R(w_k)) = |w_1| + \dots + |w_k| = |w|. \quad \square$$

Définition 4.32. Using the same notations as in previous [définition 4.30](#), we say that t is a **red-packed tree** if it satisfies :

$$\left\{ \begin{array}{l} d = 0, \\ i_k = k \text{ for all } k \leq p, \\ \ell_1, \dots, \ell_g \text{ are red-packed trees.} \end{array} \right. \quad \text{or} \quad \left\{ \begin{array}{l} d \geq 1, \\ 1 \leq i_1 \leq \omega(r_1), \\ 1 \leq p + \omega(f_r) + 1 - i_p \leq \omega(r_d), \\ \ell_1, \dots, \ell_g \text{ and } r_1, \dots, r_d \text{ are red-packed trees.} \end{array} \right.$$

An ordered list of red-packed trees is a **red-packed forest**.

Remarque 4.33. Red-skeleton trees can be interpreted as flattened representations of red-packed trees. Symmetrically, red-packed trees can be interpreted as unfolded representations of red-skeleton trees. We use the operation ϕ_I to change between red-packed and red-skeleton trees.

Notations 4.34. From now on, we use these notations :

- \mathfrak{F}_{Rn} the set of red-packed forests of weight n , ($\mathfrak{F}_{Rn} := \{\mathbf{F}_R(w)\}_{w \in \mathbf{PW}_n}$),
- \mathfrak{T}_{Rn} the set of red-packed trees of weight n , ($\mathfrak{T}_{Rn} := \{\mathbf{T}_R(w)\}_{w \in \mathbf{PW}_n}$ with w irreducible),
- \mathfrak{N}_{Rn} the set of red-packed trees of weight n such that the left forest of the root is empty, ($\mathfrak{N}_{Rn} := \{\mathbf{T}_R(w)\}_{w \in \mathbf{PW}_n}$ with w red-irreducible). In particular, the red-skeleton of a tree of \mathfrak{N}_{Rn} consist of a single node labeled by a red-irreducible word.

Remarque 4.35. The set \mathfrak{N}_{Rn} can be described as a disjointed union of sets depending on $I = [i_1, \dots, i_p]$. Let \mathfrak{F}_{Rn}^I denote the set of red-packed forests of weight n that can be right children of a node labeled by I (see [définition 4.32](#) for conditions), we have the following description :

$$\mathfrak{N}_{Rn} = \bigsqcup_I \{\mathbf{Node}_R(I, [], f_r) \mid f_r \in \mathfrak{F}_{Rn-p}^I\}. \quad (4.5)$$

Analogously, we use \mathfrak{F}_{Rske} , \mathfrak{T}_{Rske} and \mathfrak{N}_{Rske} for red-skeleton forests, trees and trees with only one node.

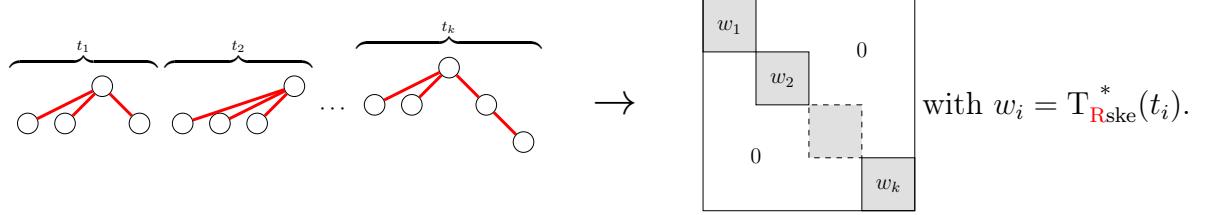
We can remark that for $n = 1$ we have $\mathfrak{N}_{R1} = \mathfrak{T}_{R1} = \mathfrak{F}_{R1}$ and $\forall n > 1, \mathfrak{N}_{Rn} \subsetneq \mathfrak{T}_{Rn} \subsetneq \mathfrak{F}_{Rn}$.

As with [définition 4.26](#), we want to prove that the functions \mathbf{F}_R and \mathbf{T}_R are bijections. To do that we first define the two inverse functions.

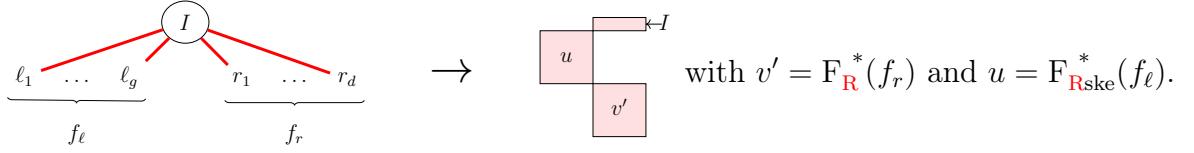
Définition 4.36. We define here the functions \mathbf{F}_R^* (resp. \mathbf{T}_R^*) that transform a red-packed forest f (resp. tree t) into a packed word. We reverse all instructions of [définition 4.28](#) as follows :

- $\mathbf{F}_R^*([]) = \epsilon$,

- for any non empty red-packed forest $f = [t_1, t_2 \dots, t_k]$, then $F_R^*(f) = T_R^*(t_1)/T_R^*(t_2)/\dots/T_R^*(t_k)$.



- for any non empty red-packed tree $t = \text{Node}_R(I, f_\ell, f_r)$, then $T_R^*(t) = F_R^*(f_\ell) \blacktriangleright \phi_I(F_R^*(f_r))$.



There might be a problem with this definition since $\phi_I(F_R^*(f_r))$ is only defined if $i_p \leq |F_R^*(f_r)| + p$ (see [définition 4.1](#)). We prove in the following [lemme 4.37](#) that the inequality holds if $t \in \mathfrak{T}_R$.

Lemme 4.37. *For any red-packed forest f , $F_R^*(f)$ is a well defined word of size $\omega(f)$. For any red-packed tree t , $T_R^*(t)$ is a well defined word of size $\omega(t)$.*

Démonstration. We prove by induction with the following hypothesis, for $n \in \mathbb{N}$:

$$\begin{aligned} \forall f \in \mathfrak{F}_{R \leq n}, \quad & F_R^*(f) \text{ is well defined and } |F_R^*(f)| = \omega(f), \\ \forall t \in \mathfrak{T}_{R \leq n}, \quad & T_R^*(t) \text{ is well defined and } |T_R^*(t)| = \omega(t). \end{aligned} \tag{4.6}$$

The base case is given by the first item of [définition 4.36](#) as $F_R^*([]) = \epsilon$ and $\omega([]) = |\epsilon| = 0$.

Let us fix $n \geq 1$ and suppose that the hypothesis (4.6) holds. Let $f = [t_1, \dots, t_k] \in \mathfrak{F}_{Rn+1}$.

• If $k = 1$, it is sufficient to prove the second item of (4.6). Let $t = \text{Node}_R(I, f_\ell, f_r) \in \mathfrak{T}_{Rn+1}$. According to [définition 4.32](#) with notations of [définition 4.30](#) there are two cases :

• $d = 0$ and $I = [1, \dots, p]$. We have that $i_p = p$ and $|F_R^*(f_r)| = |\epsilon| = 0$ so $i_p \leq 0 + p$ and $\phi_I(\epsilon) = 11\dots11$ of size p . Now by induction on f_ℓ , we have that $F_R^*(f_\ell)$ is a well defined word of size $\omega(f_\ell)$. Finally $T_R^*(t) = F_R^*(f_\ell) \blacktriangleright \phi_I(\epsilon) = F_R^*(f_\ell) \setminus \phi_I(\epsilon)$ is a well defined word of size $|F_R^*(f_\ell)| + p = \omega(t)$.

• $d \geq 1$. As $p > 0$ we can apply the hypothesis (4.6) on f_r and f_ℓ . According to [définition 4.32](#) we have that

$$\begin{aligned} 1 &\leq p + \omega(f_r) + 1 - i_p, \\ i_p &\leq p + |F_R^*(f_r)|. \end{aligned}$$

So $T_R^*(t)$ is well defined. Moreover

$$\begin{aligned} |T_R^*(t)| &= |F_R^*(f_\ell) \blacktriangleright \phi_I F_R^*(f_r)| \\ &= |F_R^*(f_\ell)| + p + |F_R^*(f_r)| \\ &= \omega(f_\ell) + p + \omega(f_r) = \omega(t). \end{aligned}$$

- If $k \geq 2$, the weight of trees are at least 1 so we can apply (4.6) on trees of f . \square

Théorème 4.38. *The functions F_R and F_R^* (resp. T_R and T_R^*) are two inverse bijections between packed words of size n and red-packed forests (resp. irreducible packed words and red-packed trees) of weight n . That is to say $F_R^{-1} = F_R^*$ and $T_R^{-1} = T_R^*$.*

Démonstration. The proof is very similar to the one of lemme 4.27. Indeed, we start to prove that domain and codomain are as announced (see Items (a) and (b) bellow), then we prove that the functions F_R and F_R^* (resp. T_R and T_R^*) are inverse to each other (see Items (c) and (d) bellow).

We now give the differences with the proof of lemme 4.27 and we advise the reader to read the two proofs in parallel. While for Item (a) in lemme 4.27 it was simple, we need to do an induction here to prove that conditions on labels are respected. For Items (b), (c) and (d), the same inductions are done with one additional argument, so only the different argument of the induction is explicitated here.

(a) We prove by a mutual induction that F_R returns a red-packed forest and that T_R returns a red-packed tree. Indeed, we do an induction on the size of the word w . Here is our induction hypothesis for $n \in \mathbb{N}$:

$$\begin{aligned} \forall v \in \mathbf{PW}, \text{ irreducible packed word of size } \leq n, \quad T_R(v) \text{ is a red-packed tree,} \\ \forall w, \text{ packed word of size } \leq n, \quad F_R(w) \text{ is a red-packed forest.} \end{aligned} \quad (4.7)$$

The base case ($n = 0$) is given by the first item of définition 4.28.

Now let us fix $n \geq 1$ and suppose that the hypothesis (4.7) holds. Let $w \in \mathbf{PW}$ be a packed word of size $n + 1$ and let $w_1 / \dots / w_k$ be the global descent decomposition of w .

• If $k = 1$ (w is irreducible), then $F_R(w)$ is reduced to a single tree $T_R(w)$. We need to prove that $T_R(w)$ is a red-packed tree (which also gives that $F_R(w)$ is a red-packed forest). Let $w = u \blacktriangleright v$ be the red-factorization of w . With $\phi_I(v') = v$, $f_\ell = F_R(u)$ and $f_r = F_R(v')$ we have that $T_R(w) = \text{Node}_R(I, f_\ell, f_r)$. The inequalities on I and v' in lemme 4.7 are the same as the inequalities on I and f_r in définition 4.32. Therefore by lemme 4.31 and (4.7) on f_r , we have that $T_R(w)$ belongs to \mathfrak{T}_R .

• If $k \geq 2$, the hypothesis (4.7) can be applied to each factors.

(b) Compared to the proof of lemme 4.27 we use the same general arguments to prove that F_R^* and F_{Rske}^* return a packed word. First of all the base case and the case were the size of the forest is $k \geq 2$ are dealt with by a similar argumentation. It remains to prove that T_R^* returns an irreducible packed word. We thus suppose that the induction hypothesis (4.8) holds for a given $n \in \mathbb{N}$:

$$\begin{aligned} \forall t, \text{ red-packed tree of size } \leq n, \quad T_R^*(t) \text{ is an irreducible packed word,} \\ \forall f, \text{ red-packed forest of size } \leq n, \quad F_R^*(f) \text{ is a packed word.} \end{aligned} \quad (4.8)$$

Let $t = \text{Node}_{\mathbf{R}}(I, f_\ell, f_r)$ be a red-packed tree of size $n + 1$. By induction we have that $\mathbf{F}_{\mathbf{R}}^*(f_\ell)$ and $\mathbf{F}_{\mathbf{R}}^*(f_r)$ are packed words. Moreover either $f_r = \epsilon$ and $I = [1, \dots, p]$ or $1 \leq p + \omega(f_r) + 1 - i_p \leq \omega(r_d)$ with $I = [i_1, \dots, i_p]$ and $f_r = [r_1, \dots, r_d]$. In both cases, $\phi_I(\mathbf{F}_{\mathbf{R}}^*(f_r))$ is a red-irreducible packed word. Indeed, we recognize the two cases of [définition 4.10](#) with the same inequalities. Finally $\mathbf{T}_{\mathbf{R}}^*(t) = \mathbf{F}_{\mathbf{R}}^*(f_\ell) \blacktriangleright \phi_I(\mathbf{F}_{\mathbf{R}}^*(f_r))$ is an irreducible packed word according to [corollaire 4.14](#).

(c) We now want to prove that for any forest f (resp. t), $\mathbf{F}_{\mathbf{R}}(\mathbf{F}_{\mathbf{R}}^*(f)) = f$ (resp. $\mathbf{T}_{\mathbf{R}}(\mathbf{T}_{\mathbf{R}}^*(t)) = t$). As in Item (b), the arguments are the same as is the proof of [lemme 4.27](#) for $\mathbf{F}_{\mathbf{R}}$ and $\mathbf{F}_{\mathbf{Rske}}$. In the case of $\mathbf{T}_{\mathbf{R}}$ the new arguments are the same as in Item (b) (*i.e.* $\phi_I(\mathbf{F}_{\mathbf{R}}^*(f_r))$ is a red-irreducible packed word).

(d) Finally, we want to prove that for any packed word w (resp. irreducible packed word v), $\mathbf{F}_{\mathbf{R}}^*(\mathbf{F}_{\mathbf{R}}(w)) = w$ (resp. $\mathbf{T}_{\mathbf{R}}^*(\mathbf{T}_{\mathbf{R}}(v)) = v$). Once again, the only difference with [lemme 4.27](#) is the former second case. It remains to prove that point and we thus suppose that the induction hypothesis (4.9) holds for $n \in \mathbb{N}$:

$$\begin{aligned} \forall v \in \mathbf{PW}, \text{ irreducible packed word of size } \leq n, \quad & \mathbf{T}_{\mathbf{R}}^*(\mathbf{T}_{\mathbf{R}}(v)) = v, \\ \forall w \in \mathbf{PW}, \text{ packed word of size } \leq n, \quad & \mathbf{F}_{\mathbf{R}}^*(\mathbf{F}_{\mathbf{R}}(w)) = w. \end{aligned} \quad (4.9)$$

Let v be an irreducible packed word of size $n+1$. Let $v = v' \blacktriangleright \phi_I(v'')$ be the red-factorization of v . We have by [définition 4.28](#) that $\mathbf{T}_{\mathbf{R}}(v) = \text{Node}_{\mathbf{R}}(I, \mathbf{F}_{\mathbf{R}}(v'), \mathbf{F}_{\mathbf{R}}(v''))$. As $|I| > 0$, the sizes of v' and v'' are smaller than n so we can apply (4.9). We have :

$$\begin{aligned} \mathbf{T}_{\mathbf{R}}^*(\mathbf{T}_{\mathbf{R}}(v)) &= \mathbf{T}_{\mathbf{R}}^*(\text{Node}_{\mathbf{R}}(I, \mathbf{F}_{\mathbf{R}}(v'), \mathbf{F}_{\mathbf{R}}(v''))) \\ &= \mathbf{F}_{\mathbf{R}}^*(\mathbf{F}_{\mathbf{R}}(v')) \blacktriangleright \phi_I(\mathbf{F}_{\mathbf{R}}^*(\mathbf{F}_{\mathbf{R}}(v''))) \\ &= v' \blacktriangleright \phi_I(v'') = v. \end{aligned}$$

□

Exemple 4.39. There is a unique forest in $\mathfrak{F}_{\mathbf{R}1}$, namely $\circled{1}$, here are the 3 forests of $\mathfrak{F}_{\mathbf{R}2}$ with the associated packed word : $\mathbf{F}_{\mathbf{R}}(12) = \circled{1} \circled{2}$, $\mathbf{F}_{\mathbf{R}}(21) = \circled{2} \circled{1}$, $\mathbf{F}_{\mathbf{R}}(11) = \circled{1,2}$. We show below the 13 forests of $\mathfrak{F}_{\mathbf{R}3}$ with the corresponding packed word :

$$\begin{aligned} \mathbf{F}_{\mathbf{R}}(123) &= \circled{1} \circled{2} \circled{3}, \quad \mathbf{F}_{\mathbf{R}}(132) = \circled{1} \circled{3} \circled{2}, \quad \mathbf{F}_{\mathbf{R}}(213) = \circled{2} \circled{1} \circled{3}, \quad \mathbf{F}_{\mathbf{R}}(231) = \circled{2} \circled{3} \circled{1}, \\ \mathbf{F}_{\mathbf{R}}(312) &= \circled{3} \circled{1} \circled{2}, \quad \mathbf{F}_{\mathbf{R}}(321) = \circled{3} \circled{2} \circled{1}, \quad \mathbf{F}_{\mathbf{R}}(122) = \circled{1} \circled{2,2}, \quad \mathbf{F}_{\mathbf{R}}(212) = \circled{2,1} \circled{1}, \\ \mathbf{F}_{\mathbf{R}}(221) &= \circled{2,2} \circled{1}, \quad \mathbf{F}_{\mathbf{R}}(112) = \circled{1,2} \circled{1}, \quad \mathbf{F}_{\mathbf{R}}(121) = \circled{1} \circled{2} \circled{1}, \quad \mathbf{F}_{\mathbf{R}}(211) = \circled{2} \circled{1,2}, \\ \mathbf{F}_{\mathbf{R}}(111) &= \circled{1,2,3}. \end{aligned}$$

More examples can be found in the annexes section with [tables A.1 à A.6](#).

We conclude by the main theorem of this subsection. It is a generalization of the construction of [\[Foi11\]](#) for **FQSym** and permutations to **WQSym** and packed words. Indeed, if we restrict the construction to permutations and we consider right children of a

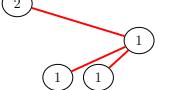
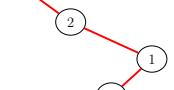
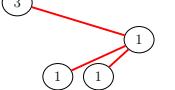
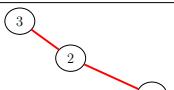
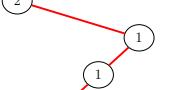
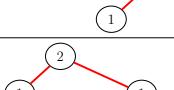
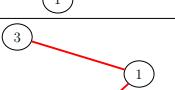
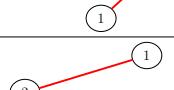
\cdot_T with $T = (\mathbf{V}, 1)$		\cdot_T with $T = (\cdot_T; 1)$ with $T' = (\mathbf{i}, 1)$	
\cdot_T with $T = (\mathbf{V}, 2)$		\cdot_T with $T = (\cdot_T; 2)$ with $T' = (\mathbf{i}, 1)$	
\cdot_T with $T = (\mathbf{i}, 1)$		\mathbf{i}^T with $T = (\mathbf{i}, 1)$	
\cdot_T with $T = (\mathbf{i}, 2)$		\mathbf{i}_T with $T = (\mathbf{i}, 1)$	

TABLE 4.2 – Equivalence between trees of [Foi11] and red-packed trees.

node as label of this node, we have the same construction as in [Foi11] with a shift of 1 for labels. Here are some examples of trees in [Foi11] and the equivalent red-packed tree.

All the constructions with red-packed forests have been done in order to have this theorem.

Théorème 4.40. *For all $n \in \mathbb{N}$ we have the three following equalities :*

$$\dim(\mathbf{WQSym}_n^*) = \#\mathfrak{F}_{\mathbf{R}n} \quad \text{and} \quad \dim(\mathbf{Prim}_n^*) = \#\mathfrak{T}_{\mathbf{R}n} \quad \text{and} \quad \dim(\mathbf{TPrim}_n^*) = \#\mathfrak{N}_{\mathbf{R}n}$$

Démonstration. théorème 4.38 proves the first equality. It also gives a relation between $\#\mathfrak{F}_{\mathbf{R}}$ and $\#\mathfrak{T}_{\mathbf{R}}$. Indeed a red-packed forest of weight n is an ordered sequence of red-packed trees of weight (n_k) such that $\sum_k (n_k) = n$. This relation is the same between $\dim(\mathbf{WQSym}_n^*)$ and $\dim(\mathbf{Prim}_n^*)$ according to proposition 3.25 (i.e. $\mathcal{A} = \mathcal{P}/(1 - \mathcal{P})$).

Red-skeleton trees are equivalent to ordered trees decorated by red-irreducible words as said in remarque 4.33. Recall that a basis of primitive elements is given by théorème 3.27 as ordered trees decorated by totally primitive elements. Elements of $\mathfrak{N}_{\mathbf{R}}$ are by definition in bijections with red-irreducible words, labels of red-skeleton trees. \square

4.2 Primal (Blue)

Now we do the same work for the primal side : **WQSym**. This subsection follows the same structure of statements as the previous one. Recall that in section 4.1 we constructed a bijection between packed words and red-forests by recursively decomposing packed words

using global descent and removal of maximums. In this section we follow the same path : inserting the last letter using ψ_{i^α} (lowercase i designates the integer value) instead of new maximums using ϕ_I (uppercase I designates the list of their positions). We define a blue-factorization of packed words. When used recursively, blue-factorization and global descent decomposition construct a bijection between PW and so-called blue-packed forests. Since the general structure of proofs are the same as in the previous section, we will mostly focus on the differences between combinatorial arguments.

4.2.1 Decompositions of packed words through last letter

In this section, we define two combinatorial operations on packed words (ψ_{i^α} and \blacktriangle) and the blue-factorization that use them. The unary operation ψ_{i^α} insert the new value i at the end of a given word. A word that cannot be factorized $u\blacktriangle v$ in a non trivial way is called blue-irreducible. Blue-irreducible words will index a new basis of **TPrim**.

Définition 4.41. Fix $n \in \mathbb{N}$ and $w \in \mathbf{PW}_n$. For any $1 \leq i \leq \max(w) + 1$ (with the convention $\max(\epsilon) = 0$), we denote by $\psi_{i^\circ}(w) = u_1 \cdots u_n \cdot i$ the packed word defined by $u_k = w_k$ if $w_k < i$ and $u_k = w_k + 1$ otherwise. We also define $\psi_{i^\bullet}(w) = w \cdot i$ for any $1 \leq i \leq \max(w)$.

Exemple 4.42. $\psi_{2^\circ}(1232) = 1343\textcolor{blue}{2}$, $\psi_{2^\bullet}(1232) = 1232\textcolor{blue}{2}$, $\psi_{4^\circ}(1232) = 1232\textcolor{blue}{4}$ and $\psi_{1^\circ}(\epsilon) = \textcolor{blue}{1}$.

Lemme 4.43. For any $W \in \mathbf{PW}_\ell$ where $\ell > 0$ there exists a unique triplet (i, α, w) where $i \in [1 \dots \ell + 1]$, $\alpha \in \{\circ, \bullet\}$ and w is a packed word, such that $W = \psi_{i^\alpha}(w)$.

Depending on α , the box diagram can be represented as

$$W = \begin{array}{c} \text{Diagram showing } W \text{ as a stack of boxes labeled } w, \text{ with a box labeled } i^\circ \text{ on top.} \\ \text{pack} \rightarrow w \end{array} \quad \text{or } W = \begin{array}{c} \text{Diagram showing } W \text{ as a stack of boxes labeled } w, \text{ with a box labeled } i^\bullet \text{ on top.} \\ \text{In the general case, we will note } W = \begin{array}{c} \text{Diagram showing } W \text{ as a stack of boxes labeled } w, \text{ with a dashed box labeled } i^\alpha \text{ on top.} \end{array} \end{array} .$$

Démonstration. Let $W \in \mathbf{PW}_\ell$ with $\ell > 0$ and i the value of the last letter of W .

- If i appears multiple times in W , then let $w = W_1 \dots W_{\ell-1}$, we only remove the last letter i of W . We have $W = \psi_{i^\bullet}(w)$.
- Otherwise, i appears only as the last letter, then let $w = \text{pack}(W_1 \dots W_{\ell-1})$, we remove the last letter i of W and pack the word. We have $W = \psi_{i^\circ}(w)$.

If $\psi_{i^\alpha}(u) = \psi_{j^\beta}(v)$ then the last letter is the same so $i = j$, the multiplicity of this letter is the same so $\alpha = \beta$ and the prefix are the same $u = v$. \square

Définition 4.44. Let $u, v \in \mathbf{PW}$ with $v \neq \epsilon$. By lemme 4.43, there is a unique triplet (i, α, v') such that $v = \psi_{i^\alpha}(v')$. Let $i' = i + \max(u)$, we define $u\blacktriangle v := \psi_{i'^\alpha}(v'/u)$. In other words, we remove the last letter of the right word, perform a reversed left shifted concatenation and adding back the last letter also shifted.

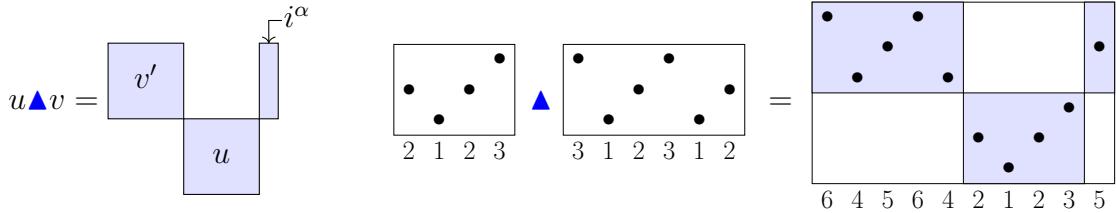


FIGURE 4.2 – Box diagrams : the operation \blacktriangle

Exemple 4.45. $2123 \blacktriangle 312312 = 2123 \blacktriangle \psi_{(2)} \bullet (31231) = \psi_{(2+3)} \bullet (645642123) = 6456421235$.

Lemme 4.46. Let w be an irreducible packed word. There exists a unique factorization of the form $w = u \blacktriangle v$ which maximizes the size of u . In this factorization, let v' and i^α such that $v = \psi_{i^\alpha}(v')$,

- either $v' = \epsilon$ and $i^\alpha = 1^\circ$,
- or v' is irreducible and $1 \leq i \leq \max(v')$.

We call it the **blue-factorization** of a word.

Exemple 4.47. Here is a first detailed example of a blue-factorization of an irreducible packed word :

Consider the irreducible packed word $w = 654623314$.

- The first step is to remove the last letter $i = 4$. Here there are multiple occurrences of the last letter in w , then $\alpha = \bullet$, we get $w' = 65462331$ which is a packed word, but is not irreducible.
 - As it is not the particular case where the last letter is the unique occurrence of the maximum letter, the second step is to set w'_1 as the first irreducible factor of w' and u the rest of w' . This way $w' = w'_1/u$ and the size of u is maximized. Here $w'_1 = 3213$ and $u = 2331$. Let $i' = i - \max(u) = 4 - 3 = 1$.
 - Finally, we get the following decomposition of w (see [définition 4.41](#) for ψ and [définition 4.44](#) for \blacktriangle) :
- $$w = 654623314 = u \blacktriangle \psi_{i'}(w'_1) = (2331) \blacktriangle \psi_1 \bullet (3213) = 2331 \blacktriangle 32131.$$

Exemple 4.48. Here are some other blue-factorizations :

$$\begin{aligned} 234313 &= 1 \blacktriangle \psi_2 \bullet (1232) = 1 \blacktriangle 12322 & 245413 &= 1 \blacktriangle \psi_{2^\circ}(1232) = 1 \blacktriangle 13432 \\ 11 &= \epsilon \blacktriangle \psi_1 \bullet (1) = \epsilon \blacktriangle 11 & 112 &= 11 \blacktriangle \psi_1 \bullet (\epsilon) = 11 \blacktriangle 1 \end{aligned}$$

Démonstration. Let w be irreducible and let (i, α, w') be the unique triplet such that $w = \psi_{i^\alpha}(w')$ according to [lemme 4.43](#).

If $i = \max(w)$ and it appears only one time (*i.e.* $\alpha = \circ$ and $i = \max(w') + 1$) then the blue-factorizations is $w = w' \blacktriangle \psi_1 \circ(\epsilon)$.

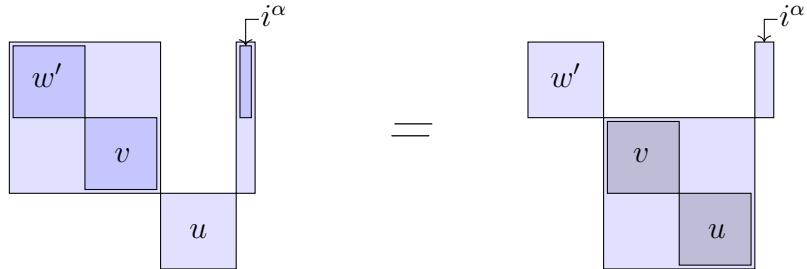
In any other case, we write $w' = w'_1/w'_2/\dots/w'_k$, the decomposition into irreducibles. Let $u = w'_2/\dots/w'_k$ and $i' = i - \max(u)$. We have that $i' \leq \max(w'_1)$ otherwise w wouldn't be packed and $1 \leq i'$ otherwise w wouldn't be irreducible. If $\alpha = \circ$ then $1 \neq i'$ otherwise w wouldn't be irreducible. Then we have $w = u \blacktriangle \psi_{i'\alpha}(w'_1)$ where the size of u is maximized. \square

Remarque 4.49. When restricted to permutations, blue-factorization is equal to a red-factorization applied to the inverse. Let σ be a permutation and $\sigma = \mu \blacktriangleright \nu$ be the red-factorization of σ , then $\sigma^{-1} = \mu^{-1} \blacktriangle \nu^{-1}$ is the blue-factorization of σ^{-1} .

Définition 4.50. A packed word w is **blue-irreducible** if w is irreducible and $w = u \blacktriangle v$ implies that $u = \epsilon$ (and $w = v$).

Here are some useful lemmas on the operation \blacktriangle . There are some similarities with [lemmes 4.11 et 4.12](#), [corollaire 4.14](#), et [proposition 4.15](#).

Lemme 4.51. For any $u, v, w \in \mathbf{PW}$ with $w \neq \epsilon$, we have $u \blacktriangle (v \blacktriangle w) = (v/u) \blacktriangle w$.

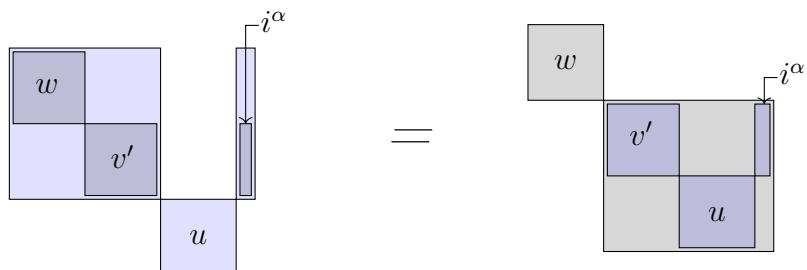


Démonstration. Let $u, v, w \in \mathbf{PW}$ with $w \neq \epsilon$, and let w' and i^α such that $w = \psi_{i^\alpha}(w')$.

$$\begin{aligned} u \blacktriangle (v \blacktriangle w) &= u \blacktriangle (\psi_{(i+\max(v))^\alpha}(w'/v)) \\ &= \psi_{(i+\max(v)+\max(u))^\alpha}((w'/v)/u) \\ &= \psi_{(i+\max(v)+\max(u))^\alpha}(w'/(v/u)) \\ &= (v/u) \blacktriangle \psi_{i^\alpha}(w') \\ &= (v/u) \blacktriangle w. \end{aligned}$$

\square

Lemme 4.52. For any $u, v, w \in \mathbf{PW}$ with $v \neq \epsilon$, we have $u \blacktriangle (w/v) = w/(u \blacktriangle v)$.



Démonstration. Let $u, v, w \in \mathbf{PW}$ with $v \neq \epsilon$ and let v' and i^α such that $v = \psi_{i^\alpha}(v')$.

$$\begin{aligned} u \blacktriangle (w/v) &= u \blacktriangle (w/\psi_{i^\alpha}(v')) \\ &= u \blacktriangle \psi_{i^\alpha}(w/v') \\ &= \psi_{(i+\max(u))^\alpha}((w/v')/u) \\ &= \psi_{(i+\max(u))^\alpha}(w/(v'/u)) \\ &= w/\psi_{(i+\max(u))^\alpha}(v'/u) \\ &= w/(u \blacktriangle v). \end{aligned}$$

□

Remarque 4.53. Theses relations are the same up to symmetry as the one with ► ([lemmes 4.11 et 4.12](#)). So adding the associativity of shifted concatenation $u/(v/w) = (u/v)/w$, the two operations \blacktriangle and $/$ verify relations of the *skew-duplicial operad* [[BDO20](#)].

Corollaire 4.54. *For any $u, v \in \mathbf{PW}$, we have that $u \blacktriangle v$ is irreducible if and only if v is irreducible.*

Démonstration. By contradiction, if $v = v_1/v_2$ then by [lemme 4.52](#) $u \blacktriangle v = v_1/(u \blacktriangle v_2)$. Now if $u \blacktriangle v = w_1/w_2$ as the value of the last letter of $u \blacktriangle v$ is greater than $\max(u)$ we have that $w_2 = w'_2 \cdot w''_2 \cdot i$ such that $\text{pack}(w''_2) = u$. We also have that $w_1/\text{pack}(w'_2 \cdot i) = v$. □

Proposition 4.55. *For any word w , $w = u \blacktriangle v$ is the blue-factorization of w if and only if v is blue-irreducible.*

Démonstration. Let $w \in \mathbf{PW}$ and let $u \blacktriangle v$ be the blue-factorization of w . Let v_1 and v_2 such that $v = v_1 \blacktriangle v_2$, then $(v_1/u) \blacktriangle v_2 = w$ by [lemme 4.51](#), but in the blue-factorization the size of u is maximized so $|(v_1/u)| \leq |u|$ and then we have that $v_1 = \epsilon$ so v is blue-irreducible.

Let $w \in \mathbf{PW}$ and let u and v such that $w = u \blacktriangle v$ and v is blue-irreducible. By contradiction, suppose that there exists u', v' such that $w = u' \blacktriangle v'$ with $|u| < |u'|$ and $v' \neq \epsilon$. Then necessarily u is a suffix of u' . Let u'' such that $u' = u'' \cdot u$, then $\text{pack}(u'') \blacktriangle v' = v$. But v is blue-irreducible. So the size of u is maximal if v is blue-irreducible. □

Thanks to [remarque 4.49](#) the following proposition is immediate.

Proposition 4.56. *A permutation σ is blue-irreducible if and only if σ^{-1} is red-irreducible.*

4.2.2 Blue-forests from decomposed packed words using ψ

As in [section 4.1.2](#) we will apply recursively the blue-factorization of the former section to construct a bijection between packed words and a certain kind of labeled biplane trees.

In this construction, the labels can be a blue-irreducible word for skeleton, or an integer with a sign $\alpha \in \{\circ, \bullet\}$. In order to differentiate the trees from the one of the previous section, we will draw them in [blue](#). As before, for a labeled biplane tree, we denote the trees by $\text{Node}_B(x, f_\ell, f_r)$.

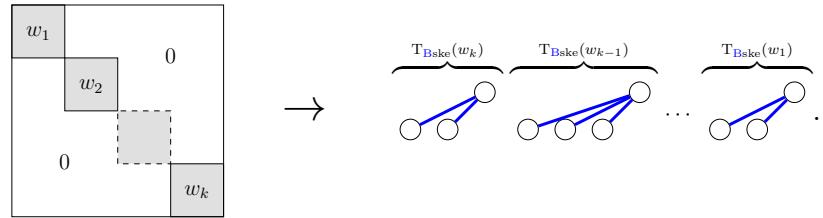
Exemple 4.57.

$\text{Node}_B(1^\circ, [], []) = \textcircled{1^\circ}$, and $\text{Node}_B(1^\bullet, [\text{Node}_B(1^\circ, [], [])], [\text{Node}_B(1^\circ, [], [])]) = \begin{array}{c} \textcircled{1^\bullet} \\ \textcircled{1^\circ} \end{array} \text{---} \begin{array}{c} \textcircled{1^\circ} \end{array}$.

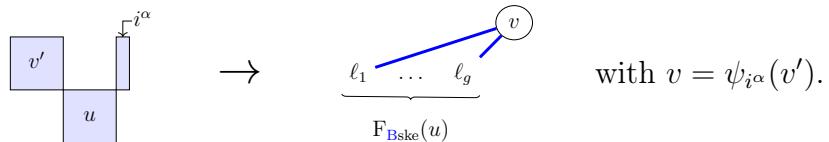
We apply recursively the global descent decomposition and the blue-factorization of lemme 4.46. We obtain an algorithm which takes a packed word and returns a biplane forest where nodes are decorated by blue-irreducible words :

Définition 4.58. Exactly as définition 4.23 of F_{Rske} and T_{Rske} , we now define two functions F_{Bske} and T_{Bske} . These functions transform respectively a packed word and an irreducible packed word into respectively a biplane forest and a biplane tree. These functions are defined in a mutual recursive way as follow :

- $F_{Bske}(\epsilon) = []$ (empty forest),
- for any packed word w , let $w_1/w_2/\dots/w_k$ be the global descent decomposition of w , then $F_{Bske}(w) := [T_{Bske}(w_k), T_{Bske}(w_{k-1}), \dots, T_{Bske}(w_1)]$ (notice the inversion compared to définition 4.23).

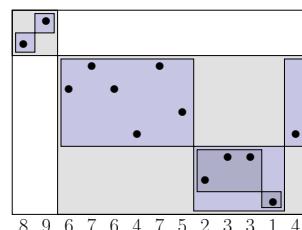


- for any irreducible packed word w , we define $T_{Bske}(w) := \text{Node}_B(v, F_{Bske}(u), [])$ where $w = u \blacktriangle v$ is the blue-factorization of w .



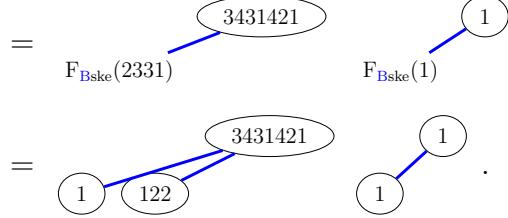
Exemple 4.59. Let $w = 8967647523314$, here is the global descent decomposition $w = w_1/w_2$ with $w_1 = 12$ and $w_2 = 67647523314$. Now, we have the blue-factorization of w_1 and w_2 using lemme 4.46 as

$$w_1 = 1 \blacktriangle \psi_{1^\circ}(\epsilon) \quad \text{and} \quad w_2 = 2331 \blacktriangle \psi_{1^\bullet}(343142) = (122/1) \blacktriangle 3431421,$$



It gives the following forest :

$$F_{Bske}(8967647523314) = [T_{Bske}(12), T_{Bske}(67647523314)]$$

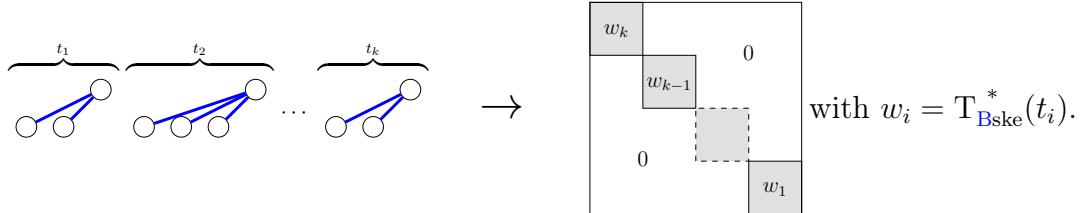


Définition 4.60. A labeled biplane forest (resp. tree) is a **blue-skeleton forest** (resp. **tree**) if and only if it is labeled by blue-irreducible words and no node has a right child.

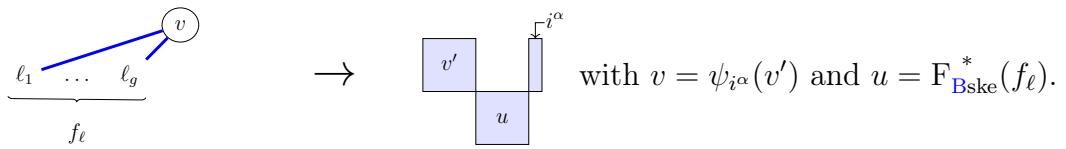
We want to prove that the functions F_{Bske} and T_{Bske} are bijections. To do that, as for F_{Rske} and T_{Rske} , we first define the two inverse functions.

Définition 4.61. We now define two functions F_{Bske}^* and T_{Bske}^* that transform respectively a blue-skeleton forest and tree into packed words. These functions are defined in a mutual recursive way as follow :

- $F_{Bske}^*([]) = \epsilon$,
- for any blue-skeleton forest $f = [t_1, \dots, t_k]$, we define
 $F_{Bske}^*(f) := T_{Bske}^*(t_k) / \dots / T_{Bske}^*(t_1)$.
 (notice the inversion compared to [définition 4.26](#))



- for any blue-skeleton tree $t = \text{Node}_B(v, f_r, []), we define
 $T_{Bske}^*(t) := F_{Bske}^*(f_r) \blacktriangle v$.$



Lemme 4.62. The functions F_{Bske} and F_{Bske}^* (resp. T_{Bske} and T_{Bske}^*) are two inverse bijections between packed words and blue-skeleton forests (resp. irreducible packed words and blue-skeleton trees). That is to say $F_{Bske}^{-1} = F_{Bske}^*$ and $T_{Bske}^{-1} = T_{Bske}^*$.

Démonstration. The proof structure is the same as the one of [lemme 4.27](#) with use of statements coming from this subsection. We can see in this table some of the main statements that are exchanged for this dual part :

lemme 4.7	lemme 4.46	red-factorization and blue-factorization.
définition 4.10	définition 4.50	red-irreducible words and blue-irreducible words.
corollaire 4.14	corollaire 4.54	$u \blacktriangleright v$ irreducible $\iff v$ irreducible $u \blacktriangle v$ irreducible $\iff v$ irreducible.
proposition 4.15	proposition 4.55	$u \blacktriangleright v$ red-factorization $\iff v$ red-irreducible $u \blacktriangle v$ blue-factorization $\iff v$ blue-irreducible.
définition 4.23	définition 4.58	F_{Rske} , T_{Rske} and F_{Bske} , T_{Bske} .
définition 4.25	définition 4.60	red-skeleton forest and blue-skeleton forest.
définition 4.26	définition 4.61	F_{Rske}^* , T_{Rske}^* and F_{Bske}^* , T_{Bske}^* .

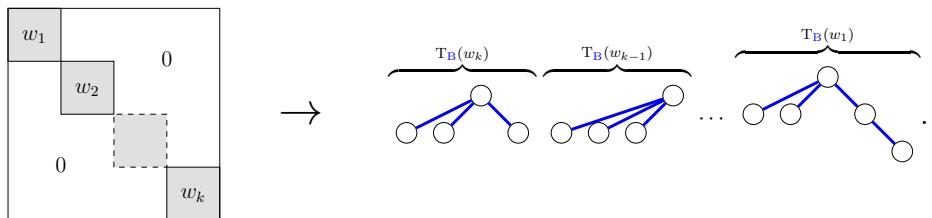
□

Now that we have the blue-skeleton, we will add right forests to every nodes to have biplane trees. For every nodes, if v is the blue-irreducible word in the label, then with lemme 4.43 $v = \psi_{i^\alpha}(v')$, i^α is the new label and $F_B(v')$ is the new right forest.

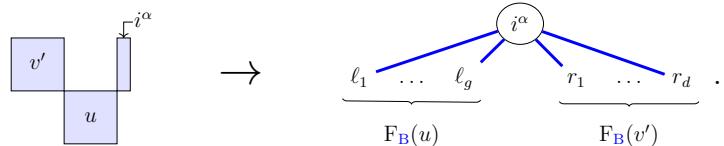
Here is the formal definition of $F_B(w)$ and $T_B(w)$ which is very similar to définition 4.58, only the third item is different. The labels are now pairs of an integer and a sign $\alpha \in \{\circ, \bullet\}$.

Définition 4.63. The forest $F_B(w)$ (resp. tree $T_B(w)$) associated to a packed word (resp. irreducible packed word) w are defined in a mutual recursive way as follows :

- $F_B(\epsilon) = []$ (empty forest),
- for any packed word w , let $w_1/w_2/\dots/w_k$ be the global descent decomposition of w , then $F_B(w) := [T_B(w_k), T_B(w_{k-1}), \dots, T_B(w_1)]$ (notice the inversion compared to définition 4.28).

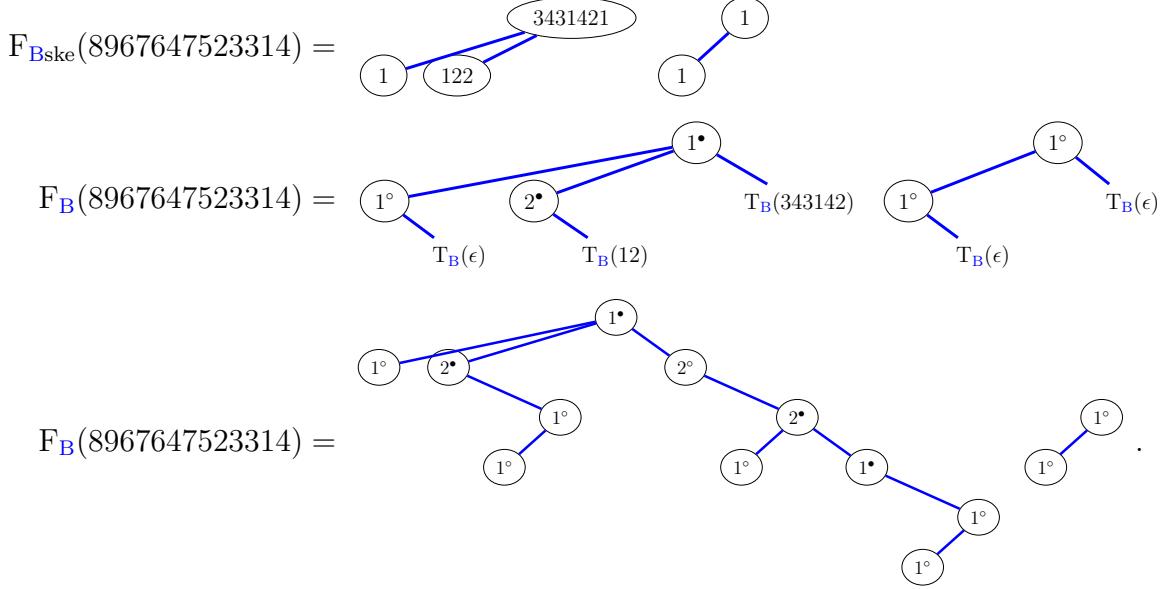


- for any irreducible packed word w , we define $T_B(w) := \text{Node}_B(i^\alpha, F_B(u), F_B(v'))$ where $w = u \blacktriangle \psi_{i^\alpha}(v')$ is the blue-factorization of w . In practice, as v' is irreducible, $F_B(v')$ will always be composed of zero or one tree.

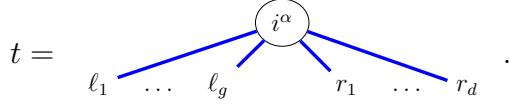


Exemple 4.64. Consider again $w = 8967647523314$. We start from the blue-skeleton forest

from exemple 4.59.



Définition 4.65. Let t be a labeled biplane tree. We write $t = \text{Node}_B(i^\alpha, f_\ell, f_r)$ where $i \in \mathbb{N}_{>0}$, $\alpha \in \{\circ, \bullet\}$, $f_\ell = [\ell_1, \dots, \ell_g]$, and $f_r = [r_1, \dots, r_d]$, which is depicted as follows :



The **weight** of t ($\omega(t)$) is the number of nodes with \circ in t . By extension, the **weight** of a forest is the sum of the weight of its trees.

Lemme 4.66. The weight of a forest (resp. a tree) obtain by the functions F_B (resp. T_B) is equal to the maximum value of the word. i.e. $\forall w \in \mathbf{PW}, \omega(F_B(w)) = \max(w)$, $\forall w \in \mathbf{PW}$ with w irreducible, $\omega(T_B(w)) = \max(w)$.

Démonstration. We prove by induction with the following hypothesis, for $n \in \mathbb{N}$:

$$\begin{aligned} \forall w \in \mathbf{PW}_n, \quad & \omega(F_B(w)) = \max(w), \\ \forall w \in \mathbf{PW}_n \text{ with } w \text{ irreducible}, \quad & \omega(T_B(w)) = \max(w). \end{aligned} \tag{4.10}$$

The base case is given by the first item of [définition 4.63](#) as $F_B(\epsilon) = []$ and $\omega([]) = \max(\epsilon) = 0$ by convention.

Let us fix $n \geq 1$ and suppose that the hypothesis (4.10) holds. Let $w \in \mathbf{PW}_{n+1}$ and $w = w_1/w_2/\dots/w_k$ be the global descent decomposition of w .

• If $k = 1$, we have $F_B(w) = [T_B(w)]$. Let $w = u \blacktriangle \psi_{i^\alpha}(v)$ with $i \in \mathbb{N}_{>0}$ and $\alpha \in \{\circ, \bullet\}$ be the blue-factorization of w , then, depending of α the node is counted or not :

$\omega(T_B(w)) = \omega(\text{Node}_B(i^\alpha, F_B(u), F_B(v)))$ $\stackrel{\text{if } \alpha=\circ}{=} (1+) \overbrace{\omega(F_B(u)) + \omega(F_B(v))}^{\text{The sizes of } u \text{ and } v \text{ are at most } n}$. The sizes of u and v are at most n so by induction $\omega(T_B(w)) = (1+) \max(u) + \max(v) = \max(w)$.

- If $k \geq 2$, by induction on each factors, we have that

$$\omega(F_B(w)) = \omega(T_B(w_1)) + \cdots + \omega(T_B(w_k)) = \max(w_1) + \cdots + \max(w_k) = \max(w). \quad \square$$

Définition 4.67. Using the same notations as in previous [définition 4.65](#), we say that t is a **blue-packed tree** if it satisfies :

$$\left\{ \begin{array}{l} d = 0, \\ i^\alpha = 1^\circ, \\ \ell_1, \dots, \ell_g \text{ are blue-packed trees.} \end{array} \right. \text{ or } \left\{ \begin{array}{l} d = 1, \\ i^\alpha \neq 1^\circ, \\ 1 \leq i \leq \omega(r_1), \\ \ell_1, \dots, \ell_g \text{ and } r_1 \text{ are blue-packed trees.} \end{array} \right. \quad (4.11)$$

An ordered list of blue-packed trees is a **blue-packed forest**.

Remarque 4.68. The same remark as [remarque 4.33](#) can be done with blue-skeleton trees that can be interpreted as flattened representations of blue-packed trees. Symmetrically, blue-packed trees can be interpreted as unfolded representations of blue-skeleton trees. We use the operation ψ_{i^α} to change between blue-packed and blue-skeleton trees.

Notations 4.69. In the same way as notations [4.34](#) we add the following notations :

- \mathfrak{F}_{Bn} the set of blue-packed forests of size n , ($\mathfrak{F}_{Bn} := \{F_B(w)\}_{w \in PW_n}$),
- \mathfrak{T}_{Bn} the set of blue-packed trees of size n , ($\mathfrak{T}_{Bn} := \{T_B(w)\}_{w \in PW_n}$ with w irreducible),
- \mathfrak{N}_{Bn} the set of blue-packed trees of size n such that the left forest of the root is empty, ($\mathfrak{N}_{Bn} := \{T_B(w)\}_{w \in PW_n}$ with w blue-irreducible). In particular, the blue-skeleton of a tree of \mathfrak{N}_{Bn} consist of a single node labeled by a blue-irreducible word.

Remarque 4.70. The set \mathfrak{N}_{Bn} can be described as a disjointed union of sets depending on i and α . Let $\mathfrak{F}_{Bn}^{i^\alpha}$ denote the set of blue-packed forests of weight n that can be right children of a node labeled by i^α (see [définition 4.67](#) for conditions), we have the following description :

$$\mathfrak{N}_{Bn} = \bigsqcup_{i,\alpha} \{\text{Node}_B(i^\alpha, [], f_r) \mid f_r \in \mathfrak{F}_{Bn-p}^{i^\alpha}\} \quad (4.12)$$

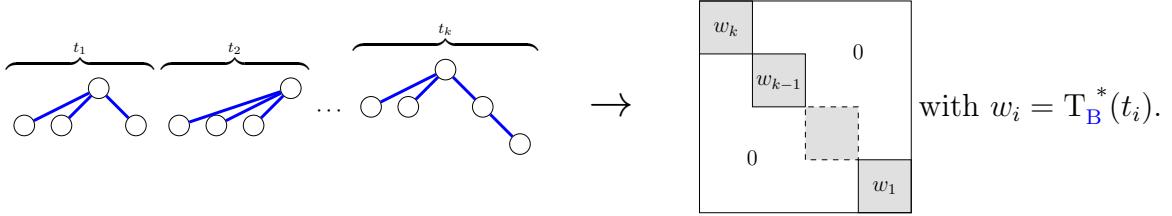
Analogously, we use \mathfrak{F}_{Bske} , \mathfrak{T}_{Bske} and \mathfrak{N}_{Bske} for blue-skeleton forests, trees and trees with only one node.

We can remark that for $n = 1$ we have $\mathfrak{N}_{B1} = \mathfrak{T}_{B1} = \mathfrak{F}_{B1}$ and $\forall n > 1, \mathfrak{N}_{Bn} \subsetneq \mathfrak{T}_{Bn} \subsetneq \mathfrak{F}_{Bn}$.

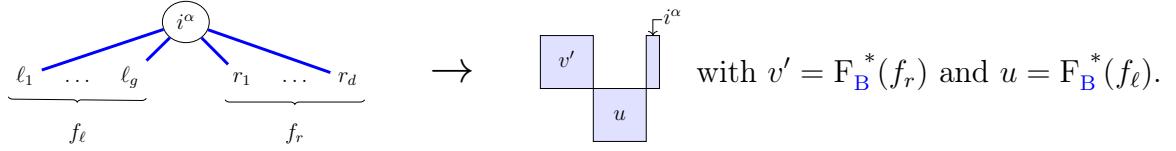
Once again we define two functions in order to prove that F_B and T_B are bijections.

Définition 4.71. We define here the functions F_B^* (resp. T_B^*) that transform blue-packed forest f (resp. tree t) into a packed word. We reverse all instructions of [définition 4.63](#) as follows :

- $F_B^*([]) = \epsilon$,
- for any non empty blue-packed forest $f = [t_1, t_2 \dots, t_k]$, then
 $F_B^*(f) = T_B^*(t_k)/T_B^*(t_{k-1})/\dots/T_B^*(t_1)$
(notice the inversion compared to [définition 4.36](#)).



- for any non empty blue-packed tree $t = \text{Node}_B(i^\alpha, f_\ell, f_r)$, then
 $T_B^*(t) = F_B^*(f_\ell) \blacktriangle \psi_{i^\alpha}(F_B^*(f_r))$.



As $\psi_{i^\alpha}(F_B^*(f_r))$ is only defined if $i \leq \max(F_B^*(f_r)) + 1$ (see [définition 4.41](#)), there might be a problem with this definition. We prove in the following [lemme 4.72](#) that this is the case if $t \in \mathfrak{T}_B$.

Lemme 4.72. *For any blue-packed forest f , $F_B^*(f)$ is a well defined word and its maximum value is $\omega(f)$. For any blue-packed tree t , $T_B^*(t)$ is a well defined word and its maximum is $\omega(t)$.*

Démonstration. We prove by induction with the following hypothesis, for $n \in \mathbb{N}$:

$$\begin{aligned} \forall f \in \mathfrak{F}_{B \leq n}, \quad & F_B^*(f) \text{ is well defined and } \max(F_B^*(f)) = \omega(f), \\ \forall t \in \mathfrak{T}_{B \leq n}, \quad & T_B^*(t) \text{ is well defined and } \max(T_B^*(t)) = \omega(t). \end{aligned} \tag{4.13}$$

The base case is given by the first item of [définition 4.71](#) as $F_B^*([]) = \epsilon$ and $\omega([]) = \max(\epsilon) = 0$.

Let us fix $n \geq 1$ and suppose that the hypothesis (4.13) holds. Let $f = [t_1, \dots, t_k] \in \mathfrak{F}_{Bn+1}$.

• If $k = 1$, it is sufficient to prove the second item of (4.13). Let $t = \text{Node}_B(i^\alpha, f_\ell, f_r) \in \mathfrak{T}_{Bn+1}$. According to [définition 4.67](#) with notations of [définition 4.65](#) there are two cases :

• $d = 0$ and $i^\alpha = 1^\circ$. We have that $\max(F_B^*(f_r)) = \max(\epsilon) = 0$ so $i \leq 0 + 1$ and $\psi_{i^\alpha}(\epsilon) = 1$. Now by induction on f_ℓ , we have that $F_B^*(f_\ell)$ is a well defined word and its maximum value is $\omega(f_\ell)$. Finally $T_B^*(t) = F_B^*(f_\ell) \blacktriangle \psi_{i^\alpha}(\epsilon) = F_B^*(f_\ell) \setminus \psi_{i^\alpha}(\epsilon)$ is a well defined word and its maximum is the last value : $\max(F_B^*(f_\ell)) + 1 = \omega(t)$.

• $d = 1$. In this case, we can directly apply the hypothesis (4.13) on f_r and f_ℓ . According to [définition 4.67](#) we have that

$$\begin{aligned} i &\leq \omega(r_1), \\ i &\leq \max(F_B^*(f_r)). \end{aligned}$$

So $T_B^*(t)$ is well defined. Moreover

$$\begin{aligned} \max(T_B^*(t)) &= \max(F_B^*(f_\ell) \blacktriangleleft \psi_{i^\alpha} F_B^*(f_r)) \\ &= \max(F_B^*(f_\ell)) + \max(F_B^*(f_r))(+1) \\ &= \omega(f_\ell) + \omega(f_r)(+1) = \omega(t). \end{aligned}$$

- If $k \geq 2$, the weight of trees are at least 1 so we can apply (4.13) on trees of f .

□

Théorème 4.73. *The functions F_B and F_B^* (resp. T_B and T_B^*) are two inverse bijections between packed words of size n and blue-packed forests (resp. irreducible packed words and blue-packed trees) of size n . That is to say $F_B^{-1} = F_B^*$ and $T_B^{-1} = T_B^*$.*

Démonstration. The proof structure is the same as the one of [théorème 4.38](#) which is similar to the one of [lemmes 4.27 et 4.62](#). But with use of statements comming from this subsection. We can see in this table some of the main statements that are exchanged with their counterpart :

lemme 4.7	lemme 4.46	red-factorization and blue-factorization.
définition 4.10	définition 4.50	red-irreducible words and blue-irreducible words.
corollaire 4.14	corollaire 4.54	$u \blacktriangleright v$ irreducible $\iff v$ irreducible $u \blacktriangleleft v$ irreducible $\iff v$ irreducible.
proposition 4.15	proposition 4.55	$u \blacktriangleright v$ red-factorization $\iff v$ red-irreducible $u \blacktriangleleft v$ blue-factorization $\iff v$ blue-irreducible.
définition 4.28	définition 4.63	F_R , T_R and F_B , T_B .
définition 4.30	définition 4.65	weight of red-forests (\sum size of labels) weight of blue-forests (\sum nodes with \circ).
lemme 4.31	lemme 4.66	$\omega(F_R(w)) = w $ and $\omega(F_B(w)) = \max(w)$.
définition 4.32	définition 4.67	red-packed forest and blue-packed forest.
définition 4.36	définition 4.71	F_R^* , T_R^* and F_B^* , T_B^* .

□

Remarque 4.74. As we can see in [sections 4.1 et 4.2](#), the role of size and weight are exchanged for red and blue-forests. For red forests, the size (number of nodes) is equal to the maximum letter of the word associated while the weight ([définition 4.30](#)) is the number of letter of the associated word. For blue-forests, it is the opposite, the number of letters of the associated word is equal to the size of the forest while le maximum letter is equal to the weight ([définition 4.65](#)) of the forest. That is why we denote the set of red packed forests of **weight** n by \mathfrak{F}_R^n and the set of blue-packed forests of **size** n by \mathfrak{F}_B^n .

Exemple 4.75. There is a unique forest in \mathfrak{F}_{B1} , namely \circlearrowleft , here are the 3 forests of \mathfrak{F}_{B2} with the associated packed word : $F_B(12) = \circlearrowleft \circlearrowright$, $F_B(21) = \circlearrowleft \circlearrowleft$,

$F_B(11) = \circlearrowleft \circlearrowleft$. We show below the forests of \mathfrak{F}_{B3} :

$$\begin{aligned} F_B(123) &= \circlearrowleft \circlearrowleft \circlearrowleft, \quad F_B(132) = \circlearrowleft \circlearrowleft \circlearrowright, \quad F_B(213) = \circlearrowleft \circlearrowleft \circlearrowright, \quad F_B(231) = \circlearrowleft \circlearrowleft \circlearrowleft, \\ F_B(312) &= \circlearrowleft \circlearrowleft \circlearrowleft, \quad F_B(321) = \circlearrowleft \circlearrowleft \circlearrowleft, \quad F_B(122) = \circlearrowleft \circlearrowleft \circlearrowleft, \quad F_B(212) = \circlearrowleft \circlearrowleft \circlearrowleft, \\ F_B(221) &= \circlearrowleft \circlearrowleft \circlearrowleft, \quad F_B(112) = \circlearrowleft \circlearrowleft \circlearrowleft, \quad F_B(121) = \circlearrowleft \circlearrowleft \circlearrowleft, \\ F_B(111) &= \circlearrowleft \circlearrowleft \circlearrowleft. \end{aligned}$$

More examples can be found in the annexes section with tables A.1 à A.6.

We conclude by the main theorem of this subsection. It is the dual of théorème 4.40.

Théorème 4.76. For all $n \in \mathbb{N}$ we have the three following equalities :

$$\dim(\mathbf{WQSym}_n) = \#\mathfrak{F}_{Bn} \quad \text{and} \quad \dim(\mathbf{Prim}_n) = \#\mathfrak{T}_{Bn} \quad \text{and} \quad \dim(\mathbf{TPrim}_n) = \#\mathfrak{N}_{Bn}$$

Démonstration. The proof is similar to the one of théorème 4.40 thanks to théorème 4.73 instead of théorème 4.38. \square

4.3 Intervals for half shuffle products

We define two operations \triangleleft and \triangleright that are some kind of reversal of \blacktriangleright and \blacktriangleleft (see définitions 4.5 et 4.44). These operations are only defined for proposition 4.81.

Définition 4.77. Let $u, v \in \mathbf{PW}$ with $u \neq \epsilon$. By lemme 4.4, there is a unique pair (I, u') such that $u = \phi_I(u')$. We define $u \triangleleft v := \phi_I(u' \setminus v)$. In other words, we remove the maximum letter of the left word, perform a right shifted concatenation and reinsert the removed letters as new maximums.

Exemple 4.78. $23123 \triangleleft 322312 = \phi_{2,5}(212) \triangleleft 322312 = \phi_{2,5}(212544534) = 26126544534$.

Définition 4.79. Let $u, v \in \mathbf{PW}$ with $u \neq \epsilon$. By lemme 4.43, there is a unique triplet (i, α, u') such that $u = \psi_{i\alpha}(u')$. We define $u \triangleright v := \psi_{i\alpha}(u' \setminus v)$. In other words, we remove the last letter of the left word, perform a right shifted concatenation and adding back the last letter.

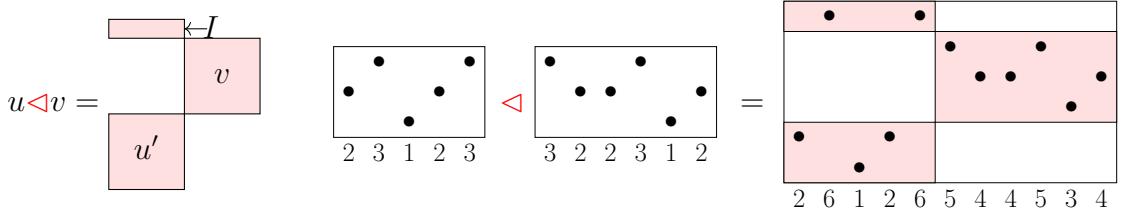


FIGURE 4.3 – Box diagrams : the operation \triangleleft

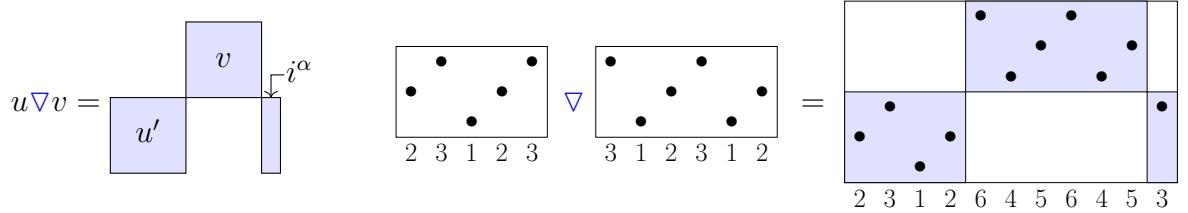


FIGURE 4.4 – Box diagrams : the operation \triangledown

Exemple 4.80.

$$23123 \triangledown 312312 = \psi_{(3)} \bullet (2312) \triangleright 312312 = \psi_{(3)} \bullet (2312645645) = 23126456453.$$

This proposition is a generalization of [LR02, Proposition 4.6] and it is proved in [PR04, Theorem 33] using different notations.

Proposition 4.81. *For all $u, v \in \mathbf{PW}$:*

$$\mathbb{R}_u \overline{\prec} \mathbb{R}_v = \sum_{w \in [u \triangledown v, v \setminus u]_R} \mathbb{R}_w, \quad (4.14)$$

$$\mathbb{R}_u \overline{\subset} \mathbb{R}_v = \sum_{w \in [u/v, u \blacktriangle v]_R} \mathbb{R}_w, \quad (4.15)$$

$$\mathbb{Q}_u \overline{\exists} \mathbb{Q}_v = \sum_{w \in [u \triangleleft v, u/v]_L} \mathbb{Q}_w, \quad (4.16)$$

$$\mathbb{Q}_u \overline{\sum} \mathbb{Q}_v = \sum_{w \in [u \setminus v, u \blacktriangleright v]_L} \mathbb{Q}_w, \quad (4.17)$$

where $[a, b]_R$ (resp. $[a, b]_L$) is an interval of the right weak order (*définition 2.19*) (resp. left weak order (*définition 2.24*)).

Exemple 4.82. Let $u = 12$ and $v = 121$, we have that

$$[u \triangleleft v, u/v]_L = [14232, 34121]_L \quad \text{and} \quad [u \triangleright v, u \triangleright v]_L = [12343, 23141]_L.$$

$$\begin{aligned}\mathbb{Q}_{12} \overline{\preceq} \mathbb{Q}_{121} &= \sum_{w \in [14232, 34121]_L} \mathbb{Q}_w, \\ &= \mathbb{Q}_{14232} + \mathbb{Q}_{24131} + \mathbb{Q}_{34121}, \\ \mathbb{Q}_{12} \overline{\succeq} \mathbb{Q}_{121} &= \sum_{w \in [12343, 23141]_L} \mathbb{Q}_w, \\ &= \mathbb{Q}_{12343} + \mathbb{Q}_{13242} + \mathbb{Q}_{23141}.\end{aligned}$$

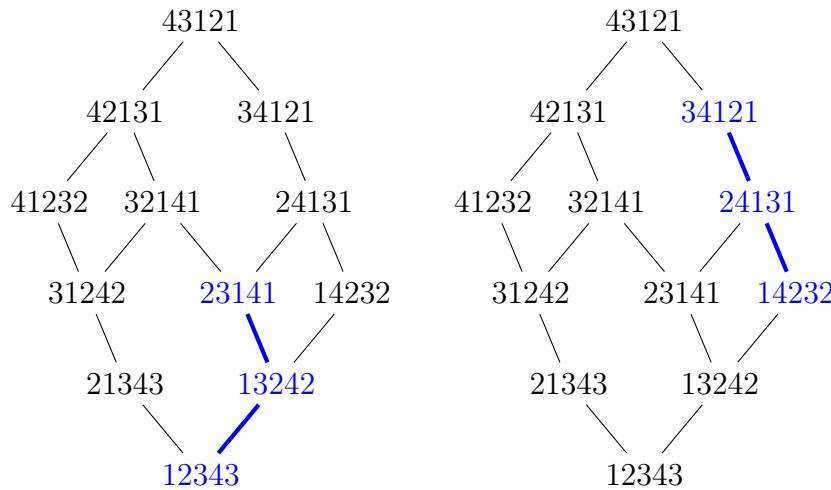


FIGURE 4.5 – Component of the left weak order for the two half products of \mathbb{Q}_{12} and \mathbb{Q}_{121} . $\overline{\preceq}$ on the left and $\overline{\succeq}$ on the right.

Exemple 4.83. Let $u = 11$ and $v = 121$, we have that

$$[u \triangleright v, v \setminus u]_R = [12321, 23211]_R \quad \text{and} \quad [u/v, u \blacktriangle v]_R = [11232, 23112]_R.$$

$$\mathbb{R}_{11} \setminus \mathbb{R}_{121} = \sum_{w \in [12321, 23211]_R} \mathbb{R}_w,$$

$$= \mathbb{R}_{12321} + \mathbb{R}_{21321} + \mathbb{R}_{23121} + \mathbb{R}_{23211},$$

$$\mathbb{R}_{11} \setminus \mathbb{R}_{121} = \sum_{w \in [11232, 23112]_R} \mathbb{R}_w,$$

$$= \mathbb{R}_{11232} + \mathbb{R}_{12132} + \mathbb{R}_{21132} + \mathbb{R}_{12312} + \mathbb{R}_{21312} + \mathbb{R}_{23112}.$$

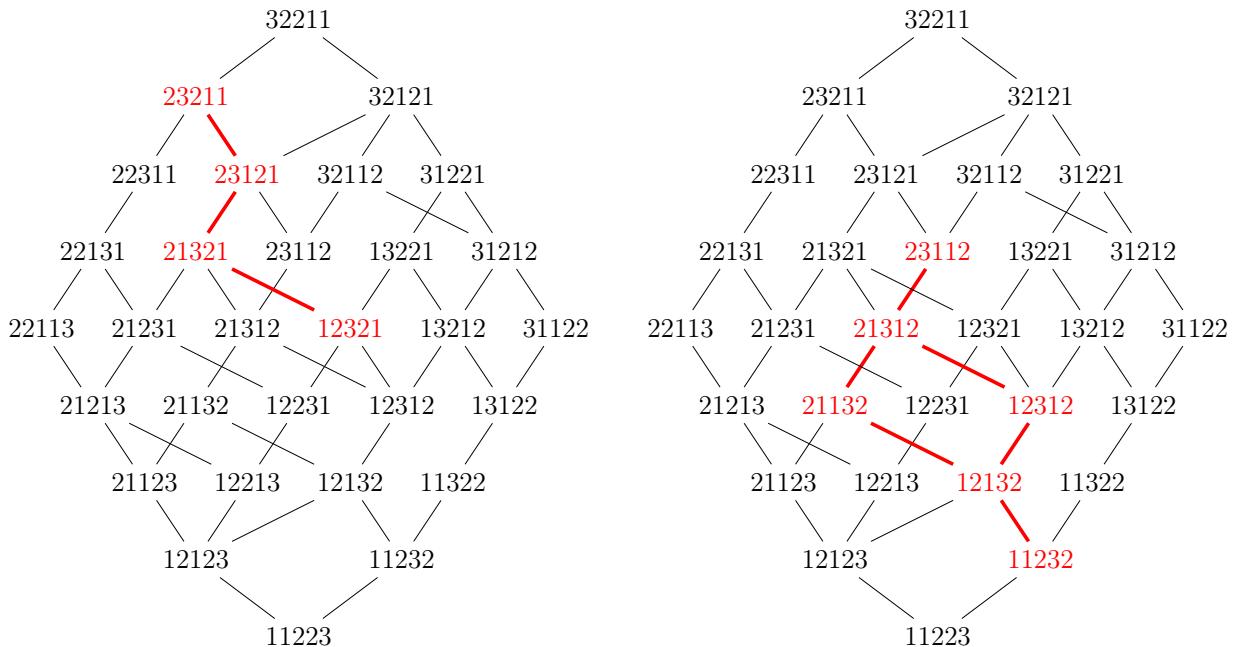


FIGURE 4.6 – Component of the right weak order for the two half products of \mathbb{R}_{11} and \mathbb{R}_{121} . \setminus on the left and \setminus on the right.

Chapitre 5

Bases for totally primitive elements

In this chapter we construct two bases of primitive and totally primitive elements of **WQSym** and **WQSym**^{*}. Thanks to théorèmes 4.40 et 4.76 we now have the combinatorial objects to index those bases and we know that their numbers agree with the dimensions. We therefore only need to show that they are linearly independent. We will proceed by showing that the decompositions through maximum and through last letter preserve the total primitivity.

As in chapitre 4, we start by working on **WQSym**^{*} associated to the color red and do the same work on the primal **WQSym** associated to the color blue.

5.1 Dual (Red)

5.1.1 Decomposition through maximums and totally primitive elements

Définition 5.1. Let $I = [i_1, \dots, i_p]$ with $0 < i_1 < \dots < i_p$. We define a linear map $\Phi_I : \mathbf{WQSym}^* \rightarrow \mathbf{WQSym}^*$ as follows : for all $n \in \mathbb{N}$ and $w = w_1 \cdot w_2 \cdots w_n \in \mathbf{PW}_n$,

$$\Phi_I(\mathbb{R}_w) := \begin{cases} \mathbb{R}_{\phi_I(w)} & \text{if } i_p \leq n + p, \\ 0 & \text{if } i_p > n + p. \end{cases} \quad (5.1)$$

Définition 5.2. Let $I = (i_1, \dots, i_p)$ with $0 < i_1 < \dots < i_p$. We define a projector $\tau_I : \mathbf{WQSym}^* \rightarrow \mathbf{WQSym}^*$ as follows : for all $n \in \mathbb{N}$ and $w = w_1 \cdot w_2 \cdots w_n \in \mathbf{PW}_n$,

$$\tau_I(\mathbb{R}_w) := \begin{cases} \mathbb{R}_w & \text{if } w_i = \max(w) \text{ if and only if } i \in I, \\ 0 & \text{otherwise.} \end{cases} \quad (5.2)$$

These are orthogonal projectors in the sense that $\tau_I^2 = \tau_I$ and $\tau_I \circ \tau_J = 0$ ($I \neq J$).

Lemme 5.3. For any I , we have $\text{Im}(\Phi_I) = \text{Im}(\tau_I)$ where $\text{Im}(f)$ denotes the image of f .

Démonstration. For any I , the inclusion $\text{Im}(\Phi_I) \subset \text{Im}(\tau_I)$ is automatic by definition of Φ_I and τ_I . Indeed, for any $w \in \mathbf{PW}_n$ if $i_p \leq n + p$ then $\Phi_I(\mathbb{R}_w) = \mathbb{R}_{\phi_I(w)}$ and $\tau_I(\mathbb{R}_{\phi_I(w)}) = \mathbb{R}_{\phi_I(w)}$ and $\Phi_I(\mathbb{R}_w) = 0$ otherwise. By linearity $\text{Im}(\Phi_I) \subset \text{Im}(\tau_I)$.

For any I , the inclusion $\text{Im}(\Phi_I) \supset \text{Im}(\tau_I)$ is a consequence of [lemme 4.4](#) and linearity. Indeed, for any $w \in \mathbf{PW}$, $\tau_I(\mathbb{R}_w) = \mathbb{R}_w \Leftrightarrow (w_i = \max(w) \Leftrightarrow i \in I)$. If $\tau_I(\mathbb{R}_w) = \mathbb{R}_w$ let w' be such that $\phi_I(w') = w$ using [lemme 4.4](#), then $\Phi_I(\mathbb{R}_{w'}) = \mathbb{R}_w = \tau_I(\mathbb{R}_w)$. By linearity $\text{Im}(\Phi_I) \supset \text{Im}(\tau_I)$. \square

Lemme 5.4. *For any I , the projection by τ_I of a totally primitive element is still a totally primitive element, so that $\tau_I(\mathbf{TPrim}^*) = \text{Im}(\tau_I) \cap \mathbf{TPrim}^*$. Moreover,*

$$\mathbf{TPrim}^* = \bigoplus_I \text{Im}(\tau_I) \cap \mathbf{TPrim}^*. \quad (5.3)$$

Démonstration. Let w a packed word. We have $\Delta_{\prec}(\tau_I(\mathbb{R}_w)) = (\tau_I \otimes \text{Id}) \circ \Delta_{\prec}(\mathbb{R}_w)$ by definition of τ_I and Δ_{\prec} . Indeed, in $\Delta_{\prec}(\mathbb{R}_w)$, the deconcatenations cannot be done before the last maximum letter of w . By linearity, for all $p \in \mathbf{TPrim}^*$, we have $\Delta_{\prec}(\tau_I(p)) = (\tau_I \otimes \text{Id}) \circ \Delta_{\prec}(p) = 0$. The same argument works on the right so that $\tau_I(p) \in \mathbf{TPrim}^*$. Moreover τ_I are orthogonal projectors so $\mathbf{TPrim}^* = \bigoplus_I \tau_I(\mathbf{TPrim}^*) = \bigoplus_I \text{Im}(\tau_I) \cap \mathbf{TPrim}^*$. \square

5.1.2 The new basis \mathbb{P}

Définition 5.5. Let t_1, \dots, t_k be k red-packed trees, $I = [i_1, \dots, i_p]$, $f_l = [\ell_1, \dots, \ell_g]$ be a red-packed forest and $f_r \in \mathfrak{F}_{\mathbf{R}}^I$ be a red-packed forest that can be right children of a node labeled by I ,

$$\mathbb{P}_{[]} := \mathbb{R}_{\epsilon}, \quad (5.4)$$

$$\mathbb{P}_{t_1, \dots, t_k} := \mathbb{P}_{t_k} \overline{\prec} (\mathbb{P}_{t_{k-1}} \overline{\prec} (\dots \overline{\prec} \mathbb{P}_{t_1}) \dots), \quad (5.5)$$

$$\mathbb{P}_{\text{Node}_{\mathbf{R}}(I, f_l = [\ell_1, \dots, \ell_g], f_r)} := \langle \mathbb{P}_{\ell_1}, \mathbb{P}_{\ell_2}, \dots, \mathbb{P}_{\ell_g}; \mathbb{P}_{\text{Node}_{\mathbf{R}}(I, [], f_r)} \rangle, \quad (5.6)$$

$$\mathbb{P}_{\text{Node}_{\mathbf{R}}(I, [], f_r)} := \Phi_I(\mathbb{P}_{f_r}). \quad (5.7)$$

Exemple 5.6.

$$\begin{aligned} \mathbb{P} \quad & \text{Diagram: } \begin{array}{c} \text{Root node } (1,3) \\ \text{--- } \text{--- } \text{--- } \end{array} \quad = \mathbb{P} \quad & \text{Diagram: } \begin{array}{c} \text{Root node } 1 \\ \text{--- } \text{--- } \text{--- } \end{array} \quad \overline{\prec} \quad \mathbb{P} \quad & \text{Diagram: } \begin{array}{c} \text{Root node } (1,3) \\ \text{--- } \text{--- } \text{--- } \end{array} \\ & \quad \text{Diagram: } \begin{array}{c} \text{Root node } 1 \\ \text{--- } \text{--- } \text{--- } \end{array} \quad = (\mathbb{P}_{(1)} \overline{\succ} \mathbb{P}_{(1)} - \mathbb{P}_{(1)} \overline{\prec} \mathbb{P}_{(1)}) \overline{\prec} \Phi_{1,3}(\mathbb{P}_{(1)}) \\ & \quad \text{Diagram: } \begin{array}{c} \text{Root node } 1 \\ \text{--- } \text{--- } \text{--- } \end{array} \quad = \mathbb{R}_{14342} + \mathbb{R}_{41342} + \mathbb{R}_{43142} + \mathbb{R}_{43412} - \mathbb{R}_{24341} - \mathbb{R}_{42341} - \mathbb{R}_{43241} \end{aligned}$$

More examples can be found in the annexes section with [table A.7](#) and [figures A.1 et A.3](#).

Théorème 5.7. *For all $n \in \mathbb{N}_{>0}$:*

1. $(\mathbb{P}_f)_{f \in \mathfrak{F}_{Rn}}$ is a basis of \mathbf{WQSym}^* ,
2. $(\mathbb{P}_t)_{t \in \mathfrak{N}_{Rn}}$ is a basis of \mathbf{Prim}^* ,
3. $(\mathbb{P}_t)_{t \in \mathfrak{N}_{Rn}}$ is a basis of \mathbf{TPrim}_n^* .

Démonstration. We proceed by a recursive induction on n to prove these three items. As $\dim(\mathbf{WQSym}_1^*) = \dim(\mathbf{Prim}_1^*) = \dim(\mathbf{TPrim}_1^*) = 1$ the base case is trivial. By [proposition 3.25](#), point 2 up to degree n implies point 1 up to degree n . Similarly, [théorème 3.27](#) shows that point 3 up to degree n implies point 2 up to degree n . By induction it is sufficient to show that points 1 et 2 up to degree $n - 1$ implies point 3 for n .

For all $k \in \mathbb{N}$, let π_k be the canonical projector on the homogeneous component of degree k of \mathbf{WQSym}^* . We define $\pi_{<k} := \sum_{i=0}^{k-1} \pi_i$. Fix $I = [i_1, \dots, i_p]$ with $p \leq n$ and u a packed word of size $n - p$. Notice that if $p = n$ we immediately have $u = \epsilon$ and $\Delta_{\overline{\mathcal{I}}}(\Phi_I(\mathbb{R}_\epsilon)) = 0$. We suppose now that $p < n$. By [équation \(3.59\)](#), in the half coproduct $\Delta_{\overline{\mathcal{I}}}(\Phi_I(\mathbb{R}_u))$ all the maximums must be in the left tensor factor, which therefore must be at least of degree i_p . By linearity, for all $x \in \mathbf{WQSym}_{n-p}^*$,

$$\Delta_{\overline{\mathcal{I}}}(\Phi_I(x)) = \left(\sum_{j=i_p}^{n-1} \Phi_I \circ \pi_j \otimes \pi_{n-1-j} \right) \circ \tilde{\Delta}(x). \quad (5.8)$$

Thanks to [corollaire 3.29](#) for $f = [r_1, \dots, r_d] \in \mathfrak{F}_{Rn-p}$, the coproduct $\tilde{\Delta}(\mathbb{P}_f)$ is computed by deconcatenation of forests. So if $f \in \mathfrak{F}_{Rn-p}^I$, in particular r_d is of weight at least $n - i_p + 1$ then for $j \geq i_p$ we have $n - 1 - j < \omega(r_d)$ so that all the terms in the previous sum vanish. A similar reasoning applies to $\Delta_{\overline{\mathcal{I}}}(\Phi_I(x))$ using the fact that $f \in \mathfrak{F}_{Rn-p}^I$ implies $1 \leq i_1 \leq \omega(r_1)$.

So for all $t = \text{Node}_{\mathbf{R}}(I, \emptyset, f_r) \in \mathfrak{N}_{Rn}$, we have that $\Delta_{\overline{\mathcal{I}}}(\mathbb{P}_t) = \Delta_{\overline{\mathcal{I}}}(\Phi_I(\mathbb{P}_{f_r})) = 0$ and $\Delta_{\overline{\mathcal{I}}}(\mathbb{P}_t) = \Delta_{\overline{\mathcal{I}}}(\Phi_I(\mathbb{P}_{f_r})) = 0$.

Moreover, by induction we have that $\{\mathbb{P}_f \mid f \in \mathfrak{F}_{Rn-p}^I\}$ are linearly independent as $\{\mathbb{P}_f \mid f \in \mathfrak{F}_{Rn-p}\}$ is a basis of \mathbf{WQSym}_{n-p}^* . Recall the description of \mathfrak{N}_{Rn} in [remarque 4.35](#) as a disjointed union of sets depending on I :

$$\mathfrak{N}_{Rn} = \bigsqcup_I \{\text{Node}_{\mathbf{R}}(I, \emptyset, f_r) \mid f_r \in \mathfrak{F}_{Rn-p}^I\} \quad (5.9)$$

Since Φ_I is injective on \mathbf{WQSym}_{n-p}^* then $\{\Phi_I(\mathbb{P}_f) \mid f \in \mathfrak{F}_{Rn-p}^I\}$ are linearly independent. According to [lemme 5.3](#), for all $f \in \mathfrak{F}_{Rn-p}^I$ we have $\Phi_I(\mathbb{P}_f) \in \text{Im}(\tau_I) \cap \mathbf{TPrim}_n^*$. Moreover, thanks to the direct sum of [équation \(5.3\)](#) :

$$\mathbf{TPrim}_n^* = \bigoplus_I \text{Im}(\tau_I) \cap \mathbf{TPrim}_n^*$$

and by definition of \mathbb{P} , in particular $\mathbb{P}_{\text{Node}_{\mathbf{R}}(I, \emptyset, f_r)} := \Phi_I(\mathbb{P}_{f_r})$, the family $\{\mathbb{P}_t \mid t \in \mathfrak{N}_{Rn}\}$ are linearly independent. Finally, by cardinalities of [théorème 4.40](#) it is a basis of \mathbf{TPrim}_n^* . \square

Remarque 5.8. The basis \mathbb{P} is indexed by red-packed forests ($\mathfrak{F}_{\mathbf{R}}$). We will also use red-skeletons (\mathfrak{F}_{Rske}) or packed words (\mathbf{PW}) as index thanks to the bijections of [remarque 4.33](#) and $F_{\mathbf{R}}$ of [définition 4.28](#).

5.2 Primal (Blue)

5.2.1 Decomposition through last letter and totally primitive elements

Définition 5.9. Let $i \in \mathbb{N}_{>0}$ and $\alpha \in \{\circ, \bullet\}$. We define a linear map $\Psi_{i^\alpha} : \mathbf{WQSym} \rightarrow \mathbf{WQSym}$ as follows : for all $n \in \mathbb{N}$ and $w \in \mathbf{PW}_n$,

$$\Psi_{i^\alpha}(\mathbb{Q}_w) := \begin{cases} \mathbb{Q}_{\psi_{i^\circ}(w)} & \text{if } \alpha = \circ \text{ and } 1 \leq i \leq \max(w) + 1, \\ \mathbb{Q}_{\psi_{i^\bullet}(w)} & \text{if } \alpha = \bullet \text{ and } 1 \leq i \leq \max(w), \\ 0 & \text{otherwise.} \end{cases} \quad (5.10)$$

Définition 5.10. Let $i \in \mathbb{N}_{>0}$ and $\alpha \in \{\circ, \bullet\}$. We define a projector $\tau_{i^\alpha} : \mathbf{WQSym} \rightarrow \mathbf{WQSym}$ as follows : for all $n \in \mathbb{N}$ and $w = w_1 \cdot w_2 \cdots w_n \in \mathbf{PW}_n$,

$$\tau_{i^\alpha}(\mathbb{Q}_w) := \begin{cases} \mathbb{Q}_w & \text{if } w_n = i \text{ and } \alpha = \bullet \text{ and } i \in [w_1, \dots, w_{n-1}], \\ \mathbb{Q}_w & \text{if } w_n = i \text{ and } \alpha = \circ \text{ and } i \notin [w_1, \dots, w_{n-1}], \\ 0 & \text{otherwise.} \end{cases} \quad (5.11)$$

These are orthogonal projectors in the sense that $\tau_{i^\alpha}^2 = \tau_{i^\alpha}$ and $\tau_{i^\alpha} \circ \tau_{j^\beta} = 0$ ($i \neq j$ or $\alpha \neq \beta$).

Lemme 5.11. For any i and α , we have $\text{Im}(\Psi_{i^\alpha}) = \text{Im}(\tau_{i^\alpha})$ where $\text{Im}(f)$ denotes the image of f .

Démonstration. For any i and α , the inclusion $\text{Im}(\Psi_{i^\alpha}) \subset \text{Im}(\tau_{i^\alpha})$ is automatic by definition of Ψ_{i^α} and τ_{i^α} and linearity. Indeed, for any $w \in \mathbf{PW}_n$ $\tau_{i^\alpha}(\Psi_{i^\alpha}(\mathbb{Q}_w)) = \Psi_{i^\alpha}(\mathbb{Q}_w)$.

For any i and any α , the inclusion $\text{Im}(\Psi_{i^\alpha}) \supset \text{Im}(\tau_{i^\alpha})$ is a consequence of [lemme 4.43](#) and linearity. Indeed, for any $w \in \mathbf{PW}$, if $\tau_{i^\alpha}(\mathbb{Q}_w) = \mathbb{Q}_w$ then $w_n = i$. With $w' = \text{pack}(w_1 \dots w_{n-1})$ we have $\Psi_{i^\alpha}(\mathbb{Q}_{w'}) = \mathbb{Q}_w = \tau_{i^\alpha}(\mathbb{Q}_w)$. \square

Lemme 5.12. For any i and α , the projection by τ_{i^α} of a totally primitive element is still a totally primitive element, so that $\tau_{i^\alpha}(\mathbf{TPrim}) = \text{Im}(\tau_{i^\alpha}) \cap \mathbf{TPrim}$. Moreover,

$$\mathbf{TPrim} = \bigoplus_{\alpha, i} \text{Im}(\tau_{i^\alpha}) \cap \mathbf{TPrim}. \quad (5.12)$$

Démonstration. Let w a packed word. We have $\Delta_{\overline{\mathcal{Z}}}(\tau_{i^\alpha}(\mathbb{Q}_w)) = (\tau_{i^\alpha} \otimes \text{Id}) \circ \Delta_{\overline{\mathcal{Z}}}(\mathbb{Q}_w)$ by definition of τ_{i^α} and $\Delta_{\overline{\mathcal{Z}}}$. Indeed, in $\Delta_{\overline{\mathcal{Z}}}(\mathbb{Q}_w)$, the decomposition can't be done under the last letter of w . By linearity, for all $p \in \mathbf{TPrim}$, we have $\Delta_{\overline{\mathcal{Z}}}(\tau_{i^\alpha}(p)) = (\tau_{i^\alpha} \otimes \text{Id}) \circ \Delta_{\overline{\mathcal{Z}}}(p) = 0$. The same argument works on the right so that $\tau_{i^\alpha}(p) \in \mathbf{TPrim}$. Moreover τ_{i^α} are orthogonal projectors so $\mathbf{TPrim} = \bigoplus_{\alpha, i} \tau_{i^\alpha}(\mathbf{TPrim}) = \bigoplus_{\alpha, i} \text{Im}(\tau_{i^\alpha}) \cap \mathbf{TPrim}$. \square

5.2.2 The new basis \mathcal{O}

Définition 5.13. Let $t_1, \dots, t_k, r \in \mathfrak{T}_B$, $f_r \in \{[], [r]\}$ and $f_l = [\ell_1, \dots, \ell_g] \in \mathfrak{F}_B$,

$$\mathbb{O}_{\|} := \mathbb{Q}_\epsilon, \quad (5.13)$$

$$\mathbb{O}_{t_1, \dots, t_k} := \mathbb{O}_{t_k} \preceq (\mathbb{O}_{t_{k-1}} \preceq (\dots \preceq \mathbb{O}_{t_1}) \dots), \quad (5.14)$$

$$\mathbb{O}_{\text{Node}_{\mathbf{B}}(i^\alpha, f_l = [\ell_1, \dots, \ell_q], f_r)} := \langle \mathbb{O}_{\ell_1}, \mathbb{O}_{\ell_2}, \dots, \mathbb{O}_{\ell_q}; \mathbb{O}_{\text{Node}_{\mathbf{B}}(i^\alpha, [], f_r)} \rangle, \quad (5.15)$$

$$\mathbb{O}_{\text{Node}_{\mathbf{B}}(i^\alpha, \lceil, f_r)} := \Psi_{i^\alpha}(\mathbb{O}_{f_r}). \quad (5.16)$$

Exemple 5.14.

$$\begin{aligned}
& \emptyset = \emptyset \quad \emptyset \subseteq \emptyset \\
& = (\emptyset \cup \emptyset \cup \emptyset - \emptyset \cup \emptyset \cup \emptyset) \subseteq \Psi_2(\emptyset) \\
& = \mathbb{Q}_{34122} + \mathbb{Q}_{24133} + \mathbb{Q}_{14233} + \mathbb{Q}_{43212} + \mathbb{Q}_{42313} + \mathbb{Q}_{41323} \\
& \quad - \mathbb{Q}_{34212} - \mathbb{Q}_{24313} - \mathbb{Q}_{14323} - \mathbb{Q}_{43122} - \mathbb{Q}_{42133} - \mathbb{Q}_{41233}
\end{aligned}$$

More examples can be found in the annexes section with table A.7 et figure A.4.

Théorème 5.15. *For all $n \in \mathbb{N}_{>0}$:*

1. $(\mathbb{O}_f)_{f \in \mathfrak{F}_{B_n}}$ is a basis of \mathbf{WQSym}_n ,
 2. $(\mathbb{O}_t)_{t \in \mathfrak{T}_{B_n}}$ is a basis of \mathbf{Prim}_n ,
 3. $(\mathbb{O}_t)_{t \in \mathfrak{N}_{B_n}}$ is a basis of \mathbf{TPrim}_n .

Démonstration. The proof structure is the same as the one of théorème 5.7 except for some statements that are exchanged as we can see in this table :

équation (3.59)	équation (3.56)	left coproduct in basis \mathbb{R} and \mathbb{Q} .
remarque 4.35	remarque 4.70	$\mathfrak{N}_{Rn} = \bigsqcup_I \{\text{Node}_R(I, \emptyset, f_r) f_r \in \mathfrak{F}_{Rn-p}^I\}$ $\mathfrak{N}_{Bn} = \bigsqcup_{i,\alpha} \{\text{Node}_B(i^\alpha, \emptyset, f_r) f_r \in \mathfrak{F}_{Bn-p}^{i^\alpha}\}.$
lemme 5.3	lemme 5.11	$\text{Im}(\Phi_I) = \text{Im}(\tau_I)$ and $\text{Im}(\Psi_{i^\alpha}) = \text{Im}(\tau_{i^\alpha})$.
lemme 5.4	lemme 5.12	$\mathbf{TPrim}^* = \bigoplus_I \text{Im}(\tau_I) \cap \mathbf{TPrim}^*$. $\mathbf{TPrim} = \bigoplus_{\alpha,i} \text{Im}(\tau_{i^\alpha}) \cap \mathbf{TPrim}$.
théorème 4.40	théorème 4.76	$\dim(\mathbf{TPrim}_n^*) = \#\mathfrak{N}_{Rn}$ and $\dim(\mathbf{TPrim}_n) = \#\mathfrak{N}_{Bn}$.

Remarque 5.16. The same remark as [remarque 5.8](#) can be done on the basis \mathbb{O} . Indeed, it is defined with blue-packed forests (\mathfrak{F}_B), it is nevertheless possible to use the blue-skeletons ($\mathfrak{F}_{B\text{ske}}$) or packed words(\mathbf{PW}) thanks to the bijections of [remarque 4.68](#) and F_B of [définition 4.63](#).

Chapitre 6

Isomorphism between \mathbf{WQSym} and \mathbf{WQSym}^*

According to corollaire 3.28 \mathbf{WQSym} (resp. \mathbf{WQSym}^*) is freely generated as a dendriform algebra by \mathbf{TPrim} (resp. \mathbf{TPrim}^*). Therefore, any linear isomorphism between \mathbf{TPrim} and \mathbf{TPrim}^* would lead to a bidendriform isomorphism between \mathbf{WQSym} and its dual. Thanks to the two bases \mathbb{P} and \mathbb{O} any graded bijection between red-irreducible and blue-irreducible packed words leads to such an isomorphism. We first make explicit how this is done. Then the bijection is actually obtained as the restriction to red-irreducibles of an involution on all packed words. The definition of the bijection require a new kind of forest mixing red and blue factorizations, namely bicolored-packed forests.

6.1 A combinatorial solution to an algebraic problem

In this chapitre 6, we use the skeleton representation for bases \mathbb{P} and \mathbb{O} as said in remarques 5.8 et 5.16. Moreover we fix a graded bijection μ between red-irreducible and blue-irreducible packed words.

Définition 6.1. Recall that $(\mathbb{P}_t)_{t \in \mathfrak{N}_{R_n}}$ is a basis of \mathbf{TPrim}_n^* (théorème 5.7) and $(\mathbb{O}_t)_{t \in \mathfrak{N}_{B_n}}$ is a basis of \mathbf{TPrim}_n (théorème 5.15). By linearity, setting

$$t' := \textcircled{\mu(v)} \in \mathfrak{N}_B, \quad \text{and} \quad M_\mu(\mathbb{P}_t) := \mathbb{O}_{t'}. \quad (6.1)$$

for all $t = \textcircled{v} \in \mathfrak{N}_R$, defines a linear isomorphism between the vector spaces \mathbf{TPrim}_n^* and \mathbf{TPrim}_n .

Définition 6.2. We define σ_μ as the extension of μ from red-skeleton to blue-skeleton forests by :

$$\forall f = [t_1, \dots, t_k] \in \mathfrak{F}_R, \quad \sigma_\mu(f) := [\sigma_\mu(t_1), \dots, \sigma_\mu(t_k)] \quad (6.2)$$

$$\forall t = \text{Node}_R(v, f_\ell, []) \in \mathfrak{T}_R, \quad \sigma_\mu(t) := \text{Node}_B(\mu(v), \sigma_\mu(f_\ell), []). \quad (6.3)$$

Définition 6.3. We denote Σ_μ the unique bidendriform isomorphism from \mathbf{WQSym}^* to \mathbf{WQSym} which verify for all $f \in \mathfrak{F}_R$:

$$\Sigma_\mu(\mathbb{P}_f) := \mathbb{O}_{\sigma_\mu(f)} \quad (6.4)$$

The existence and unicity are guaranteed by [corollaire 3.28](#).

Exemple 6.4.

$$\sigma_\mu \left(\overbrace{\begin{array}{c} w_1 \\ \swarrow \quad \searrow \\ w_2 \quad w_3 \end{array}}^{f_R} \right) = \overbrace{\begin{array}{c} \mu(w_1) \\ \swarrow \quad \searrow \\ \mu(w_2) \quad \mu(w_3) \end{array}}^{f_B} \quad \Sigma_\mu(\mathbb{P}_{f_R}) = \mathbb{O}_{f_B}$$

In the following example, we take $\mu(212) := 122$ and $\mu(w) := w$ for other words with a size less than 3 as they are simultaneously red-irreducible and blue-irreducible packed words.

Here are all red and blue-irreducible packed words of size 1, 2 and 3 :

$$\text{Red : } 1, \quad 11, \quad 132 \quad 121 \quad 212 \quad 111, \quad \text{Blue : } 1, \quad 11, \quad 132 \quad 121 \quad 122 \quad 111.$$

Exemple 6.5.

$$\sigma_\mu \left(\begin{array}{c} (212) \\ \quad \quad \quad 1 \end{array} \right) = \begin{array}{c} (122) \\ \quad \quad \quad 1 \end{array}$$

These two forests are the same as those used in [exemples 5.6 et 5.14](#). So we have here the first example of the isomorphism from the basis \mathbb{R} to the basis \mathbb{Q} :

$$\Sigma_\mu \left(\begin{array}{c} \mathbb{R}_{14342} + \mathbb{R}_{41342} + \mathbb{R}_{43142} + \mathbb{R}_{43412} \\ -\mathbb{R}_{24341} - \mathbb{R}_{42341} - \mathbb{R}_{43241} - \mathbb{R}_{43421} \end{array} \right) = \left(\begin{array}{c} \mathbb{Q}_{34122} + \mathbb{Q}_{24133} + \mathbb{Q}_{14233} + \mathbb{Q}_{43212} \\ + \mathbb{Q}_{42313} + \mathbb{Q}_{41323} - \mathbb{Q}_{34212} - \mathbb{Q}_{24313} \\ - \mathbb{Q}_{14323} - \mathbb{Q}_{43122} - \mathbb{Q}_{42133} - \mathbb{Q}_{41233} \end{array} \right)$$

We now have a construction of a bidendriform isomorphism for any graded bijection μ between red-irreducible and blue-irreducible packed words.

6.2 Full decomposition of packed words into bicolored forests

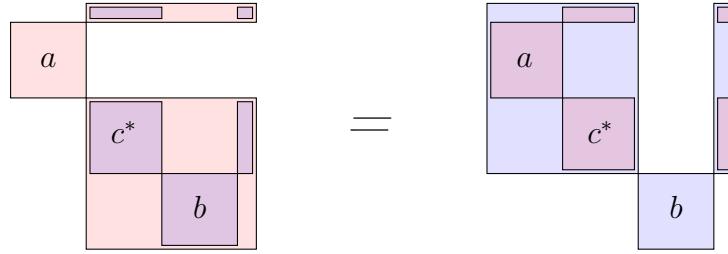
In order to define a bijection between red-irreducible and blue-irreducible packed words, we need a new kind of forests that mixes up the red and blue factorizations. More precisely,

we will recursively alternate these factorizations. We start with an unexpected lemma which implies that starting by red or blue does not matter.

Lemme 6.6. *For all $a, b, c \in \mathbf{PW}$, with $c \neq \epsilon$, the following relations hold :*

$$a \blacktriangleright (b \blacktriangle c) = b \blacktriangle (a \blacktriangleright c) \quad \text{and} \quad a \blacktriangleright 1 = a \blacktriangle 1, \quad (6.5)$$

where 1 is the packed word of size 1.



Démonstration. Let $a \in \mathbf{PW}$ and 1 the packed word of size 1, by définitions 4.5 et 4.44 $a \blacktriangleright 1 = a \setminus 1$ and $a \blacktriangle 1 = a \setminus 1$.

Let $a, b, c \in \mathbf{PW}$, with $c = c_1 \cdots c_n$ of size $n > 0$. We start by assuming that $c_n \neq \max(c)$ which implies that $c = \phi_I(\psi_{i^\alpha}(c^*)) = \psi_{i^\alpha}(\phi_I(c^*))$ with I, i, α, c^* unique by lemmes 4.4 et 4.43. With this relation we can deduce :

$$\begin{aligned} a \blacktriangleright (b \blacktriangle c) &= a \blacktriangleright (b \blacktriangle \psi_{i^\alpha}(\phi_I(c^*))) \\ &= a \blacktriangleright (\psi_{i+\max(b)^\alpha}(\phi_I(c^*)/b)) \\ &= a \blacktriangleright (\psi_{i+\max(b)^\alpha}(\phi_I(c^*/b))) \\ &= a \blacktriangleright (\phi_I(\psi_{i+\max(b)^\alpha}(c^*/b))) \\ &= \phi_{I+|a|}(a/\psi_{i+\max(b)^\alpha}(c^*/b)) \\ &= \phi_{I+|a|}(\psi_{i+\max(b)^\alpha}(a/c^*/b)) \end{aligned}$$

$$\begin{aligned} b \blacktriangle (a \blacktriangleright c) &= b \blacktriangle (a \blacktriangleright \phi_I(\psi_{i^\alpha}(c^*))) \\ &= b \blacktriangle (\phi_{I+|a|}(a/\psi_{i^\alpha}(c^*))) \\ &= b \blacktriangle (\phi_{I+|a|}(\psi_{i^\alpha}(a/c^*))) \\ &= b \blacktriangle (\psi_{i^\alpha}(\phi_{I+|a|}(a/c^*))) \\ &= \psi_{i+\max(b)^\alpha}(\phi_{I+|a|}(a/c^*)/b) \\ &= \psi_{i+\max(b)^\alpha}(\phi_{I+|a|}(a/c^*/b)) \\ &= \phi_{I+|a|}(\psi_{i+\max(b)^\alpha}(a/c^*/b)). \end{aligned}$$

The case where $c_n = \max(c)$ can be decomposed into different particular cases. In each of these cases, it is possible to find a relation with two different writings of c that begin with ϕ or ψ just like $c = \phi_I(\psi_{i^\alpha}(c^*)) = \psi_{i^\alpha}(\phi_I(c^*))$. These cases with the associated relation are :

- the case where c is the packed word 1 then $c = \phi_1(\epsilon) = \psi_{1^\circ}(\epsilon)$,
- the case where c is of the form $c' \setminus 1$ then $c = \phi_1(c') = \psi_{1^\circ}(c')$,
- the more general case where there is more than 1 maximum including the one at the end then $c = \phi_{I \cdot c_n}(c^*) = \psi_{c_n^\bullet}(\phi_I(c^*))$.

In each of these cases it is possible to prove with a similar method that $a \blacktriangleright (b \blacktriangle c) = b \blacktriangle (a \blacktriangleright c)$. \square

Exemple 6.7. Here are some examples of this relation :

$$\begin{aligned} 1 \blacktriangleright (1 \blacktriangle 1) &= 213 = 1 \blacktriangle (1 \blacktriangleright 1), \\ 11 \blacktriangleright (12 \blacktriangle 2111) &= 44533123 = 12 \blacktriangle (11 \blacktriangleright 2111), \\ 11 \blacktriangleright (21 \blacktriangle 123) &= 5534216 = 21 \blacktriangle (11 \blacktriangleright 123), \\ 1 \blacktriangleright (112 \blacktriangle 3132) &= 56361124 = 112 \blacktriangle (1 \blacktriangleright 3132). \end{aligned}$$

Définition 6.8. Let w be an irreducible packed word. Let $w = x \blacktriangleright u$ be the red-factorization of w and let $u = y \blacktriangle z$ be the blue-factorization of u . Then $w = x \blacktriangleright (y \blacktriangle z)$ is called the **red-blue-factorization** of w . Symmetrically we define $w = y' \blacktriangle (x' \blacktriangleright z')$ the **blue-red-factorization** of w .

Lemme 6.9. Let $w = x \blacktriangleright (y \blacktriangle z)$ be the red-blue-factorization and let $w = y' \blacktriangle (x' \blacktriangleright z')$ be the blue-red-factorization of an irreducible packed word w .

With these two factorizations, we have that $z = z'$ and it is both red-irreducible and blue-irreducible packed word. Moreover,

- either $z = z' = 1$, $y = x' = \epsilon$ and $x = y'$
- or $x = x'$, $y = y'$.

Exemple 6.10. Here are some examples of red-blue-factorization and blue-red-factorization :

$$\begin{aligned} 12 \blacktriangleright (\epsilon \blacktriangle 1) &= 213 = 12 \blacktriangle (\epsilon \blacktriangleright 1), \\ 11 \blacktriangleright (12 \blacktriangle 1211) &= 44353123 = 12 \blacktriangle (11 \blacktriangleright 1211), \\ 553421 \blacktriangleright (\epsilon \blacktriangle 1) &= 5534216 = 553421 \blacktriangle (\epsilon \blacktriangleright 1), \\ 1 \blacktriangleright (112 \blacktriangle 3132) &= 56361124 = 112 \blacktriangle (1 \blacktriangleright 3132). \end{aligned}$$

Démonstration. We start by proving the case where $z = z' = 1$, $y = x' = \epsilon$ and $x = y'$. Let w' be an irreducible packed word and $w = w' \setminus 1$. We have that $w = w' \blacktriangleright 1$ is the red-factorization of w and $w = w' \blacktriangle 1$ is the blue-factorization of w . In this case we immediately have that $w = w' \blacktriangleright (\epsilon \blacktriangle 1)$ is the red-blue-factorization of w and that $w = w' \blacktriangle (\epsilon \blacktriangleright 1)$ is the blue-red-factorization of w .

Now let w be an irreducible packed word of size n that cannot be written as $w' \setminus 1$. In other words, there is a maximum strictly before the last letter of w ($\exists i < n, w_i = \max(w)$).

We define the two sets of triplet of packed words that verify equations of the factorizations for w :

$$\begin{aligned} S_{RB}(w) &:= \{(a, b, c) \in \mathbf{PW}, c \neq \epsilon, w = a \blacktriangleright (b \blacktriangle c)\}, \\ S_{BR}(w) &:= \{(a, b, c) \in \mathbf{PW}, c \neq \epsilon, w = b \blacktriangle (a \blacktriangleright c)\}. \end{aligned}$$

Thanks to [lemme 6.6](#) these two sets are equal, we define $S(w) := S_{RB}(w) = S_{BR}(w)$.

In the red-blue-factorization $w = x \blacktriangleright (y \blacktriangle z)$, we maximize the size of x , then we maximize the size of y in the remaining word. In the blue-red-factorization we commute the order of maximizations. We will characterize $S(w)$ and see the limit of the two maximizations to prove that they can commute.

Let w^* be the packed word comming from w where the last letter and all occurences of the maximum are removed and let $I = [i_1, \dots, i_p]$, i, α such that $w = \psi_{i^\alpha}(\phi_I(w^*))$. By hypothesis, we have that $I \neq \emptyset$. Let $w^* = w_1 / \dots / w_k$ be the global descent decomposition of w^* . Let ℓ be the maximum such that $|w_1 / \dots / w_\ell| \leq i_1$, i_1 being the position of the first maximum of w . Let r be the minimum such that $|w_r / \dots / w_k| \leq n - i_p - 1$, i_p being the position of the last maximum of w before the last letter. As $i_1 \leq i_p$ by definition, we have that $\ell < r$. We can caracterize the set $S(w)$:

$$\begin{aligned} S(w) = \{ &(a = w_1 / \dots / w_{r^0}, \\ &b = w_{\ell^0} / \dots / w_k, \\ &c = \psi_{(i-\max(b))^\alpha}(\phi_{I-|a|}(w_{r^0+1} / \dots / w_{\ell^0-1})), \\ &\text{with } r^0 \leq r \text{ and } \ell \leq \ell^0 \}. \end{aligned}$$
□

Here are all packed words that are both red-irreducible and blue-irreducible of size less than 4 :

1,	11,	111	121	132,								
1111	1121	1132	1211	1212	1221	1231	1232	1243	1312	1321		
1322	1323	1332	1342	1423	1432	2121	2122	2132	2143	3132		

n	1	2	3	4	5	6	7	8	9
$\mathfrak{i}_n \in \mathbf{PW}_n$	1	1	3	22	196	2 008	23 184	297 456	4 199 216
$\mathfrak{i}_n \in \mathfrak{S}_n$	1	0	1	5	32	236	1 951	17 827	178 418

TABLE 6.1 – Number of both red-and-blue-irreducible packed words and permutations.

Recall our notations for Hilbert series of an algebra A ,

$$\mathcal{A}(z) := \sum_{n=1}^{+\infty} \dim(A_n) z^n, \quad (6.6)$$

$$\mathcal{P}(z) := \sum_{n=1}^{+\infty} \dim(\text{Prim}(A_n)) z^n, \quad (6.7)$$

$$\mathcal{T}(z) := \sum_{n=1}^{+\infty} \dim(\text{TPrim}(A_n)) z^n. \quad (6.8)$$

Recall the relations between these series :

$$\mathcal{P} = \mathcal{A}/(1 + \mathcal{A}) \text{ or equivalently } \mathcal{A} = \mathcal{P}/(1 - \mathcal{P}) \text{ (see proposition 3.25),}$$

$$\mathcal{T} = \mathcal{A}/(1 + \mathcal{A})^2 \text{ or equivalently } \mathcal{P} = \mathcal{T}(1 + \mathcal{A}) \text{ (see théorème 3.27).}$$

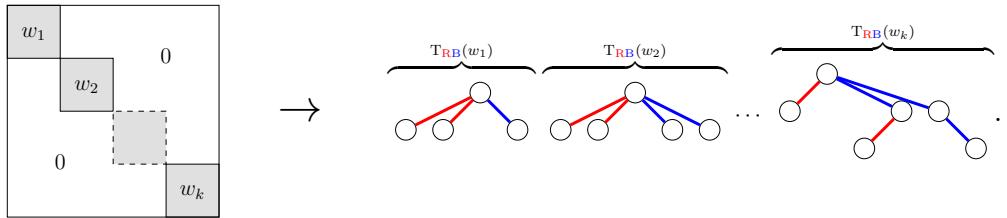
If we define the serie $\mathcal{I} = \sum_{n=1}^{+\infty} i_n z^n$ where i_n is the number of both red-and-blue-irreducible words of size n , then we have the following relation :

$$\mathcal{I} = \mathcal{A}/(1 + \mathcal{A})^3 + z\mathcal{P} \text{ or equivalently } \mathcal{T} = (\mathcal{I} - z)(1 + \mathcal{A}) + z.$$

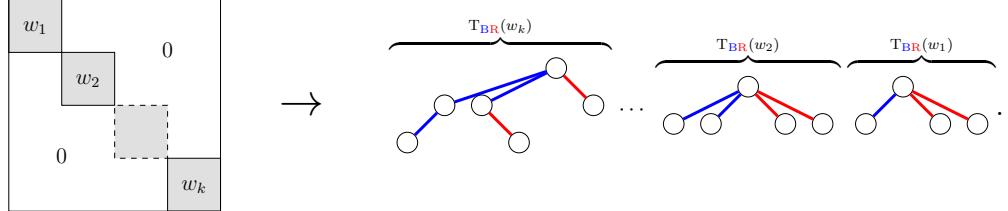
So far we have seen red-biplane trees and blue-biplane trees. In this chapter we define red-blue-biplane trees and blue-red-biplane trees, the edges of these trees are of two different colors and the labels are red-and-blue-irreducible packed words. We denote by $\text{Node}_{\text{RB}}(x, f_\ell, f_r)$ (resp. $\text{Node}_{\text{BR}}(x, f_\ell, f_r)$) the biplane tree whose edges between the root and the left forest f_ℓ are red (resp. blue) and edges between the root and the right forest f_r are blue (resp. red).

Définition 6.11. The bicolored forests $F_{\text{RB}}(w)$ and $F_{\text{BR}}(w)$ (resp. trees $T_{\text{RB}}(w)$ and $T_{\text{BR}}(w)$) associated to a packed word (resp. irreducible packed word) w are defined in a mutual recursive way as follows :

- $F_{\text{RB}}(\epsilon) = F_{\text{BR}}(\epsilon) = []$ (empty forest),
- for any packed word w , let $w = w_1/w_2/\dots/w_k$ be the global descent decomposition, then $F_{\text{RB}}(w) := [T_{\text{RB}}(w_1), T_{\text{RB}}(w_2), \dots, T_{\text{RB}}(w_k)]$,

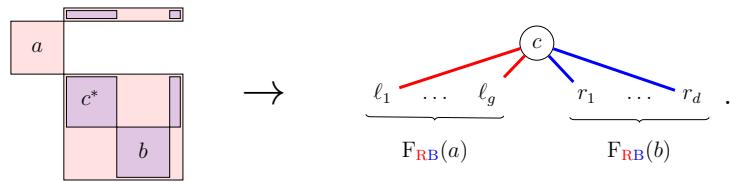


- for any packed word w , let $w = w_1/w_2/\dots/w_k$ be the global descent decomposition, then $F_{BR}(w) := [T_{BR}(w_k), T_{BR}(w_{k-1}), \dots, T_{BR}(w_1)]$.

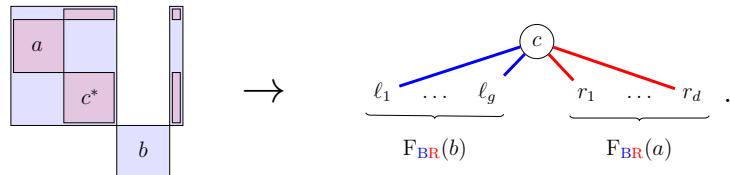


(notice the same inversion as in [définitions 4.58 et 4.63](#) for F_{BR} .)

- for any irreducible packed word w , let $w = a \blacktriangleright (b \blacktriangle c)$ be the red-blue-factorization, then $T_{RB}(w) := \text{Node}_{RB}(c, F_{RB}(a), F_{RB}(b))$.



- for any irreducible packed word w , let $w = b \blacktriangle (a \blacktriangleright c)$ be the blue-red-factorization, then $T_{BR}(w) := \text{Node}_{BR}(c, F_{BR}(b), F_{BR}(a))$.

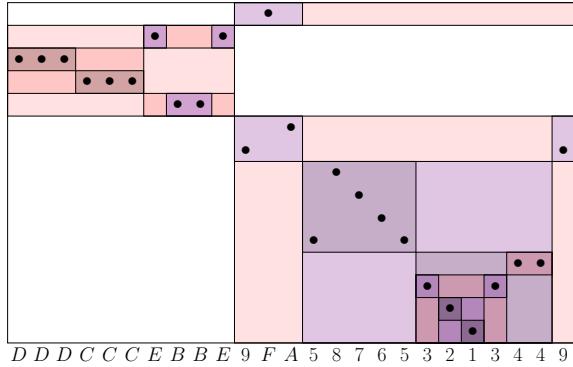


Exemple 6.12. For this example we write the word w in hexadecimal in order to have a big and clear example, let $w = DDDCCCEBBE9FA587653213449$. The word w is irreducible so there is only one tree in the forest. To have $T_{RB}(w)$, we start by the blue-red-factorization,

$$w = \underline{DDDC}CCEBBE\ 9FA\ 58765321344\ 9 = \underline{3332224114} \blacktriangleright (\underline{58765\ 321344} \blacktriangle 1321).$$

Then we decompose each sub-word according to their global descents and do blue-red-factorizations recursively until we have only both red-irreducible and blue-irreducible packed words :

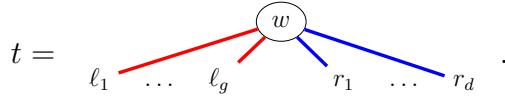
$$\begin{aligned} w &= \underline{3332224114} \blacktriangleright (\underline{58765\ 321344} \blacktriangle 1321), \\ &= \underline{333222\ 4\ 11\ 4} \blacktriangleright ((\underline{14321}/\underline{3213\ 44}) \blacktriangle 1321), \\ &\quad [\dots] \\ &= ((111/111) \blacktriangleright (11 \blacktriangle 11)) \blacktriangleright ((14321/((1/1) \blacktriangle 11) \blacktriangleright 11)) \blacktriangle 1321 \end{aligned}$$



$$\begin{aligned}
 T_{\text{RB}}(w) &= \frac{\text{F}_{\text{RB}}(3332224114)}{\text{F}_{\text{RB}}(1321)} = \frac{\text{F}_{\text{RB}}(58765321344)}{\text{F}_{\text{RB}}(1321)} = \frac{\text{T}_{\text{RB}}(3332224114)}{\text{T}_{\text{RB}}(14321)} = \frac{\text{T}_{\text{RB}}(321344)}{\text{T}_{\text{RB}}(1321)} \\
 &[\dots] \\
 &= \frac{\text{F}_{\text{RB}}(1321)}{\text{F}_{\text{RB}}(111)} = \frac{\text{F}_{\text{RB}}(111)}{\text{F}_{\text{RB}}(11)} = \frac{\text{F}_{\text{RB}}(11)}{\text{F}_{\text{RB}}(1)} = \frac{\text{F}_{\text{RB}}(1)}{\text{F}_{\text{RB}}(1)}
 \end{aligned}$$

More examples can be found in the annexes section with [tables A.1 à A.6](#).

Définition 6.13. There are two types of bicolored trees, the only difference is that colors **red** and **blue** are inverted. Let t be a labeled biplane tree. We write $t = \text{Node}_{\text{RB}}(w, f_\ell, f_r)$ where $w \in \mathbf{PW}$, $f_\ell = [\ell_1, \dots, \ell_g]$ is the left forest of t and $f_r = [r_1, \dots, r_d]$ is the right forest of t . We depict t as follows :



We say that t is a **red-blue-packed tree** if it satisfies :

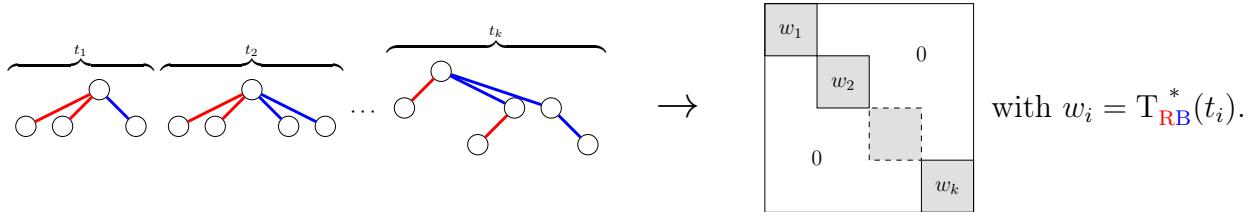
$$\left\{ \begin{array}{l} w = 1, \\ f_r = \emptyset, \\ f_\ell \text{ is a red-blue-packed forest.} \end{array} \right. \text{ or } \left\{ \begin{array}{l} w \neq 1 \text{ is red-irreducible and blue-irreducible,} \\ f_r \text{ and } f_\ell \text{ are red-blue-packed forest.} \end{array} \right.$$

Définition 6.14. The **weight** of a bicolored-packed tree is the sum of the size of packed words in the nodes.

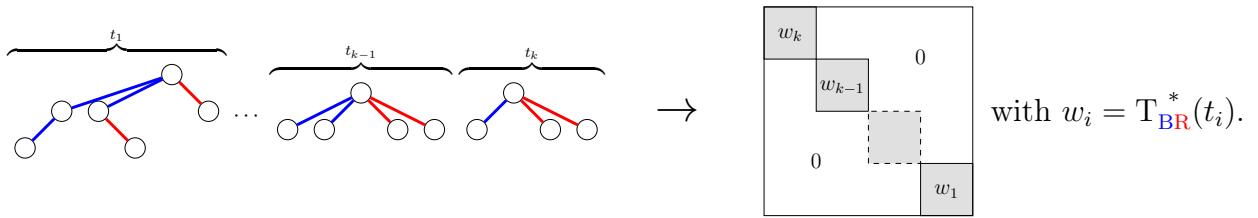
We have already done it four times ([définitions 4.26, 4.36, 4.61 et 4.71](#)) and we will do it one last time, to prove that $\text{F}_{\text{RB}}, \text{T}_{\text{RB}}, \text{F}_{\text{BR}}, \text{T}_{\text{BR}}$ are bijections, we define $\text{F}_{\text{RB}}^*, \text{T}_{\text{RB}}^*, \text{F}_{\text{BR}}^*, \text{T}_{\text{BR}}^*$ and prove that they are the inverse maps.

Définition 6.15. We define here the maps F_{RB}^* , T_{RB}^* , F_{BR}^* , T_{BR}^* that transforms bicolored-packed forests and trees into packed words. We reverse all instructions of [définition 6.13](#) as follows :

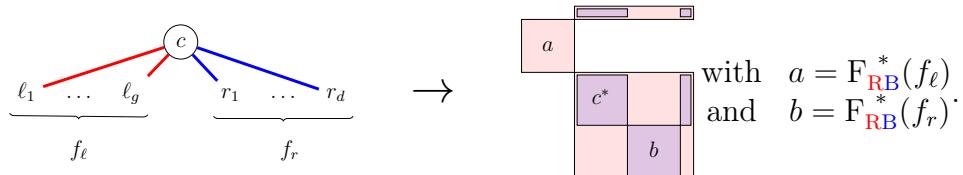
- $F_{RB}^*([]) = F_{BR}^*([]) = \epsilon$ (empty packed word),
 - for any red-blue-packed forest $f = [t_1, t_2 \dots, t_k]$, we have $F_{RB}^*(f) := [T_{RB}^*(t_1), T_{RB}^*(t_2), \dots, T_{RB}^*(t_k)]$.



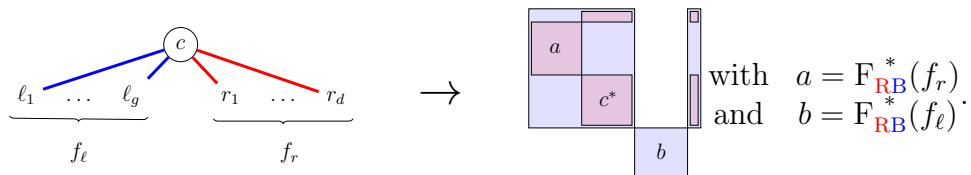
- for any blue-red-packed forest $f = [t_1, t_2 \dots, t_k]$, we have $F_{\text{BR}}^*(f) := [T_{\text{BR}}^*(t_k), T_{\text{BR}}^*(t_{k-1}), \dots, T_{\text{BR}}^*(t_1)]$.



- for any red-blue-packed tree $t = \text{Node}_{\text{RB}}(c, f_\ell, f_r)$, we have $\text{T}_{\text{RB}}^*(t) := \text{F}_{\text{RB}}^*(f_\ell) \blacktriangleright (\text{F}_{\text{RB}}^*(f_r) \blacktriangle c)$.



- for any blue-red-packed tree $t = \text{Node}_{\text{BR}}(c, f_\ell, f_r)$, we have $\text{T}_{\text{BR}}^*(t) := \text{F}_{\text{BR}}^*(f_r) \blacktriangle (\text{F}_{\text{BR}}^*(f_\ell) \blacktriangleright c)$.



Théorème 6.16. *The maps F_{RB} and F_{RB}^* (resp. T_{RB} and T_{RB}^*) are two inverse bijections between packed words of size n and red-blue-packed forests (resp. irreducible packed words and red-blue-packed trees) of weight n . That is to say $F_{RB}^{-1} = F_{RB}^*$ and $T_{RB}^{-1} = T_{RB}^*$. We have the same result with inversions of red and blue.*

Démonstration. It is simple to prove by induction on the size of the trees that domain and codomain are as announced and that the functions are inverse to each other. The proof is similar to the proofs of [lemmes 4.27 et 4.62](#) et [théorèmes 4.38 et 4.73](#) using [définition 6.11](#) of F_{RB} , T_{RB} , F_{BR} , T_{BR} , [définition 6.13](#) of bicolored-packed forests and trees and [définition 6.15](#) of F_{RB}^* , T_{RB}^* , F_{BR}^* , T_{BR}^* . \square

Remarque 6.17. We now have two new families of forests \mathfrak{F}_{RB} and \mathfrak{F}_{BR} that are in bijection with packed words and therefore in bijection with red-packed and blue-packed forests. As in [remarques 5.8 et 5.16](#), this gives us two other ways to index bases \mathbb{O} and \mathbb{P} of \mathbf{WQSym} and \mathbf{WQSym}^* .

6.3 An involution on packed words

We are now in position to define a bijection between red-irreducible and blue-irreducible packed words. This bijection is actually the restriction of an involution defined on all packed words. Precisely, we will define two transformations on bicolored forests.

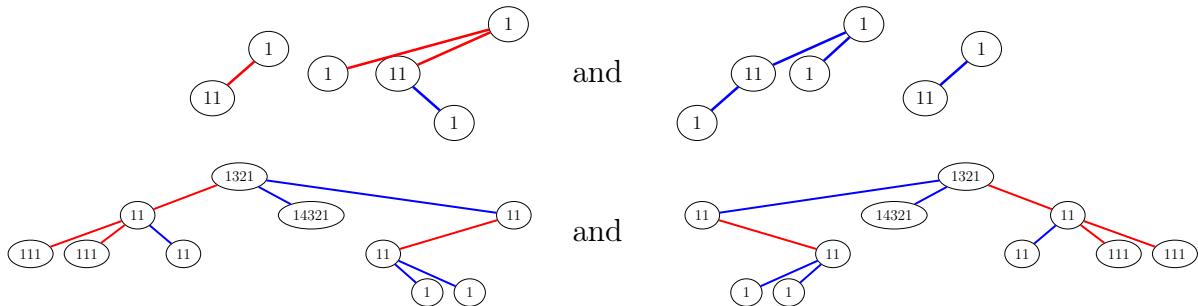
We need to define the notion of mirror transformation of bicolored-packed forests and trees. This transformation is defined from a red-blue to blue-red or from blue-red to red-blue, so in the notations we will use XY instead of RB or BR to point out where the swap is made.

Définition 6.18. The mirror transformation of a bicolored-packed forest $f = [t_1, \dots, t_k]$ is given by $\tilde{f} := [\tilde{t}_k, \dots, \tilde{t}_1]$ where \tilde{t}_i is the mirror transformation of t_i recursively defined as follows. For any $t = \text{Node}_{XY}(z, f_\ell, f_r)$ then

$$\tilde{t} := \begin{cases} \text{Node}_{YX}(z, \tilde{f}_r, \tilde{f}_\ell) & \text{if } z \neq 1, \\ \text{Node}_{YX}(1, \tilde{f}_\ell, \emptyset) & \text{if } z = 1. \end{cases}$$

Note that when $z \neq 1$, the left and right forests are swapped whereas they are not when $z = 1$. But in the latter case, we have necessarily $f_r = \tilde{f}_r = \emptyset$. These two cases correspond to the two cases of [définition 6.13](#) so the mirror transformation of a red-blue-packed forest is indeed a blue-red-packed forest.

Exemple 6.19. Here are two examples of mirror transformations.



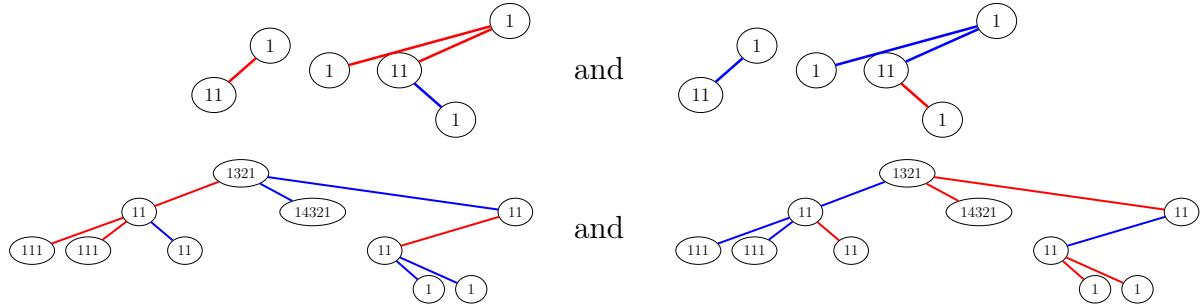
Proposition 6.20. *For all packed words w , both associated bicolored-packed forests $F_{\text{RB}}(w)$ and $\widehat{F}_{\text{BR}}(w)$ are mirror image of each other.*

Démonstration. The proof is a computation of mirror transformation (définition 6.18) on each items of définition 6.11 of F_{RB} . For the first three items the computation of $\tilde{f} := [\tilde{t}_k, \dots, \tilde{t}_1]$ is sufficient. Thanks to the relation of lemme 6.9 ($x \blacktriangleright (y \blacktriangle z) = y \blacktriangle (x \blacktriangleright z)$) in the case $z \neq 1$) the two remaining items are also simple computation of \tilde{t} . \square

Définition 6.21. The color swap of a bicolored-packed forest $f = [t_1, \dots, t_k]$ is given by $\widehat{f} := [\widehat{t}_k, \dots, \widehat{t}_1]$ where \widehat{t}_i is the color swap of t_i recursively defined as follows. For any $t = \text{Node}_{XY}(z, f_\ell, f_r)$ then $\widehat{t} := \text{Node}_{YX}(z, \widehat{f}_\ell, \widehat{f}_r)$.

In other words, it is a recoloration of each edges using the other color. Every blue edges become red and vice versa.

Exemple 6.22. Here are two examples of color swaps.



More examples can be found in the annexes section with table A.7.

When we focus on the packed words associated to these forest, the color swap correspond to the swap of the two operations \blacktriangleright and \blacktriangle in a bicolored-factorization. More precisely, if w is an irreducible packed word and $w = x \blacktriangleright (y \blacktriangle z)$ is the red-blue-factorization of w , then the color swap on the associated forest correspond to $w' = x \blacktriangle (y \blacktriangleright z)$.

Lemme 6.23. *Mirror transformation and color swap commute. It means that for all bicolored-packed forest f , we have $\widehat{(\widehat{f})} = (\widehat{\widehat{f}})$.*

Démonstration. The proof is immediate. Indeed, the definition of mirror transformation is independant of color swap and symmetrically, the color swap is independant of the tree shape. \square

Corollaire 6.24. *The diagram on figure 6.1 is commutative. So $\widehat{w} := F_{\text{RB}}^{-1}(\widehat{F}_{\text{BR}}(w))$ is an involution on packed words.*

Démonstration. Thanks to proposition 6.20 et lemme 6.23 the diagram is immediately commutative. The mirror transformation and the color swap are independant involutions so the conjunction is an involution. \square

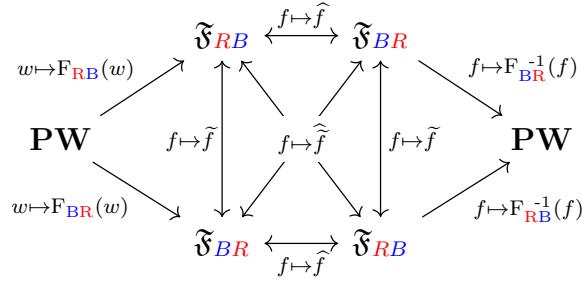


FIGURE 6.1 – Commutative diagram of maps on bicolored-packed forests

Corollaire 6.25. *The application $w \mapsto \hat{w}$ send blue (resp. red) irreducibles packed words to red (resp. blue) irreducibles packed words.*

Some examples can be found in th annexes section with [tables A.8 et A.9](#).

Démonstration. If w is a red-irreducible packed word, then the red-blue-factorization of w is of the form $w = \epsilon \blacktriangleright (y \blacktriangle z)$. Then the color swap correspond to the words $w' = \epsilon \blacktriangle (y \blacktriangleright z)$ which is blue-irreducible. \square

6.4 Main theorem

In [section 6.1](#) we fixed a graded bijection μ between red-irreducible and blue-irreducible packed words. After that, we extend it to all red-skeleton forests as σ_μ . We finished by defining Σ_μ as a bidendriform isomorphism from \mathbf{WQSym}^* to \mathbf{WQSym} . Now we can set $\mu : w \mapsto \hat{w}$ as a graded bijection. The extension σ_μ correspond to the color swap on red-packed forests (*i.e.* $\sigma_\mu : f \mapsto \hat{f}$). Finally we have the following theorem :

Théorème 6.26. *The linear map $\Sigma : \mathbf{WQSym}^* \rightarrow \mathbf{WQSym}$ defined as for all packed forest f ,*

$$\Sigma(\mathbb{P}_f) := \mathbb{O}_{\hat{f}}$$

is a bidendriform isomorphism between \mathbf{WQSym}^ and \mathbf{WQSym} .*

Démonstration. This theorem is a direct consequence of [corollaires 3.28 et 6.25](#). \square

Troisième partie

Autour de **WQSym**, résultats et perspectives

Généralisation de l'inversion

Quand on regarde **FQSym**, qui est l'algèbre de Hopf indexée par les permutations, l'inversion des permutations constitue un isomorphisme bidendriforme de cette algèbre ([équation \(3.26\)](#)). Ce qui était complexe avec **WQSym**, c'est que l'inversion des mots tassés n'était pas définie. Néamoins, notre construction a permis d'obtenir un isomorphisme bidendriforme. Dans celle-ci, la restriction aux permutations ne correspond pas à l'inversion. En particulier, pour les permutations irréductibles rouges et bleus, comme pour tous les mots tassés irréductibles rouges et bleus, nous avons choisi l'identité pour ne pas faire des permutations un cas particulier. Nous aurions pu choisir de prendre l'inversion pour ces permutations précises, notre construction aurait alors donné l'inversion pour toute les permutations et aurait permis de s'approcher d'une certaine compatibilité avec l'inversion des permutations.

La question qui se pose alors est la suivante : existe-t-il une généralisation de l'inversion des permutations pour les mots tassés ? Notre travail a l'intérêt de réduire le champ de recherche au sous-ensemble des mots tassés irréductibles rouges et bleus.

Généralisation de la théorie des arbres biplans

L'algèbre **PQSym** indexée par les **fonctions de parking** est très similaire à **WQSym**, mais aussi plus complexe et son étude serait un premier pas vers la généralisation de la théorie des arbres biplans. En effet **PQSym** est également une bigèbre bidendriforme et les fonctions de parking sont un sur-ensemble des mots tassés. La question de généraliser la théorie aux fonctions de parking relève à la fois de la combinatoire et de l'algèbre.

Nous avons eu l'occasion de travailler avec Aaron Lauve et Amy Pang, qui cherchaient s'il existait une base de **PQSym** dans laquelle le produit de mélange était sans chevau-chement. Nous avons au moins une piste de recherche prometteuse, mais non présentable en son état actuel. Les résultats de cette recherche pourraient faire l'objet d'un notebook similaire à ceux présentés dans la [partie IV](#). La construction de ce changement de base serait aussi le premier ingrédient nécessaire pour généraliser la théorie des arbres biplans sur **PQSym**.

Les pistes de recherche induites par ce travail sont les suivantes : comment généraliser la théorie des arbres biplans sur **PQSym** ? On pourra aussi s'intéresser aux **matrices tassées** et encore généraliser toutes les constructions de la [partie II](#) à toutes bigèbres bidendriformes. On cherchera alors quels sont les ingrédients nécessaires et suffisants pour développer la théorie des arbres biplans et obtenir un auto-morphisme bidendriforme.

Lien entre bigèbre bidendriforme et l'opérade dupliaque déformée

Nous avons travaillé sur **WQSym** en tant que bigèbre bidendriforme. Dans notre construction, \blacktriangleright et $/$ vérifient de façon inattendue des relations similaires à l'opérade du-

pliciale déformée [BDO20]. À symétrie près, les relations entre \blacktriangle et $/$ sont les mêmes. Ces relations font apparaître une nouvelle application de l'opérade duplique déformée sur des mots tassés.

Les questions issues de cette nouvelle application sont :
peut-on trouver une structure duplique déformée sur **WQSym** qui soit en lien avec la structure bidendriforme ?
Et, plus généralement, y a-t-il un lien entre bigèbre bidendriforme et duplique déformée ?

Lien entre bigèbre bidendriforme et *L*-algèbre

Nous avons introduit une nouvelle famille d'objets combinatoires : les **forêts d'arbres biplans**. Après une recherche sur OEIS [SI20], il a été constaté que le nombre d'arbres biplans à n noeuds est égal à la dimension de la composante homogène de la *L*-algèbre [Ler07] libre à un seul générateur.

Les questions induites par cette constatation sont :
est-il possible de mettre une structure de *L*-algèbre qui réalise concrètement la *L*-algèbre à un seul générateur sur les arbres biplans ?
Et plus généralement, y a-t-il un lien entre bigèbre bidendriforme et *L*-algèbre ?

Parmi ces quatre séries de questions, nous avons choisi de commencer par nous focaliser sur les opérades liées aux nouvelles opérations \blacktriangleright et \blacktriangle . Grâce à cette étude, nous déblayons le terrain pour les deux dernières séries de questions. En effet, nous étudions, pour l'ensemble des mots tassés, les algèbres sur l'opérade duplique déformée et la *L*-algèbre. Nous donnons également toutes les relations supplémentaires et les ensembles de générateurs.

Chapitre 7

Opérades et mots tassés

Dans la [partie II](#), nous avons définis plusieurs opérations sur les mots tassés. Ces opérations permettent de munir l'ensemble des mots tassés de structure d'algèbre sur différentes opérades. Nous résumons dans ce chapitre les résultats que nous avons avec le point de vu de l'étude des opérades.

7.1 Rappels sur les opérades

Une opérade est une structure algébrique qui constitue une abstraction d'un type d'algèbre. Nous renvoyons à [\[VL12, Mar08\]](#) pour les références classiques et à [\[Cha08, Gir18, Gir11, HP19\]](#) pour une approche plus combinatoire.

7.1.1 Définitions de bases

Définition 7.1. Une **opérade** (unitaire et non symétrique) est un espace vectoriel gradué $\mathcal{O} \bigoplus_{p \geq 1} \mathcal{O}(p)$ muni d'un élément distinctif $\mathbb{1} \in \mathcal{O}(1)$ (appelé unité de \mathcal{O}) et d'applications linéaires (appelées **compositions partielles** de \mathcal{O})

$$\circ_i : \mathcal{O}(p) \otimes \mathcal{O}(q) \rightarrow \mathcal{O}(p+q-1) \quad \text{pour } p, q \geq 1 \text{ et } i \in [1 \dots p]$$

qui vérifient les trois relations suivantes (**unité**, **composition en série**, **composition en parallèle**) :

$$\begin{aligned} \mathbb{1} \circ_1 \mathfrak{p} &= \mathfrak{p} = \mathfrak{p} \circ_i \mathbb{1} && \text{pour tout } \mathfrak{p} \in \mathcal{O}(p), i \in [1 \dots p], \\ (\mathfrak{p} \circ_i \mathfrak{q}) \circ_{i+j-1} \mathfrak{r} &= \mathfrak{p} \circ_i (\mathfrak{q} \circ_j \mathfrak{r}) && \text{pour tout } \mathfrak{p} \in \mathcal{O}(p), \mathfrak{q} \in \mathcal{O}(q), \mathfrak{r} \in \mathcal{O}(r), i \in [1 \dots p], j \in [1 \dots q], \\ (\mathfrak{p} \circ_i \mathfrak{q}) \circ_{j+q-1} \mathfrak{r} &= (\mathfrak{p} \circ_j \mathfrak{r}) \circ_i \mathfrak{q} && \text{pour tout } \mathfrak{p} \in \mathcal{O}(p), \mathfrak{q} \in \mathcal{O}(q), \mathfrak{r} \in \mathcal{O}(r), i < j \in [1 \dots p]. \end{aligned}$$

Les éléments de $\mathcal{O}(p)$ sont appelés **opérations** d'arité p .

Définition 7.2. La **série de Hilbert** d'une opérade \mathcal{O} est la série formelle $\mathcal{H}_{\mathcal{O}}(t)$ qui encode les dimensions de $\mathcal{O}(p)$ pour $p \geq 1$:

$$\mathcal{H}_{\mathcal{O}}(t) := \sum_{p \geq 1} \dim(\mathcal{O}(p)) t^p.$$

Définition 7.3. Un **arbre syntaxique** sur un ensemble gradué $\mathfrak{B} := \bigsqcup_{p \geq 1} \mathfrak{B}(p)$ est un arbre plan t où chaque noeud interne ayant p enfants est étiqueté par un élément de $\mathfrak{B}(p)$. On note $\text{Trees}(\mathfrak{B})(p)$ l'ensemble des arbres syntaxiques d'arité p et $\text{Trees}(\mathfrak{B}) := \bigsqcup_{p \geq 1} \text{Trees}(\mathfrak{B})(p)$.

Définition 7.4. L'**opérade libre** sur un ensemble gradué $\mathfrak{B} := \bigsqcup_{p \geq 1} \mathfrak{B}(p)$ est l'opérade

$$\mathbf{Free}(\mathfrak{B}) := \bigoplus_{p \geq 1} \mathbf{Free}(\mathfrak{B})(p),$$

où les trois conditions suivantes sont vérifiées :

- le $\mathbf{Free}(\mathfrak{B})(p)$ est l'espace vectoriel engendré par les arbres syntaxiques sur \mathfrak{B} d'arité p ,
- en numérotant les feuilles de la gauche vers la droite, la composition partielle \circ_i est l'application linéaire définie sur deux arbres syntaxiques s et t en greffant la racine de t sur la $i^{\text{ème}}$ feuille de s , et étendue par linéarité à $\mathbf{Free}(\mathfrak{B})$,
- l'unité est l'arbre qui a un seul noeud (et donc aucun noeud interne).

Comme son nom l'indique, une opérade libre est un objet libre dans la catégorie des opérades non symétriques de sorte que toute opérade peut être obtenue par le quotient d'une opérade libre.

Définition 7.5. Une **présentation** d'une opérade \mathcal{O} est la paire $(\mathfrak{B}, \mathfrak{R})$ où $\mathfrak{B} := \bigsqcup_{p \geq 1} \mathfrak{B}(p)$ est un ensemble gradué et \mathfrak{R} est un sous-espace de l'opérade libre $\mathbf{Free}(\mathfrak{B})$ de telle sorte que \mathcal{O} est isomorphe au quotient $\mathbf{Free}(\mathfrak{B})/\langle \mathfrak{R} \rangle$. Les éléments de \mathfrak{B} sont appelés les **générateurs** et les éléments de \mathfrak{R} sont appelés les **relations** de la présentation.

Définition 7.6. Une opérade est dite **binaire** si elle admet une présentation $(\mathfrak{B}, \mathfrak{R})$ dans laquelle les générateurs de \mathfrak{B} sont tous d'arité 2 (c'est-à-dire des arbres binaires). Une opérade binaire est dite **quadratique** si elle admet une présentation $(\mathfrak{B}, \mathfrak{R})$ dans laquelle les relations de \mathfrak{R} impliquent uniquement des arbres à 2 noeuds internes (c'est-à-dire des éléments d'opérades d'arité 3).

Dans cette section nous ne travaillerons qu'avec des opérades binaires quadratiques que nous décrirons uniquement par leur présentation (générateurs, relations).

7.1.2 Exemples d'opérades

Définition 7.7. L'opérade **dendriforme** Dend est l'opérade quadratique sur $\mathfrak{B} := \{\prec, \succ\}$ définie par les trois relations linéaires suivantes :

$$\begin{array}{ccc} \begin{array}{c} \prec \\ \diagdown \quad \diagup \\ \square \quad \square \\ \diagup \quad \diagdown \\ \prec \end{array} + \begin{array}{c} \prec \\ \diagdown \quad \diagup \\ \square \quad \succ \\ \diagup \quad \diagdown \\ \succ \end{array} & = & \begin{array}{c} \prec \\ \diagdown \quad \diagup \\ \square \quad \square \\ \diagup \quad \diagdown \\ \prec \end{array} \\ \begin{array}{c} \succ \\ \diagdown \quad \diagup \\ \square \quad \prec \\ \diagup \quad \diagdown \\ \prec \end{array} & = & \begin{array}{c} \succ \\ \diagdown \quad \diagup \\ \square \quad \prec \\ \diagup \quad \diagdown \\ \prec \end{array} \\ \begin{array}{c} \succ \\ \diagdown \quad \diagup \\ \square \quad \square \\ \diagup \quad \diagdown \\ \succ \end{array} & = & \begin{array}{c} \succ \\ \diagdown \quad \diagup \\ \square \quad \square \\ \diagup \quad \diagdown \\ \prec \end{array} + \begin{array}{c} \prec \\ \diagdown \quad \diagup \\ \square \quad \square \\ \diagup \quad \diagdown \\ \succ \end{array} \end{array} .$$

Définition 7.8. L’opérade **dupliciale** Dup est l’opérade quadratique sur $\mathfrak{B} := \{\prec, \succ\}$ définie par les trois relations suivantes :

$$\begin{array}{c} \text{Diagram 1: } \begin{array}{ccc} \begin{array}{c} \prec \\ \diagdown \quad \diagup \\ \square \quad \square \end{array} & = & \begin{array}{c} \prec \\ \diagup \quad \diagdown \\ \square \quad \square \end{array} \end{array} & \begin{array}{c} \text{Diagram 2: } \begin{array}{ccc} \begin{array}{c} \succ \\ \diagdown \quad \diagup \\ \square \quad \square \end{array} & = & \begin{array}{c} \succ \\ \diagup \quad \diagdown \\ \square \quad \square \end{array} \end{array} & \begin{array}{c} \text{Diagram 3: } \begin{array}{ccc} \begin{array}{c} \succ \\ \diagdown \quad \diagup \\ \square \quad \square \end{array} & = & \begin{array}{c} \succ \\ \diagup \quad \diagdown \\ \square \quad \square \end{array} \end{array} \end{array} \end{array}$$

La variante gauche de l’opérade suivante est nommée *skew duplicial* dans [BDO20] et les deux variantes sont appelées *left and right graft* dans [Man12].

Définition 7.9. L’opérade **dupliciale déformée gauche** Dup^\prec (resp. **droite** Dup^\succ) est l’opérade quadratique sur $\mathfrak{B} := \{\prec, \succ\}$ définie par les trois relations suivantes dans lesquelles les déformations sont symbolisées en rouge :

$$\begin{array}{c} \text{Diagram 1: } \begin{array}{ccc} \begin{array}{c} \prec \\ \diagdown \quad \diagup \\ \square \quad \square \end{array} & = & \begin{array}{c} \prec \\ \diagup \quad \diagdown \\ \square \quad \square \end{array} \end{array} & \begin{array}{c} \text{Diagram 2: } \begin{array}{ccc} \begin{array}{c} \succ \\ \diagdown \quad \diagup \\ \square \quad \square \end{array} & = & \begin{array}{c} \succ \\ \diagup \quad \diagdown \\ \square \quad \square \end{array} \end{array} & \begin{array}{c} \text{Diagram 3: } \begin{array}{ccc} \begin{array}{c} \succ \\ \diagdown \quad \diagup \\ \square \quad \square \end{array} & = & \begin{array}{c} \succ \\ \diagup \quad \diagdown \\ \square \quad \square \end{array} \end{array} \end{array} \end{array} \\ \text{resp. } \begin{array}{c} \begin{array}{c} \prec \\ \diagdown \quad \diagup \\ \square \quad \square \end{array} & = & \begin{array}{c} \prec \\ \diagup \quad \diagdown \\ \square \quad \square \end{array} \end{array} & \begin{array}{c} \begin{array}{c} \succ \\ \diagdown \quad \diagup \\ \square \quad \square \end{array} & = & \begin{array}{c} \succ \\ \diagup \quad \diagdown \\ \square \quad \square \end{array} \end{array} & \begin{array}{c} \begin{array}{c} \succ \\ \diagdown \quad \diagup \\ \square \quad \square \end{array} & = & \begin{array}{c} \succ \\ \diagup \quad \diagdown \\ \square \quad \square \end{array} \end{array} \end{array} \end{array}$$

Les quatres opérades Dend , Dup , Dup^\prec et Dup^\succ ont la même série de Hilbert, à savoir la série des nombres de Catalan \mathcal{G}_{Cat} :

n	1	2	3	4	5	6	7	8	9	OEIS [SI20]
$\mathcal{G}_{\text{Cat}_n}$	1	2	5	14	42	132	429	1430	4862	A000108

TABLE 7.1 – Nombres de Catalan inférieurs à 9.

Définition 7.10. L’opérade des *L-algèbres* \mathcal{L} est l’opérade quadratique sur $\mathfrak{B} := \{\prec, \succ\}$ définie par la relation suivante :

$$\begin{array}{c} \text{Diagram: } \begin{array}{c} \succ \\ \diagdown \quad \diagup \\ \square \quad \square \end{array} = \begin{array}{c} \succ \\ \diagup \quad \diagdown \\ \square \quad \square \end{array} . \end{array}$$

La série de Hilbert des *L-algèbre* est égale à la série génératrice des arbres biplans :

n	1	2	3	4	5	6	7	8	OEIS [SI20]
$\mathcal{G}_{\mathcal{L}_n}$	1	2	7	30	143	728	3 876	21 318	A006013

TABLE 7.2 – Nombre d’arbres biplans de taille inférieure à 8.

Définition 7.11. [Man12, Définition 28] L’opérade **bigrift** \mathbf{BG} est l’opérade quadratique sur $\mathfrak{B} := \{\star, \prec, \succ\}$ définie par les six relations suivantes :

$$\begin{array}{ccc} \text{Diagram 1: } \begin{array}{c} \star \\ \diagdown \quad \diagup \\ \square \quad \star \end{array} & = & \begin{array}{c} \star \\ \diagup \quad \diagdown \\ \star \quad \square \end{array} \\ \text{Diagram 2: } \begin{array}{c} \prec \\ \diagdown \quad \diagup \\ \square \quad \star \end{array} & = & \begin{array}{c} \star \\ \diagup \quad \diagdown \\ \prec \quad \square \end{array} \\ \text{Diagram 3: } \begin{array}{c} \succ \\ \diagdown \quad \diagup \\ \square \quad \star \end{array} & = & \begin{array}{c} \star \\ \diagup \quad \diagdown \\ \star \quad \square \end{array} \\ \text{Diagram 4: } \begin{array}{c} \prec \\ \diagdown \quad \diagup \\ \square \quad \prec \end{array} & = & \begin{array}{c} \prec \\ \diagup \quad \diagdown \\ \prec \quad \square \end{array} \\ \text{Diagram 5: } \begin{array}{c} \star \\ \diagdown \quad \diagup \\ \square \quad \prec \end{array} & = & \begin{array}{c} \prec \\ \diagup \quad \diagdown \\ \star \quad \square \end{array} \\ \text{Diagram 6: } \begin{array}{c} \succ \\ \diagdown \quad \diagup \\ \square \quad \prec \end{array} & = & \begin{array}{c} \prec \\ \diagup \quad \diagdown \\ \prec \quad \square \end{array} \end{array} .$$

Autrement dit, les opérateurs $\{\star, \prec, \succ\}$ de l’opérade bigraft vérifie les mêmes relations que dans les opérades dupliaires déformées gauche et droite et que dans l’opérade de L -algèbre.

Proposition 7.12. *Les opérades dupliaires déformées gauche et droite et l’opérade de L -algèbre sont des sous opérades de l’opérade bigraft.*

$$\mathbf{BG}(\star, \prec, \succ) \iff \mathbf{Dup}^\prec(\star, \succ) \text{ et } \mathbf{Dup}^\succ(\prec, \star) \text{ et } \mathcal{L}(\prec, \succ).$$

Cette proposition est prouvée dans la [section 7.4](#).

La série de Hilbert de l’opérade bigraft est égale à la série génératrice des forêts d’arbres biplans :

n	1	2	3	4	5	6	7	8	OEIS [SI20]
$\mathcal{G}_{\mathbf{BG}_n}$	1	3	12	55	273	1 428	7 752	43 263	A001764

TABLE 7.3 – Nombre de forêts biplanes de taille inférieure à 8.

7.1.3 Algèbre sur une opérade

Les résultats que nous présenterons dans les sections suivantes sont des expressions d’algèbres sur des opérades avec les opérations que nous avons définies dans la [partie II](#). Nous donnons donc ici les définitions utiles à la présentation de ces résultats.

Définition 7.13. On appelle **algèbre sur une opérade** \mathcal{O} (ou \mathcal{O} **algèbre**) un espace vectoriel \mathbf{Alg} tel que les éléments de l’opérade \mathcal{O} sont en correspondances avec des opérations linéaires sur \mathbf{Alg} et que les compositions partielles sur \mathcal{O} correspondent aux compositions partielles dans \mathbf{Alg} .

Plus formellement \mathbf{Alg} est munie d’une action $\cdot : \mathcal{O}(p) \otimes \mathbf{Alg}^{\otimes p} \rightarrow \mathbf{Alg}$ qui satisfait les relations imposées par les compositions dans \mathcal{O} .

Pour tout $\mathfrak{p} \in \mathcal{O}(p)$, $\mathfrak{q} \in \mathcal{O}(q)$, $a_1 \otimes \cdots \otimes a_{p+q-1} \in \mathbf{Alg}^{\otimes p+q-1}$ et $i \in [1, \dots, p]$:

$$(\mathfrak{p} \circ_i \mathfrak{q}) \cdot (a_1 \otimes \cdots \otimes a_{p+q-1}) = \mathfrak{p} \cdot (a_1 \otimes \cdots \otimes a_{i-1} \otimes \mathfrak{q} \cdot (a_i \otimes \cdots \otimes a_{i+q-1}) \otimes a_{i+q} \otimes \cdots \otimes a_{p+q-1}).$$

Exemple 7.14. Comme nous l'avons déjà dit plus tôt, l'algèbre **WQSym** munie des opérations \prec, \succ est une algèbre dendriforme.

La notion d'algèbre sur une opérade admet toujours un objet libre. Nous avons expliqué comment ci-dessus avant d'en décrire une version gradué ci-dessous.

Définition 7.15. Soit E un ensemble, \mathcal{O} une opérade et $(\mathcal{B}_n)_n$ une base de \mathcal{O} , avec donc $\mathcal{B}_n = (\circ_1^n, \circ_2^n, \dots, \circ_{\dim(\mathcal{O}(n))}^n)$ pour tout n . Nous construisons la **\mathcal{O} algèbre libre sur E** de la façon suivante :

$$\langle E \rangle_{\mathcal{O}} := \bigoplus_{n>0} \mathcal{B}_n \times E^n. \quad (7.1)$$

Définition 7.16. Il est possible d'ajouter une **graduation** à la \mathcal{O} algèbre libre sur un ensemble E si celui-ci est gradué. En effet pour toute base \mathcal{B}_n et tous e_1, \dots, e_n éléments de E nous posons :

$$\deg(\mathcal{B}_n \times (e_1, \dots, e_n)) := \sum_{i=1}^n \deg(e_i). \quad (7.2)$$

Le degré d'un élément de l'algèbre est égal à la somme des degrés des éléments de l'ensemble qui le compose. En particulier, comme cette graduation est compatible avec la structure d'opérade et que nous étudions uniquement des opérades binaires quadratiques, soit a et b deux éléments de E et \square un générateur de l'opérade \mathcal{O} , nous avons :

$$\deg(a \square b) = \deg(a) + \deg(b). \quad (7.3)$$

Proposition 7.17. Soit \mathcal{C} une classe combinatoire telle que $\mathcal{C}_0 = \emptyset$ avec $\mathcal{G}_{\mathcal{C}}$ sa série génératrice et \mathcal{O} une opérade quadratique avec $\mathcal{H}_{\mathcal{O}}$ sa série de Hilbert. Nous avons la série de Hilbert de la \mathcal{O} algèbre libre sur \mathcal{C} :

$$\mathcal{H}_{\langle \mathcal{C} \rangle_{\mathcal{O}}} = \mathcal{H}_{\mathcal{O}} \circ \mathcal{G}_{\mathcal{C}}. \quad (7.4)$$

Démonstration. Immédiate avec la [définition 7.16](#) de la graduation d'une \mathcal{O} algèbre et le fait que les relations d'une opérade soient homogènes. \square

Cette proposition sera très utile dans les sections suivantes pour prouver que les ensembles sont générés librement par un sous-ensemble.

7.2 Une algèbre duplicitale déformée sur les mots tassés

WQSym, qui est l'objet de cette thèse, est une algèbre dendriforme. Au cours de cette étude, nous avons été conduit à créer deux nouvelles opérations sur les mots \blacktriangleright et \blacktriangleleft ([définitions 4.5 et 4.44](#)). Pour cette section nous définissons l'opération \blacktriangledown comme le symétrique de \blacktriangleleft , c'est-à-dire : $v \blacktriangledown u = u \blacktriangleleft v$.

Nous avons pu constater dans les [remarques 4.13 et 4.53](#) que ces opérations vérifient les relations des opérades dupliaires déformées. En effet, pour tous mots tassés non vides u , v et w :

$$u/(v/w) = (u/v)/w, \quad (7.5)$$

$$u\blacktriangleright(v\blacktriangleright w) = (u/v)\blacktriangleright w, \quad (\text{voir lemme 4.11}), \quad (7.6)$$

$$u\blacktriangleright(v/w) = (u\blacktriangleright v)/w, \quad (\text{voir lemme 4.12}), \quad (7.7)$$

$$(u/v)/w = u/(v/w), \quad (7.8)$$

$$(u\blacktriangledown v)\blacktriangledown w = u\blacktriangledown(v/w), \quad (\text{voir lemme 4.51}), \quad (7.9)$$

$$(u/v)\blacktriangledown w = u/(v\blacktriangledown w), \quad (\text{voir lemme 4.52}). \quad (7.10)$$

Proposition 7.18. *L'ensemble des mots tassés muni des opérations / et \blacktriangleright (resp. / et \blacktriangledown) forme une algèbre dupliaire déformée droite (resp. gauche) et cette algèbre est librement engendrée par les mots rouges-irréductibles (resp. bleus-irréductibles).*

Démonstration. Nous prouverons ce résultat pour les opérations / et \blacktriangleright puis nous donnerons les résultats utiles pour prouver le résultat sur les opérations / et \blacktriangledown .

Pour cette preuve, nous allons procéder en trois temps, d'abord vérifier les relations, nous montrerons ensuite que l'ensemble des mots rouges-irréductibles est générateurs, puis nous montrerons la liberté par un argument combinatoire de dimension dans un dernier temps.

Avec les [équations \(7.5\), \(7.6\) et \(7.7\)](#) et les correspondances \prec avec \blacktriangleright et \succ avec / nous avons bien les relations de dupliaire déformée droite qui sont vérifiées.

D'après le [lemme 1.45](#) tout mot tassé se décompose en mots tassés irréductibles. De plus, d'après le [lemme 4.7](#) et la [proposition 4.15](#), tout mot tassé irréductible se décompose en $u\blacktriangleright v$ avec v rouge-irréductible et u un mot tassé. Nous avons déjà utilisé ces deux résultats et nous avons construit des arbres rouges-squelettes ([définition 4.25](#)) qui symbolisent cette décomposition récursive.

Les mots tassés sont donc générés par l'ensemble des mots tassés rouges-irréductibles avec les opérations / et \blacktriangleright en tant qu'algèbre dupliaire déformée droite.

Pour montrer la liberté, nous allons utiliser la [proposition 7.17](#) afin de justifier que tous les mots tassés sont décomposables de façon unique. Cette fois aussi les arbres rouges-squelettes nous donnent une intuition car ce sont des objets de Catalan dont les nœuds sont étiquetés par des mots rouges-irréductibles. D'après le [théorème 4.40](#) les cardinalités de l'ensemble des mots tassés rouges-irréductibles et des mots tassés concordent avec la relation suivante :

$$\mathcal{G}_{\mathbf{PW}} = \mathcal{G}_{\mathbf{Cat}} \circ \mathcal{G}_{Red-Irr},$$

où $\mathcal{G}_{\mathbf{PW}}$ est la série génératrice des mots tassés, $\mathcal{G}_{\mathbf{Cat}}$ est la série des nombres de Catalan et $\mathcal{G}_{Red-Irr}$ est la série des mots tassés rouges-irréductibles. Ainsi, les dimensions de l'ensemble

des mots tassés muni des opérations / et \blacktriangleright coïncident avec celles de l'algèbre libre, ce qui montre qu'il n'y a pas d'autre relation. Cette algèbre est donc libre.

De même, avec les correspondances \prec avec / et \succ avec \blacktriangledown nous avons bien les relations de duplike déformée gauche qui sont vérifiées. Ensuite, d'une part d'après le [lemme 4.46](#) et la [proposition 4.55](#), les mots tassés sont générés par l'ensemble des mots tassés bleus-irréductibles avec les opérations / et \blacktriangledown . D'autre part d'après le [théorème 4.76](#) et la [proposition 7.17](#) cette génération est libre. \square

Remarque 7.19. L'ensemble engendré par 1 (le mot tassé de longueur 1) muni des opérations / et \blacktriangleright (resp. \blacktriangledown) est l'ensemble de toutes les permutations évitant le motif 132.

En effet, tous les mots générés sont des permutations car aucune de ces opérations ne permet d'ajouter une égalité de valeurs. De plus, si σ et μ sont deux permutations qui évitent le motif 132, alors σ/μ , $\sigma\blacktriangleright\mu$ et $\sigma\blacktriangledown\mu$ évitent également le motif 132. Enfin, d'après la proposition précédente, les dimensions de cet espace suivent la série de Catalan.

7.3 Une L -algèbre sur les mots tassés

Par ailleurs, d'après le [lemme 6.6](#), les opérations \blacktriangleright et \blacktriangledown vérifient une relation de L -algèbre. En effet, pour tous mots tassés u , v et w avec v non vide :

$$u\blacktriangleright(v\blacktriangledown w) = (u\blacktriangleright v)\blacktriangledown w \quad (7.11)$$

Proposition 7.20. *L'ensemble des mots tassés irréductibles muni des opérations \blacktriangleright et \blacktriangledown est une L -algèbre. Cette algèbre est engendrée par les mots tassés à la fois rouges-irréductibles et bleus-irréductibles, avec une seule relation : $u\blacktriangleright 1 = 1\blacktriangledown u$.*

Démonstration. Nous prouverons ce résultat de la même façon que pour la [proposition 7.18](#), c'est-à-dire que nous allons procéder en trois temps. D'abord nous vérifierons les relations, nous montrerons ensuite que l'ensemble des mots à la fois rouges-irréductibles et bleus-irréductibles est générateurs, puis nous montrerons qu'il n'y a qu'une seule autre relation.

Avec l'[équation \(7.11\)](#) et les correspondances \prec avec \blacktriangledown et \succ avec \blacktriangleright nous avons bien la relation des L -algèbres qui est vérifiée.

D'après la [définition 6.8](#) et le [lemme 6.9](#), tout mot tassé irréductible se décompose en $u\blacktriangleright v\blacktriangledown w$ (les parenthèses sont inutiles au vu de la relation) avec v à la fois rouge-irréductible et bleu-irréductible.

Pour cette preuve nous avons besoin de réintroduire l'opération /. D'après le [lemme 1.45](#) u (resp. w) se décompose en irréductibles $u_{(1)}/\dots/u_{(k)}$ (resp. $w_{(1)}/\dots/w_{(\ell)}$). Nous pouvons ensuite utiliser la relation de l'[équation \(7.6\)](#) (resp. [équation \(7.9\)](#)) pour dire que tout mot tassé irréductible se décompose en

$$((u_{(1)}\blacktriangleright(\dots\blacktriangleright(u_{(k)}\blacktriangleright v))\blacktriangledown w_{(1)})\blacktriangledown\dots)\blacktriangledown w_{(\ell)}$$

avec v à la fois rouge-irréductible et bleu-irréductible et les $u_{(i)}$ et $w_{(j)}$ sont irréductibles pour tout i et tout j . Nous pouvons reconnaître ici la construction d'un arbre bicolore associé à un mot tassé irréductible (voir [définition 6.11](#)).

Les mots tassés irréductibles sont donc générés par les mots à la fois rouges-irréductibles et bleus-irréductibles avec les opérations \blacktriangleright et \blacktriangledown en tant que L -algèbre.

Pour montrer qu'il n'y a que $u\blacktriangleright 1 = 1\blacktriangledown u$ comme autre relation nous allons de nouveau utiliser la [proposition 7.17](#) après avoir supprimé les doublons grâce au [théorème 6.16](#).

En effet, nous avons défini les arbres rouge-bleu-tassés ([définition 6.13](#)) comme les arbres bicolores étiquetés par des mots à la fois rouges-irréductibles et bleus-irréductibles mais avec la contrainte que si le mot en étiquette est le mot 1 alors la forêt droite est vide. Nous avons ensuite montré dans le [théorème 6.16](#) que ce sous-ensemble d'arbres était en bijection avec les mots tassés irréductibles.

Plus précisément, d'après le [théorème 6.16](#) les cardinalités de l'ensemble des mots tassés irréductibles et des mots tassés à la fois rouges-irréductibles et bleus-irréductibles concordent avec la relation suivante :

$$\mathcal{G}_{Irr} = \mathcal{G}_{T_{RB}} \circ \mathcal{G}_{Red-Irr \cap Blue-Irr},$$

où \mathcal{G}_{Irr} est la série génératrice des mots tassés irréductibles, $\mathcal{G}_{T_{RB}}$ est la série génératrice des arbres bicolores avec la contrainte sur le mot 1 en étiquette et $\mathcal{G}_{Red-Irr \cap Blue-Irr}$ est la série des mots tassés à la fois rouges-irréductibles et bleus-irréductibles. \square

7.4 Une algèbre Bigraft sur les mots tassés

Enfin il est possible de s'intéresser à l'opérade bigraft sous-jacente.

Proposition 7.21. *L'ensemble des mots tassés muni des opérations $/$, \blacktriangleright et \blacktriangledown forme une algèbre bigraft. Cette algèbre est engendrée par les mots tassés à la fois rouges-irréductibles et bleus-irréductibles, avec une seule relation : $u\blacktriangleright 1 = 1\blacktriangledown u$.*

Démonstration. Le fait que les mots tassés soient engendrés par les mots tassés à la fois rouges-irréductibles et bleus-irréductibles en tant qu'algèbre bigraft est une conséquence direct des [propositions 7.18 et 7.20](#). Enfin, d'après le [théorème 6.16](#) comme il y a une bijection entre les forêts d'arbres rouges-bleus-tassés et les mots tassés, la seule relation est $u\blacktriangleright 1 = 1\blacktriangledown u$. \square

Preuve de la proposition 7.12. On observe immédiatement qu'il existe des morphismes d'opérades depuis les opérades dupliaires déformées gauche et droite et l'opérade de L -algèbre vers l'opérade bigraft. Cela signifie que les algèbres bigrafts sont des algèbres dupliaires déformées gauche et droite et des L -algèbres.

Pour prouver l'injectivité de ces morphismes, observons une algèbre bigraft sur les mots tassés en prenant un unique générateur. Pour éviter la relation $u\blacktriangleright 1 = 1\blacktriangledown u$, prenons le mot tassé 11 comme générateur.

L'algèbre bigraft engendrée par le mot tassé 11 est libre, or ce qui est généré par 11 comme algèbre dupliaire déformée gauche ou droite ou comme L -algèbre est aussi libre d'après les [propositions 7.18 et 7.20](#). Il n'y a pas d'autres relations, le morphisme d'opérade est donc injectif. \square

Quatrième partie

Expérimentation avec SageMath

Présentons brièvement **SageMath** [Sag20] qui est un logiciel libre et gratuit de mathématiques sous licence *GPL*, créé en 2005. Chaque utilisateur peut devenir développeur et proposer des modifications, corrections ou ajouts au logiciel. Ces propositions sont envoyées sous forme de *tickets* et validées par d'autres développeurs avant d'être incorporées à une nouvelle version du logiciel. Nous avons eu besoin d'implanter un ticket SageMath (*ticket #25916 implement Packed Words*) sur les mots tassés, qui sont les objets combinatoires que nous utilisons le plus. Nous avons travaillé sur ce code en collaboration avec Aaron Lauve de Loyola University Chicago. Ces 2000 lignes de ce *ticket* devraient être intégrées à SageMath, mais quelques modifications mineures et relectures sont encore nécessaires avant que ce ticket soit validé.

Notre méthodologie de recherche a été d'implanter tous les objets combinatoires que nous avons définis sous forme d'objets SageMath, ce qui nous a permis de faire des tests sur d'autres propriétés et de répondre plus facilement à des questions annexes auxquelles nous n'avions pas initialement pensé. Nous avons également implanté une multitude de tests unitaires et une fois que tous ces tests ont été validés, nous avions une compréhension forte des interactions entre les objets pour nous aider à prouver nos résultats.

Actuellement quelques bases de **WQSym** sont déjà présentes dans SageMath (entre autres \mathbb{M} équation (3.27) et \mathbb{Q} équation (3.30)). L'algèbre **WQSym** est implantée en utilisant les partitions d'ensembles ordonnées comme base. En effet, nous avons vu plus haut (section 1.1.3) que ces objets sont en bijection avec les mots tassés. Pour pouvoir utiliser ce qui était déjà présent dans Sage, nous avons souvent dû faire appel à cette bijection pour les nouvelles bases que nous avons implantées. En effet, nous avons implanté plusieurs nouvelles bases (entre autres \mathbb{R} section 3.2.2, \mathbb{O} définition 5.13 et \mathbb{P} définition 5.5) et en attente de la création d'un ticket Sage, nous avons mis à disposition des notebooks pour que ce code soit visible par les autres chercheurs, comme nous allons le voir plus loin.

Nous avons écrit en tout environ 8000 lignes de code qui concernent l'algèbre **WQSym** et les objets combinatoires que nous utilisons.

Parmi l'ensemble de tout ce code, environ 3000 lignes constituent **nos propres outils** pour nous aider à trier nombre de nos conjectures. Ainsi, face à une question combinatoire et/ou algébrique, si la réponse n'est pas instantanée (ce qui est souvent le cas), notre premier réflexe a été de coder les objets combinatoires pour avoir une vision plus large. En effet, faire des calculs sur papier prend beaucoup trop de temps étant donné l'explosion exponentielle du nombre d'objets combinatoires. Par exemple, pour avoir une vision d'ensemble sur les 541 mots tassés de taille 5 et les arbres biplans associés, plutôt que de dessiner à la main, nous avons implanté une fonction qui nous a permis de les imprimer. En plus du gain de temps, les erreurs sont plus facilement évitées.

Chapitre 8

Notebooks en ligne

Dans ce chapitre, nous allons présenter 5 **notebooks mis en ligne en juin 2022** sur GitHub [Mlo22b] dans le but de rendre public notre travail expérimental. Ces notebooks, dont la version en ligne est en constante évolution et amélioration, sont ici figés à un temps t . Nous n'en donnons ici qu'un aperçu afin d'éviter certaines redondances, c'est pourquoi la numérotation n'est pas continue.

Les 3 premiers notebooks permettent, entre autres, de vérifier nos résultats de la [partie II](#). Ceci étant un de leurs aspects positifs, un autre étant que ce code est à la disposition des chercheurs en combinatoire algébrique qui peuvent l'utiliser pour leurs propres recherches similaires.

Les 2 derniers notebooks témoignent de notre méthodologie de recherche. Nous y trouverons des conjectures que nous avons pu écarter. Faisant face à de nombreuses classes combinatoires différentes, nous avons énuméré ces objets de plusieurs façons différentes ce qui nous a permis d'en avoir une vision plus générale et une compréhension plus fine. Nous avons dans le dernier notebook un exemple de ces calculs.

Plusieurs fonctions et méthodes du code mis en ligne sont commentées avec précision, le faire pour toutes serait un premier pas pour rendre ce code plus utilisable. De plus, toutes les fonctions ne sont pas forcément optimisées. Par exemple, certains changements de bases ne sont codés que dans un sens, pour obtenir l'inverse la machine va donc être conduite à faire de nombreux calculs inutiles (calcul et inversion de la matrice complète à un degré donné) qui vont impliquer un temps d'exécution exagérément long.

8.1 Objets combinatoires

8.1.1 Mots tassés

Nous exposons ici les fonctions sur les mots tassés qui sont utiles pour l'automorphisme bidendrifforme de WQSym.

```
In [1]: from packed_words import *
```

```
In [2]: PackedWord([]) # le mot vide
Out[2]: []

In [3]: PackedWord([1]) # le mot tassé de taille 1
Out[3]: [1]

In [4]: PackedWord([1,1,2]) # un mot tassé de taille 3
Out[4]: [1, 1, 2]

In [5]: p = PackedWord([3,1,2,1,2])
        print("Le mot tassé p =", p, "est de taille", p.size(), \
              "et sa valeur maximale est", p.max(), ".")
```

Le mot tassé `p = [3, 1, 2, 1, 2]` est de taille 5 et sa valeur maximale est 3 .

Les mots tassés sont ordonnés comme les partitions ordonnées, selon l'ordre lexicographique de leurs compositions (nombre de 1, nombre de 2, ... voir [définition 1.37](#)).

```
In [7]: for p in PackedWords(3):
        print(p.to_composition(), "\t", p)

[1, 1, 1]      [1, 2, 3]
[1, 1, 1]      [1, 3, 2]
[1, 1, 1]      [2, 1, 3]
[1, 1, 1]      [2, 3, 1]
[1, 1, 1]      [3, 1, 2]
[1, 1, 1]      [3, 2, 1]
[1, 2]         [1, 2, 2]
[1, 2]         [2, 1, 2]
[1, 2]         [2, 2, 1]
[2, 1]         [1, 1, 2]
[2, 1]         [1, 2, 1]
[2, 1]         [2, 1, 1]
[3]            [1, 1, 1]
```

Nous avons dû implanter les calculs des descentes globales et de la factorisation en descentes globales car ces fonctions sont très utiles dans tous nos travaux.

```
In [10]: pw = PackedWord([7, 5, 4, 6, 3, 1, 2, 1]);
          pw.global_descents()

Out[10]: [1, 4, 5]
```

```
In [11]: pw.global_descents_factorization()
```

```
Out[11]: [[1], [2, 1, 3], [1], [1, 2, 1]]
```

Il est possible de tester si un mot tassé est irréductible, rouge-irréductible, bleu-irréductible ou encore à la fois rouge-irréductible et bleu-irréductible.

```
In [13]: pw = PackedWord([1,4,3,2,1])
```

```
In [14]: pw.is_irreducible()
```

```
Out[14]: True
```

```
In [15]: pw.is_red_irreducible()
```

```
Out[15]: True
```

```
In [16]: pw.is_irreducible("red") # équivalent à pw.is_red_irreducible()
```

```
Out[16]: True
```

```
In [17]: pw.is_blue_irreducible() # ou pw.is_irreducible("blue")
```

```
Out[17]: True
```

```
In [18]: p.is_bicolored_irreducible() # ou pw.is_irreducible("bicolored")
```

```
Out[18]: True
```

On obtient alors facilement la liste des mots tassés rouges-irréductibles (resp. bleus-irréductibles) (resp. doubles-irréductibles).

```
In [19]: for p in PackedWords(3):
    if p.is_irreducible("red"):
        print(p)
```

```
[1, 3, 2]
[2, 1, 2]
[1, 2, 1]
[1, 1, 1]
```

8.1.2 Arbres biplans

Nous utilisons les `LabelledOrderedTree` natifs pour les arbres biplans. Les étiquettes sont composés d'une paire. Le premier élément de cette paire est un entier qui correspond au nombre d'enfants gauches. Le deuxième élément correspond aux étiquettes que nous avons vu jusqu'à présent dans les nœuds des arbres.

Pour afficher les arbres biplans, nous avons la possibilité de les afficher avec du code `ascii`, mais nous avons surtout notre propre fonction pour obtenir le code L^AT_EX de ces arbres.

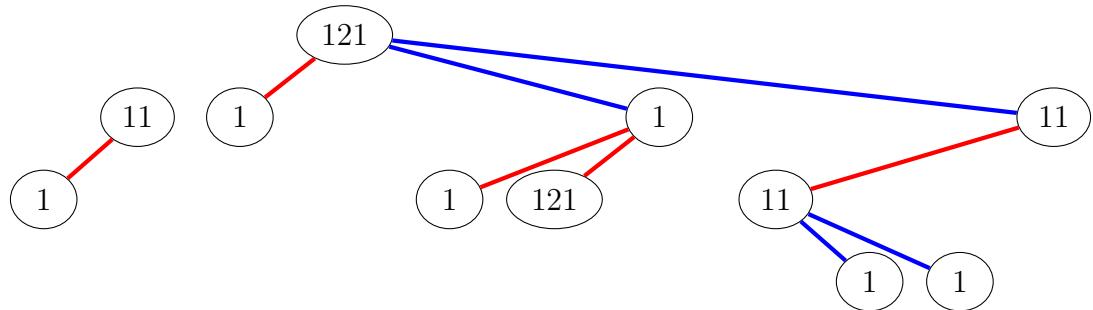
```
In [22]: from tree_latex import to_biplan_latex
```

```
In [36]: pw = PackedWord([12,13,13,10,9,11,6,7,6,5,8,3,2,1,3,4,4,9])
```

```
In [37]: ascii_art(pw.to_biplan(bicolored = ["red", "blue"]))
```

```
Out[37]: [ (1, [1, 1]), ----- (1, [1, 2, 1]) ----- ]
           [ |           /           /           / ]
           [ (0, [1])   (0, [1])   (2, [1])   (1, [1, 1]) ]
           [           /           /           |           ]
           [           (0, [1]) (0, [1, 2, 1]) (0, [1, 1]) ]
           [           /           /           (0, [1]) (0, [1]) ]
```

```
In [38]: view(to_biplan_latex(pw, bicolored = ["red", "blue"]))
```



8.2 Ordres partiels

Nous parlerons ici uniquement des généralisations des ordres faibles droit et gauche ([définitions 2.19 et 2.24](#)) sur les mots tassés. Ces deux ordres partiels décrivent les produits des bases \mathbb{Q} et \mathbb{R} .

L'ordre de Bergeron et Zabroki est déjà défini dans Sage sur les partitions ordonnées sous la nomination de `strongly_finer` et `strongly_fatter`.

Pour définir un ordre partiel en SageMath, il suffit de créer une fonction qui calcule les successeurs. Toutes les autres fonctions se déduisent automatiquement.

```
In [1]: from packed_words import *
```

Pour les généralisations des ordres faibles droit et gauche, toutes les fonctions classiques sont utilisables :

- `left_weak_order_succ`, `left_weak_order_pred`, `left_weak_order_smaller`,
`left_weak_order_greater`, `left_weak_order_interval`.
- `right_weak_order_succ`, `right_weak_order_pred`, `right_weak_order_smaller`,
`right_weak_order_greater`, `right_weak_order_interval`.

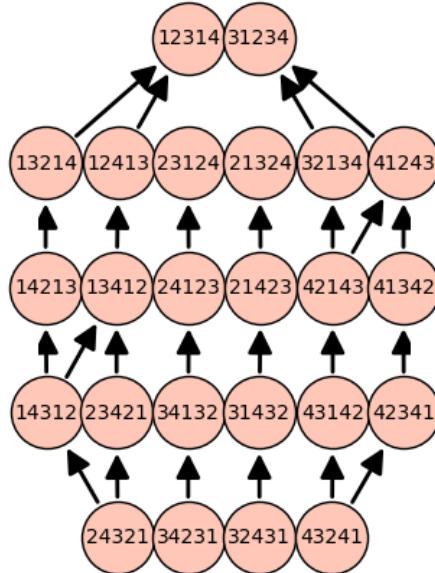
Il y a également deux fonctions d'affichage : `LWO_multi_plot` et `RWO_multi_plot` qui peuvent être améliorées, mais qui donnent une première vision des ordres partiels. Il est par exemple possible de retrouver les exemples des figures 2.5, 2.6, 2.8 et 2.9.

```
In [2]: PackedWord([1,2,1,3,4,2]).right_weak_order_succ()
```

```
Out[2]: [[2, 1, 1, 3, 4, 2], [1, 2, 3, 1, 4, 2], [1, 2, 1, 4, 3, 2]]
```

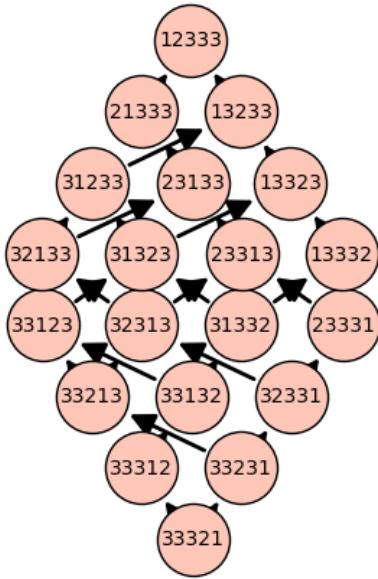
```
In [5]: PackedWords(5).LWO_multi_plot()[8]  
# on voit par exemple ici 4 composantes connexes
```

Out[5] :



```
In [6]: PackedWords(5).RWO_multi_plot()[3]  
# on voit par exemple ici 1 composante connexe
```

Out[6] :



8.3 Bases de WQSym

```
In [1]: from wqsym import *
from packed_words import *
WQSym = WordQuasiSymmetricFunctions(QQ);
WQSym.inject_shorthands();
# Cette ligne permet de pouvoir utiliser
# toutes les bases de WQSym implantées
M.options.objects = 'words';
M.options.display = 'compact'

Defining M as shorthand for Word Quasi-symmetric functions over
    Rational Field in the Monomial basis
Defining X as shorthand [...] in the Characteristic basis
Defining C as shorthand [...] in the Cone basis
Defining Q as shorthand [...] in the Q basis
Defining Phi as shorthand [...] in the Phi basis
Defining O as shorthand [...] in the O basis
Defining P as shorthand [...] in the P basis
Defining R as shorthand [...] in the RightWeakOrder basis
Defining GL as shorthand [...] in the GreaterLeft basis
Defining SR as shorthand [...] in the SmallerRight basis
```

Nous ne parlerons pas ici des bases \mathbb{X} , \mathbb{C} et Φ qui sont déjà implantées dans Sage et dont nous ne nous servons pas dans notre travail de recherche. Les bases \mathbb{M} et \mathbb{Q} sont également déjà implantées dans Sage, néanmoins nous avons ajouté la structure bidendriforme en

implantant les demi-produits et demi-coproduits. Nous les présentons entièrement dans ce notebook.

La base \mathbb{M} est la base monomiale ([équation \(3.27\)](#)) dont nous rappelons la définition :

$$\mathbb{M}_u := \sum_{\substack{w \in \mathcal{A}^*, \\ \text{pack}(w)=u}} w.$$

Les bases \mathbb{Q} , \mathbb{R} , \mathbb{O} et \mathbb{P} sont les bases définies plus haut ([équation \(3.30\)](#), [section 3.2.2](#), [et définitions 5.5 et 5.13](#)).

Opérations sur les bases de WQSym

Produits, demi-produits, coproduits et demi-coproduits sont implantés pour les bases \mathbb{M} , \mathbb{Q} et \mathbb{R} . Nous montrons ici quelques exemples de ces opérations dans la base \mathbb{M} .

In [3]: `M[1,2,1] << M[2,1]`

Out [3]: `M[14132] + M[13121] + M[24231] + M[23221] + M[34321]`

In [5]: `M[1,2,1] * M[2,1] == (M[1,2,1] << M[2,1]) + (M[1,2,1] >> M[2,1])`

Out [5]: `True`

In [7]: `M.left_coproduct(M[2,4,3,1,1,2,4,2])`

Out [7]: `M[21122] # M[212] + M[231122] # M[11]`

In [8]: `M.right_coproduct(M[2,4,3,1,1,2,4,2])`

Out [8]: `M[11] # M[132131]`

In [9]: `p = PackedWord([2,4,3,1,1,2,4,2]);
M.coproduct(M[p]) - (M.left_coproduct(M[p]) + M.right_coproduct(M[p]))`

Out [9]: `M[] # M[24311242] + M[24311242] # M[]`

De même pour les bases \mathbb{Q} et \mathbb{R} , produits, demi-produits, coproduits et demi-coproduits sont implantés. Nous avons fondé notre code sur les bases \mathbb{M} et \mathbb{Q} déjà existantes.

Nous aborderons plus tard les produits et coproduits dans les bases \mathbb{O} et \mathbb{P} .

Changements de bases

Certains changements de bases ne sont codés que dans un sens, pour obtenir l'inverse la machine va donc être conduite à faire de nombreux calculs inutiles qui vont impliquer un temps d'exécution exagérément long. Voici donc les sens qui sont implantés et qui sont donc rapides :

- $\mathbb{M} \leftrightarrow \mathbb{Q}$
- $\mathbb{O} \rightarrow \mathbb{Q}$
- $\mathbb{P} \rightarrow \mathbb{R}$

En effet, pour le calcul du changement de base $\mathbb{O} \rightarrow \mathbb{Q}$ (resp. $\mathbb{P} \rightarrow \mathbb{R}$), nous nous basons sur les définitions des forêts bleues (resp. rouges). Pour l'autre sens nous calculons la matrice complète que nous inversons. Il est sûrement possible d'améliorer cette partie.

Enfin l'isomorphisme entre \mathbb{O} et \mathbb{P} , qui sont des bases de **WQSym** et **WQSym***, est implanté comme un simple changement de base.

- $\mathbb{O} \leftrightarrow \mathbb{P}$

Nous présentons ici uniquement les deux premiers changements de base, pour plus de tests et d'exemples, nous invitons le lecteur à lire le notebook en question.

Changement de base entre \mathbb{M} et \mathbb{Q}

Rappel : Le changement de base entre \mathbb{M} et \mathbb{Q} est fondé sur l'ordre de raffinement \leq^* ([définition 2.27 et équation \(3.30\)](#)). Dans Sage, cet ordre est appelé **strongly_fatter**.

$$\mathbb{Q}_P = \sum_{P \leq^* W} \mathbb{M}_W,$$

In [13]: `p = PackedWord([7,6,2,4,3,4,5,1,1])`

In [14]: `M(Q[p])`

Out[14]: `M[762434511] + M[652434411] + M[652323411] + M[542323311]`

In [15]: `Q(M[p])`

Out[15]: `Q[762434511] - Q[652434411] - Q[652323411] + Q[542323311]`

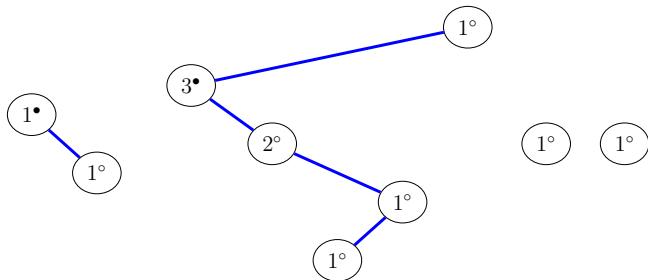
```
In [16]: matr_chgmt_base_osp(Q,M,3)
```

```
Out[16]: [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 1 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 1 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 1 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0]
[1 0 0 0 0 0 1 0 0 0 0 0 0 0 0]
[0 0 1 0 0 0 0 1 0 0 0 0 0 0 0]
[0 0 0 1 0 0 0 0 0 1 0 0 0 0 0]
[1 0 0 0 0 0 0 0 0 0 1 0 0 0 0]
[0 1 0 0 0 0 0 0 0 0 0 1 0 0 0]
[0 0 0 0 1 0 0 0 0 0 0 0 1 0 0]
[1 0 0 0 0 0 0 0 0 0 1 0 0 0 1]
```

Changement de base entre \mathbb{O} et \mathbb{Q}

```
In [18]: p = PackedWord([7,6,2,4,3,4,5,1,1])
```

```
In [19]: view(to_biplan_latex(p, "blue"))
```



Grâce à la structure de la forêt et la [définition 5.13](#) de la base \mathbb{O} , et plus précisément l'[équation \(5.14\)](#), nous déduisons cette première étape de calcul. En effet, nous avons immédiatement que $\mathbb{O}_1 = \mathbb{Q}_1$ et $\mathbb{O}_{11} = \mathbb{Q}_{11}$, il ne manque plus que le calcul de \mathbb{O}_{13234} dans la base \mathbb{Q} que nous voyons juste après.

```
In [20]: Q[1]<<(Q[1]<<(Q(O[1,3,2,3,4])<<Q[1,1])) == Q(O[p])
```

```
Out[20]: True
```

Pour le calcul de \mathbb{O}_{13234} dans la base \mathbb{Q} , nous effectuons la somme de l'algèbre de Brace (voir [équation \(5.15\)](#) et [théorème 3.27](#)).

```
In [21]: (Q(O[1,3,2,3]) >> Q[1]) - (Q[1] << Q(O[1,3,2,3])) == Q(O[1,3,2,3,4])
```

```
Out[21]: True
```

Enfin, pour le calcul de \mathbb{O}_{1323} dans la base \mathbb{Q} , nous calculons d'abord $\mathbb{O}_{12} = \mathbb{Q}_{12} - \mathbb{Q}_{21}$. Ensuite, nous insérons dans chacun des deux termes, et successivement, en dernière lettre, un 2 qui augmente le 2 déjà présent, puis un 3 (sans augmentation cette fois). Nous obtenons donc ce changement de base.

In [22]: $\mathbb{Q}(0[1,3,2,3])$

Out [22]: $\mathbb{Q}[1323] - \mathbb{Q}[3123]$

Pour conclure cet exemple, voici le changement de base complet.

In [23]: $\mathbb{Q}(0[p])$

Out [23]: $\mathbb{Q}[761424533] + \mathbb{Q}[762434511] + \mathbb{Q}[761434522] + \mathbb{Q}[761323544] - \mathbb{Q}[764124533] - \mathbb{Q}[764234511] - \mathbb{Q}[764134522] - \mathbb{Q}[763123544] - \mathbb{Q}[765142433] - \mathbb{Q}[765243411] - \mathbb{Q}[765143422] - \mathbb{Q}[765132344] + \mathbb{Q}[765412433] + \mathbb{Q}[765423411] + \mathbb{Q}[765413422] + \mathbb{Q}[765312344]$

Produits dans les bases \mathbb{O} et \mathbb{P} .

Les produits dans les bases \mathbb{O} et \mathbb{P} ne sont pas encore implantés. Dans l'état actuel, pour le calcul d'un produit, la machine effectue un changement de base vers la base \mathbb{Q} ou \mathbb{R} (ou le produit est implanté), puis elle calcule le produit et termine en inversant le changement de base. Cependant, comme nous l'avons vu, retourner dans la base initiale est très lent. En conclusion, nous constatons que les temps de calculs deviennent très longs à partir de la taille 5.

Le même raisonnement s'applique pour le coproduit, les demi-produits et demi-coproduits.

Sur le notebook en ligne vous pourrez retrouver un exemple de calcul à la main d'un produit de greffe et une vérification par la machine. Cette partie est la plus susceptible d'être améliorée dans un avenir proche.

8.4 Autres bases de WQSym

Nous nous intéressons ici aux bases GL et SR , définies par Y. Vargas (sous d'autres noms) [Var19] et fondées sur les ordres partiels que nous avons présentés dans la [section 2.3](#) et le deuxième notebook.

Ce qui nous intéresse le plus ici ce sont les demi-produits et demi-coproduits dans ces bases. Nous n'avons pas eu besoin de les planter directement. En effet, comme pour les produits des bases \mathbb{O} et \mathbb{P} nous effectuons un changement de base vers une base où le produit est implanté pour ensuite rechanter de base. Tout ce qui était nécessaire ici était simplement d'implanter les changements de bases. Nous avons ainsi pu faire des tests à grande échelle sur ces bases GL et SR puis constater que l'isomorphisme proposé par Vargas n'était pas un isomorphisme bidendriforme.

Automorphisme de Vargas

La base \mathbb{GL} (pour ‘greater left’)

C'est une base multiplicative. Elle est définie comme la somme de l'idéal supérieur pour l'ordre faible gauche sur la base \mathbb{Q}

$$\mathbb{GL}_u := \sum_{u \leq_L v} \mathbb{Q}_v.$$

Pour faire des tests sur cette base, nous invitons le lecteur à utiliser le notebook en ligne.

La base \mathbb{SR} (pour ‘smaller right’)

C'est une base multiplicative. Elle est définie comme la somme de l'idéal inférieur pour l'ordre faible droit sur la base \mathbb{R}

$$\mathbb{SR}_u := \sum_{v \leq_R u} \mathbb{R}_v.$$

Pour faire des tests sur cette base, nous invitons le lecteur à utiliser le notebook en ligne.

Isomorphisme de Vargas

L'isomorphisme d'algèbre de Hopf de Vargas est défini par

$$\mathbb{GL}_w \mapsto \mathbb{SR}_{w[:: -1]}$$

où $w[:: -1]$ désigne le mot w lu de droite à gauche.

C'est bien un morphisme de Hopf (facilement vérifiable sur les produits, plus difficilement sur les coproduits), nous donnons des exemples pour une rapide vérification.

In [9]: $\mathbb{SR}[\text{p1}] * \mathbb{SR}[\text{p2}]$

Out [9]: $\mathbb{SR}[978873212456]$

In [10]: $\mathbb{GL}[\text{p1}[:, :-1]] * \mathbb{GL}[\text{p2}[:, :-1]]$

Out [10]: $\mathbb{GL}[654212378879]$

In [11]: $\mathbb{SR}.\text{tensor}(\mathbb{SR})(\mathbb{R}(\mathbb{SR}[\text{p1}]).\text{coproduct}())$

Out [11]: $\mathbb{SR}[] \# \mathbb{SR}[3212456] + \mathbb{SR}[1] \# \mathbb{SR}[212345] + \mathbb{SR}[1] \# \mathbb{SR}[211345] + \mathbb{SR}[21] \# \mathbb{SR}[11234] + \mathbb{SR}[212] \# \mathbb{SR}[1234] + \mathbb{SR}[3212] \# \mathbb{SR}[123] + \mathbb{SR}[32124] \# \mathbb{SR}[12] + \mathbb{SR}[321245] \# \mathbb{SR}[1] + \mathbb{SR}[3212456] \# \mathbb{SR}[]$

```
In [12]: GL.tensor(GL)(Q(GL[p1[:::-1]]).coproduct())

Out[12]: GL[] # GL[6542123] + GL[1] # GL[543112] + GL[1] # GL[543212] +
          GL[12] # GL[43211] + GL[212] # GL[4321] + GL[2123] # GL[321] +
          GL[42123] # GL[21] + GL[542123] # GL[1] + GL[6542123] # GL[]
```

Par contre, ce n'est pas un morphisme bidendriforme, nous l'explicitons par un contre-exemple.

```
In [13]: SR((R(SR[1,2])>>R(SR[2,1,1])))

Out[13]: -SR[31234] + SR[33124] + SR[43123]

In [14]: SR((R(SR[1,2])<<R(SR[2,1,1])))

Out[14]: -SR[43123] + SR[31234] - SR[33124] + SR[43312]
```

```
In [15]: GL((Q(GL[2,1])>>Q(GL[1,1,2])))

Out[15]: -GL[41223] + GL[21334]

In [16]: GL((Q(GL[2,1])<<Q(GL[1,1,2])))

Out[16]: GL[41223]
```

En effet, le nombre de termes est différent alors que le morphisme est homogène.

8.5 Séries génératrices

Nous nous intéressons ici aux calculs de séries génératrices des mots tassés et des sous-ensembles des mots tassés (irréductibles, rouges-irréductibles, bleus-irréductibles, doubles-irréductibles).

Nous ajoutons ici une deuxième graduation qui correspond à la valeur maximale. Par exemple le terme $(24*r^4 + 36*r^3 + 14*r^2 + r)*z^4$ signifie que pour les mots tassés de taille 4 : il y en a 24 dont le maximum est 4, 36 dont le maximum est 3, 14 dont le maximum est 2 et 1 seul dont le maximum est 1.

Équations des théorèmes algébriques sur les séries

```
In [1]: from packed_words import *

PR.<r> = PolynomialRing(QQ)
PSR.<z> = PowerSeriesRing(PR)
# z : degré = longueur
# r : lettre max

N = 5 # À changer pour avoir plus de termes

In [2]: # Dimensions de WQSym+ (ou mots tassés non vides)
A = sum([z^n*sum(r^k*stirling_number2(n, k) * factorial(k)
                   for k in range(n+1))
         for n in range(1,6)]) + O(z^N)

Out[2]: r*z + (2*r^2 + r)*z^2 + (6*r^3 + 6*r^2 + r)*z^3 +
(24*r^4 + 36*r^3 + 14*r^2 + r)*z^4 + O(z^5)

In [3]: A == sum(z^n * sum(r^(max(p))
                           for p in PackedWords(n))
                  for n in range(1,N)) + O(z^N)

Out[3]: True

In [4]: # Dimensions de Prim(WQSym) (ou mots tassés irréductibles)
P = A / (1 + A)
P

Out[4]: r*z + (r^2 + r)*z^2 + (3*r^3 + 4*r^2 + r)*z^3 +
(13*r^4 + 23*r^3 + 11*r^2 + r)*z^4 + O(z^5)

In [5]: P == sum(z^n * sum(r^(max(p))
                           for p in PackedWords(n) if p.is_irreducible())
                  for n in range(1,5)) + O(z^N)

Out[5]: True

In [6]: # Dimensions de TPrim(WQSym) (ou mots tassés rouges-irréductibles
        # ou bleus-irréductibles)
T = A / (1 + A) ^ 2
T

Out[6]: r*z + r*z^2 + (r^3 + 2*r^2 + r)*z^3 +
(6*r^4 + 13*r^3 + 8*r^2 + r)*z^4 + O(z^5)
```

```
In [7]: T == sum(z^n * sum(r^(max(p))
                      for p in PackedWords(n)
                      if p.is_irreducible("red"))
                  for n in range(1,N)) + O(z^N)
```

Out[7]: True

Si nous regardons la série $\mathcal{A}/(1+\mathcal{A})^3$ sans le paramètre r , tous les termes sont positifs, nous pouvons donc supposer qu'il existe un sous-ensemble de mots tassés qui engendre librement tous les mots tassés en tant qu'algèbre bigraft (voir [définition 7.11](#)). C'est un exemple d'une des conjectures que nous avons faite, puis écartée grâce à l'expérimentation avec SageMath.

In [8]: A(z,1) / (1 + A(z,1))^3

Out[8]: $z + z^3 + 14z^4 + O(z^5)$

Cependant, quand nous rajoutons le paramètre r , nous remarquons un terme négatif.

In [9]: A / (1 + A)^3

Out[9]: $rz + (-r^2 + r)z^2 + rz^3 + (2r^4 + 6r^3 + 5r^2 + r)z^4 + O(z^5)$

Ce terme négatif implique qu'il n'existe pas de sous-ensemble de mots tassés qui engendre librement tous les mots tassés en tant qu'algèbre bigraft, comme nous l'avions supposé plus haut.

Voici la série des mots tassés à la fois rouges-irréductibles et bleus-irréductibles.

In [10]: # Mots tassés à la fois rouges-irréductibles et bleus-irréductibles
I = A / (1 + A)^3 + z*r*P
I

Out[10]: $rz + rz^2 + (r^3 + r^2 + r)z^3 + (5r^4 + 10r^3 + 6r^2 + r)z^4 + O(z^5)$

In [11]: I == sum(z^n * sum(r^(max(p))
 for p in PackedWords(n)
 if p.is_irreducible("bicolored"))
 for n in range(1,N)) + O(z^N)

Out[11]: True

Pompage pour obtenir les séries des forêts d'arbres bleus

Dans cette section, nous nous penchons sur une des nombreuses façons d'obtenir les séries liées aux forêts d'arbres bleus ([définition 4.67](#)). Nous utilisons ici la grammaire suivante :

- Forest $\rightarrow [] \mid \text{Tree}, \text{Forest}$
- Tree $\rightarrow \text{Node}, \text{Forest}$
- Node $\rightarrow 1^\circ \mid 1^\bullet, \text{Tree} \mid i^\alpha, \text{Tree}^*$ (avec $2 \leq i \leq \omega(\text{Tree}^*)$)

À partir de cette grammaire, nous avons implanté trois fonctions qui vont nous permettre de “pomper”, c'est-à-dire d'obtenir les termes d'un rang de plus des séries à chaque appel.

```
In [12]: def pump_forest(Tree, Forest):  
    return 1 + Tree * Forest  
  
def pump_ske(Node, Forest):  
    return Node * Forest  
  
def pump_blue_node(BTree):  
    res = r*z # correspond au mot 1  
    res+= z * BTree  
    # correspond à l'ajout d'un 1 à la fin sans augmentation  
    res+= (r + 1)*z * r^2 * (BTree/r).derivative(r)  
    # correspond à un noeud étiqueté avec i != 1  
    return res
```

```
In [13]: BForest = 1 + O(z)  
BTree = 0 + O(z)  
BNode = 0 + O(z)
```

```
In [14]: for _ in range(N-1):  
    BNode = pump_blue_node(BTree)  
    BTree = pump_ske(BNode, BForest)  
    BForest = pump_forest(BTree, BForest)
```

```
In [15]: BNode
```

```
Out[15]: r*z + r*z^2 + (r^3 + 2*r^2 + r)*z^3 +  
(6*r^4 + 13*r^3 + 8*r^2 + r)*z^4 + O(z^5)
```

```
In [16]: BTree
```

```
Out[16]: r*z + (r^2 + r)*z^2 + (3*r^3 + 4*r^2 + r)*z^3 +  
(13*r^4 + 23*r^3 + 11*r^2 + r)*z^4 + O(z^5)
```

```
In [17]: BForest
```

```
Out[17]: 1 + r*z + (2*r^2 + r)*z^2 + (6*r^3 + 6*r^2 + r)*z^3 +
(24*r^4 + 36*r^3 + 14*r^2 + r)*z^4 + 0(z^5)
```

Voici plus de détails pour le passage de la grammaire à la fonction pour la dernière étape :

$i^\alpha, Tree^*$ (avec $2 \leq i \leq \omega(Tree^*)$) qui devient $(r+1)*z*r^2 * (BTree/r).derivative(r)$.

```
In [18]: max1 = r*((BTree/(r))(z,0))
max1 # correspond aux mots dont le max est 1
```

```
Out[18]: r*z + r*z^2 + r*z^3 + r*z^4 + 0(z^5)
```

```
In [19]: max2 = r^2*((BTree-max1)/r^2)(z,0)
max2 # correspond aux mots dont le max est 2
```

```
Out[19]: r^2*z^2 + 4*r^2*z^3 + 11*r^2*z^4 + 0(z^5)
```

```
In [20]: max3 = r^3*((BTree-max1-max2)/r^3)(z,0)
max3 # correspond aux mots dont le max est 3
```

```
Out[20]: 3*r^3*z^3 + 23*r^3*z^4 + 0(z^5)
```

$BTree - max1$ correspond à l'ensemble des arbres pouvant être étiquetés par 2^α

$BTree - max1 - max2$ à l'ensemble des arbres pouvant être étiquetés par 3^α

$BTree - max1 - max2 - max3$ à l'ensemble des arbres pouvant être étiquetés par 4^α

[...]

La somme de tous ces ensembles correspond aux arbres pouvant être étiquetés par i^α .

```
In [21]: (BTree-max1) + (BTree-max1-max2) + (BTree-max1-max2-max3)
```

```
Out[21]: r^2*z^2 + (6*r^3 + 4*r^2)*z^3 +
(39*r^4 + 46*r^3 + 11*r^2)*z^4 + 0(z^5)
```

```
In [22]: BTree
```

```
Out[22]: r*z + (r^2 + r)*z^2 + (3*r^3 + 4*r^2 + r)*z^3 +
(13*r^4 + 23*r^3 + 11*r^2 + r)*z^4 + 0(z^5)
```

Nous remarquons que pour une puissance de z donnée, le coefficient de r^2 est compté 1 fois, le coefficient de r^3 est compté 2 fois, le coefficient de r^4 est compté 3 fois et ainsi de suite. La formule pour les arbres pouvant être étiquetés par i^α est donc la suivante :

```
In [23]: r^2*(BTree/r).derivative(r)
```

```
Out[23]: r^2*z^2 + (6*r^3 + 4*r^2)*z^3 +  
(39*r^4 + 46*r^3 + 11*r^2)*z^4 + O(z^5)
```

Lorsqu'un arbre de cet ensemble est étiqueté par i^α on ajoute 1 à la longueur du mot et le $r + 1$ correspond à la valeur d' α (on ajoute 1 à la valeur max si $\alpha = \circ$, zéro si $\alpha = \bullet$).

Finalement nous avons bien $(r + 1)*z * r^2 * (\text{BTree}/r).\text{derivative}(r)$

Pour conclure, nous obtenons bien les mêmes séries que plus haut.

```
In [24]: print(T - BNode)  
        print(P - BTREE)  
        print(A + 1 - BForest)
```

```
O(z^5)
```

```
O(z^5)
```

```
O(z^5)
```

Il est possible d'appliquer un raisonnement similaire pour les forêts d'arbres rouges.

Bibliographie

- [AS05] Marcelo Aguiar and Frank Sottile. Structure of the malvenuto–reutenauer hopf algebra of permutations. *Advances in Mathematics*, 191(2) :225–275, 2005. URL : <https://www.sciencedirect.com/science/article/pii/S0001870804000787>, doi:<https://doi.org/10.1016/j.aim.2004.03.007>.
- [BB94] Mary Katherine Bennett and Garrett Birkhoff. Two families of newman lattices. *algebra universalis*, 32 :115–144, 1994.
- [BDO20] Emily Burgunder and Bérénice Delcroix-Oger. Structure theorems for dendriform and tridendriform algebras. *IRMA Lectures in Mathematics and Theoretical Physics Vol. 31, Algebraic Combinatorics, Resurgence, Moulds and Applications (CARMA), Volume 1*, 2020. URL : https://www.ems-ph.org/books/book.php?proj_nr=251&srch=series%7Cirma, doi:<https://www.irif.fr/~bdelcroix/Recherche/stationnementFinal.pdf>.
- [BZ05] Nantel Bergeron and Mike Zabrocki. The Hopf algebras of symmetric functions and quasisymmetric functions in non-commutative variables are free and cofree, 2005. [arXiv:math/0509265](https://arxiv.org/abs/math/0509265).
- [Cha02] Frédéric Chapoton. Un théorème de Cartier–Milnor–Moore–Quillen pour les bigèbres dendriformes et les algèbres braces. *J. of Pure and Applied Algebra*, 168(1) :1 – 18, 2002. URL : [http://www.sciencedirect.com/science/article/pii/S0022404901000524](https://www.sciencedirect.com/science/article/pii/S0022404901000524), doi:[https://doi.org/10.1016/S0022-4049\(01\)00052-4](https://doi.org/10.1016/S0022-4049(01)00052-4).
- [Cha08] Frédéric Chapoton. Operads and algebraic combinatorics of trees. *Séminaire Lotharingien de Combinatoire*, B58c, 2008.
- [CK98] Alain Connes and Dirk Kreimer. Hopf algebras, renormalization and noncommutative geometry. *Communications in Mathematical Physics*, 199(1) :203–242, dec 1998. URL : <https://doi.org/10.1007/s002200050499>, doi:[10.1007/s002200050499](https://doi.org/10.1007/s002200050499).
- [CPP17] Grégory Chatel, Vincent Pilaud, and Viviane Pons. The weak order on integer posets. 2017. URL : <https://arxiv.org/abs/1701.07995>, doi:[10.48550/ARXIV.1701.07995](https://arxiv.org/abs/1701.07995).
- [DHNT08] Gerard H. E. Duchamp, Florent Hivert, Jean-Christophe Novelli, and Jean-Yves Thibon. Noncommutative symmetric functions vii : Free quasi-symmetric

- functions revisited. 2008. URL : <https://arxiv.org/abs/0809.4479>, doi: [10.48550/ARXIV.0809.4479](https://doi.org/10.48550/ARXIV.0809.4479).
- [DHT01] G. Duchamp, F. Hivert, and J. Y. Thibon. Noncommutative symmetric functions vi : Free quasi-symmetric functions and related algebras, 2001. [arXiv: math/0105065](https://arxiv.org/abs/math/0105065).
- [Foi02a] L. Foissy. Les algèbres de hopf des arbres enracinés décorés, i. *Bulletin des Sciences Mathématiques*, 126(3) :193–239, 2002. URL : <https://www.sciencedirect.com/science/article/pii/S0007449702011089>, doi: [https://doi.org/10.1016/S0007-4497\(02\)01108-9](https://doi.org/10.1016/S0007-4497(02)01108-9).
- [Foi02b] L. Foissy. Les algèbres de Hopf des arbres enracinés décorés, II. *Bulletin des Sciences Mathématiques*, 126(4) :249 – 288, 2002. URL : [http://www.sciencedirect.com/science/article/pii/S0007449702011132](https://www.sciencedirect.com/science/article/pii/S0007449702011132), doi: [https://doi.org/10.1016/S0007-4497\(02\)01113-2](https://doi.org/10.1016/S0007-4497(02)01113-2).
- [Foi07] Loïc Foissy. Bidendriform bialgebras, trees, and free quasi-symmetric functions. *Journal of Pure and Applied Algebra*, 209(2) :439–459, 2007. doi: [10.1016/j.jpaa.2006.06.005](https://doi.org/10.1016/j.jpaa.2006.06.005).
- [Foi11] L. Foissy. Primitive elements of the Hopf algebra of free quasi-symmetric functions. *Contemporary Mathematics Combinatorics and Physics*, page 79–88, 2011. doi: [10.1090/conm/539/10629](https://doi.org/10.1090/conm/539/10629).
- [FS09] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, USA, 1 edition, 2009.
- [FSZ91] Philippe Flajolet, B. Salvy, and P. Zimmermann. Automatic Average-case Analysis of Algorithms. *Theoretical Computer Science*, 79(1) :37–109, 1991. URL : <https://hal.inria.fr/hal-00917732>, doi: [10.1016/0304-3975\(91\)90145-R](https://doi.org/10.1016/0304-3975(91)90145-R).
- [FZS88] Philippe Flajolet, Paul Zimmermann, and Bruno Salvy. Lambda-Upsilon-Omega : an assistant algorithms analyzer. Research Report RR-0876, INRIA, 1988. URL : <https://hal.inria.fr/inria-00075678>.
- [Ges83] Ira M. Gessel. Multipartite p-partitions and inner products of skew schur functions. volume 34, 1983.
- [Gir11] Samuele Giraudo. *Combinatoire algébrique des arbres*. PhD thesis, Université Paris Est,, 2011. URL : <http://igm.univ-mlv.fr/~giraudo/These/Manuscrit.pdf>.
- [Gir18] Samuele Giraudo. *Nonsymmetric Operads in Combinatorics*. Springer International Publishing, 2018. URL : <https://hal.archives-ouvertes.fr/hal-02080929>, doi: [10.1007/978-3-030-02074-3](https://doi.org/10.1007/978-3-030-02074-3).
- [GKL⁺95] I.M. Gelfand, D. Krob, A. Lascoux, B. Leclerc, V.S. Retakh, and J.Y. Thibon. Noncommutative Symmetrical Functions. *Adv. in Math.*, 112(2) :218 – 348, 1995. URL : <http://www.sciencedirect.com/science/article/pii/S0001870885710328>, doi: <https://doi.org/10.1006/aima.1995.1032>.

- [Hiv99] Florent Hivert. Combinatoire des fonctions quasi-symétriques. PhD thesis, Informatique fondamentale Université de Marne-la-Vallée, 1999. URL : <http://www.theses.fr/1999MARN0051>.
- [HP19] Florent Hivert and Vincent Pilaud. Signaletic operads, 2019. URL : <https://arxiv.org/abs/1906.02228>, doi:10.48550/ARXIV.1906.02228.
- [KLPN01] Daniel Krob, Matthieu Latapy, Jean-Christophe Novelli, and Duong Phan. Pseudo-permutations i : First combinatorial and lattice properties. FPSAC 2001 (Formal Power Series and Algebraic Combinatorics), 2001.
- [Kre97] Dirk Kreimer. On the hopf algebra structure of perturbative quantum field theories. 1997. URL : <https://arxiv.org/abs/q-alg/9707029>, doi:10.48550/ARXIV.Q-ALG/9707029.
- [KS00] Maxim Kontsevich and Yan Soibelman. Deformations of algebras over operads and deligne's conjecture, 2000. URL : <https://arxiv.org/abs/math/0001151>, doi:10.48550/ARXIV.MATH/0001151.
- [Ler07] Philippe Leroux. L-algebras, triplicial-algebras, within an equivalence of categories motivated by graphs, 2007. arXiv:0709.3453.
- [Lod01] Jean-Louis Loday. Dialgebras, 2001. arXiv:math/0102053.
- [Lot02] M. Lothaire. Algebraic Combinatorics on Words. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2002. doi:10.1017/CBO9781107326019.
- [LR98] Jean-Louis Loday and María O. Ronco. Hopf Algebra of the Planar Binary Trees. Adv. in Math., 139(2) :293 – 309, 1998. URL : <http://www.sciencedirect.com/science/article/pii/S0001870898917595>, doi: <https://doi.org/10.1006/aima.1998.1759>.
- [LR02] Jean-Louis Loday and Maria O. Ronco. Order structure on the algebra of permutations and of planar binary trees. Journal of Algebraic Combinatorics, 2002.
- [LS96] Alain Lascoux and Marcel Paul Schützenberger. Treillis et bases des groupes de coxeter. Electr. J. Comb., 3(2), 1996. URL : <http://dblp.uni-trier.de/db/journals/combinatorics/combinatorics3.html#LascouxS96>.
- [Man12] Anthony Mansuy. The bigraft algebras, 2012. URL : <https://arxiv.org/abs/1206.5590>, doi:10.48550/ARXIV.1206.5590.
- [Mar08] Martin Markl. Operads and PROPs. Handbook of algebra., 5 :87–140, 2008.
- [Mlo22a] Hugo Młodecki. Decompositions of packed words and self duality of word quasisymmetric functions, 2022. URL : <https://arxiv.org/abs/2205.13949>, doi:10.48550/ARXIV.2205.13949.
- [Mlo22b] Hugo Młodecki. Notebooks on wqsym. https://github.com/HuxoD142/Notebooks_WQSym, 2022.

- [MM65] John W. Milnor and John C. Moore. On the structure of hopf algebras. *Annals of Mathematics*, 81 :211, 1965.
- [MR95] C. Malvenuto and C. Reutenauer. Duality between Quasi-Symmetrical Functions and the Solomon Descent Algebra. *J. of Algebra*, 177(3) :967 – 982, 1995. URL : <http://www.sciencedirect.com/science/article/pii/S0021869385713361>, doi:<https://doi.org/10.1006/jabr.1995.1336>.
- [NT03] Jean-Christophe Novelli and Jean-Yves Thibon. A hopf algebra of parking functions, 2003. URL : <https://arxiv.org/abs/math/0312126>, doi:[10.48550/ARXIV.MATH/0312126](https://doi.org/10.48550/ARXIV.MATH/0312126).
- [NT06] J. C. Novelli and J. Y. Thibon. Polynomial realizations of some trialgebras, 2006. arXiv:[math/0605061](https://arxiv.org/abs/math/0605061).
- [NW75] Albert Nijenhuis and Herbert S. Wilf. *Combinatorial algorithms / Albert Nijenhuis and Herbert S. Wilf*. Academic Press New York, 1975.
- [Pon13] V. Pons. *Combinatoire algébrique liée aux ordres sur les permutations*. Thèse de Doctorat, Université Paris-Est Marne-la-Vallée, 2013.
- [PR04] Patricia Palacios and María Ronco. Weak bruhat order on the set of faces of the permutohedra, 2004. URL : <https://arxiv.org/abs/math/0404352>, doi:[10.48550/ARXIV.MATH/0404352](https://doi.org/10.48550/ARXIV.MATH/0404352).
- [Ron00] María Ronco. Primitive elements in a free dendriform algebra. *Contemp. Math.*, 267, 01 2000. doi:[10.1090/conm/267/04274](https://doi.org/10.1090/conm/267/04274).
- [Ron02] María Ronco. Eulerian idempotents and Milnor–Moore theorem for certain non-cocommutative Hopf algebras. *J. of Algebra*, 254 :152–172, 08 2002. doi:[10.1016/S0021-8693\(02\)00097-2](https://doi.org/10.1016/S0021-8693(02)00097-2).
- [Sag20] Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.1.beta7)*, 2020. <https://www.sagemath.org>.
- [SI20] Neil J. A. Sloane and The OEIS Foundation Inc. The on-line encyclopedia of integer sequences, 2020. URL : <http://oeis.org/>.
- [Var19] Y. Vargas. *Algèbres de Hopf de mots tassés et de fonctions motifs*. PhD thesis, Université du Québec à Montréal, 2019.
- [VL12] Bruno Vallette and Jean-Louis Loday. *Algebraic Operads*. Grundlehren der mathematischen Wissenschaften. Springer, August 2012. URL : <https://hal.archives-ouvertes.fr/hal-00872465>, doi:[10.1007/978-3-642-30362-3](https://doi.org/10.1007/978-3-642-30362-3).
- [Wik22] Wikipédia. Axiomes de peano — wikipédia, l’encyclopédie libre, 2022. [En ligne ; Page disponible le 11-septembre-2022]. URL : http://fr.wikipedia.org/w/index.php?title=Axiomes_de_Peano&oldid=196870349.

Annexes

In [tables A.1 à A.6](#) we have red-packed forests, blue-packed forest and bicolored-packed forests associated to all packed words of size smaller than 4.

In [table A.7](#) we have the isomorphism between **WQSym** (bases \mathbb{O} and \mathbb{Q}) and its dual (bases \mathbb{P} and \mathbb{R}) for size smaller than 3. Basis \mathbb{O} and \mathbb{P} are indexed by bicolored-packed forests. This illustrates the main theorem of [section 6.4](#).

In [table A.8](#) we have the involution of [corollaire 6.25](#) for all packed words of size 4. They are organized by evaluations. Red-irreducible (resp. bleu-irreducible) packed words are underlined in red (resp blue) in the first (resp. second) column.

In [table A.9](#) we have the involution of [corollaire 6.25](#) for all red-irreducible packed words that are not blue-irreducible. It correspond to words underlined in red in front of a word underlined in blue in [table A.8](#).

The matrix of [figure A.1](#) is redundant with the column \mathbb{R} and \mathbb{P} of [table A.7](#). Note that even though the matrix of [figure A.2](#) is symmetric, it is not the case anymore on [figure A.5](#). Even if we restrict to permutations, the matrix is not symmetric for size 5.

	123	132	213	231	312	321	122	212	221	112	121	211	111
123	1
132	.	1	-1	.	1
213	-1	.	1
231	-1	-1	1	1	-1
312	.	.	-1	.	1
321	1	.	.	-1	-1	1
122	1
212	1	1
221	1	-1	.	.	.
112	1	.	.	.
121	-1	.	.	.	1	1	.
211	-1	1	.
111	1

FIGURE A.1 – Change-of-basis matrix from \mathbb{P}_3 to \mathbb{R}_3 .

w	$F_R(w)$	$F_B(w)$	$F_{RB}(w)$	$F_{BR}(w)$
1	(1)	(1°)	(1)	(1)

w	$F_R(w)$	$F_B(w)$	$F_{RB}(w)$	$F_{BR}(w)$
12	(1)	(1°)	(1)	(1)
21	(1) (1)	(1°) (1°)	(1) (1)	(1) (1)
11	(1, 2)	(1°) (1°)	(11)	(11)

w	$F_R(w)$	$F_B(w)$	$F_{RB}(w)$	$F_{BR}(w)$
123	(1)	(1°)	(1)	(1)
132	(2)	(2°)	(132)	(132)
213	(1) (1)	(1°) (1°)	(1) (1)	(1) (1)
231	(1)	(1°) (1°)	(1) (1)	(1) (1)
312	(1)	(1°) (1°)	(1) (1)	(1) (1)
321	(1) (1) (1)	(1°) (1°) (1°)	(1) (1) (1)	(1) (1) (1)
122	(1, 2)	(2°)	(11)	(11)
212	(1, 3)	(1°) (1°)	(11) (1)	(11)
221	(1, 2)	(1°) (1°)	(11) (1)	(1) (11)
112	(1, 2)	(1°) (1°)	(11)	(11)
121	(2)	(1°)	(121)	(121)
211	(1) (1, 2)	(1°) (1°)	(1) (11)	(11) (1)
111	(1, 2, 3)	(1°) (1°)	(111)	(111)

TABLE A.1 – All packed words of size smaller than 3 and forests associated to it.

w	$T_R(w)$	$T_B(w)$	$T_{RB}(w)$	$T_{BR}(w)$
1234				
1243				
1324				
1342				
1423				
1432				
2134				
2143				
2314				
2413				
3124				
3142				
3214				

TABLE A.2 – Packed words of size 4 and associated forests (part 1).

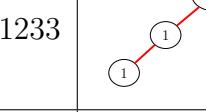
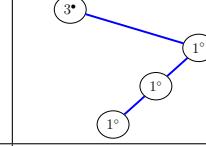
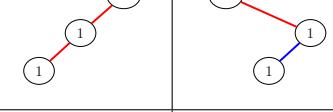
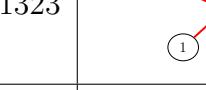
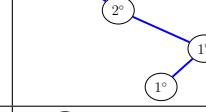
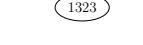
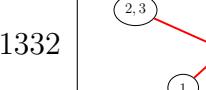
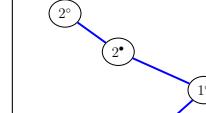
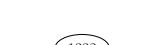
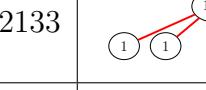
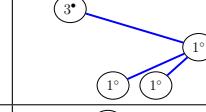
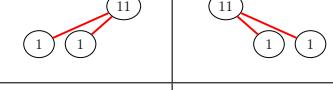
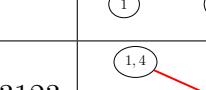
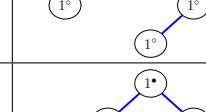
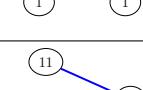
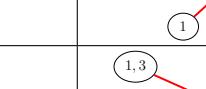
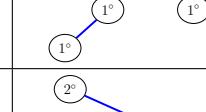
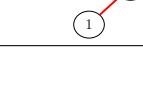
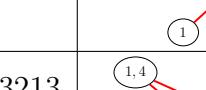
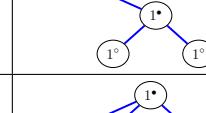
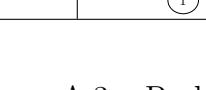
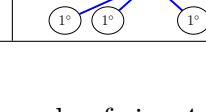
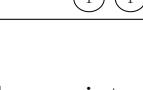
w	$T_R(w)$	$T_B(w)$	$T_{RB}(w)$	$T_{BR}(w)$
1233				
1323				
1332				
2133				
2313				
3123				
3132				
3213				

TABLE A.3 – Packed words of size 4 and associated forests (part 2).

	123	132	213	231	312	321	122	212	221	112	121	211	111
123	1
132	.	1
213	.	.	1
231	1
312	.	.	.	1
321	1
122	1	1
212	1
221	1	1	.	.
112	1	.	.	.
121	1	.	1	.	.
211	1
111	1	.

FIGURE A.2 – Change-of-basis matrix from \mathbb{Q}_3 to \mathbb{R}_3 .

w	$T_R(w)$	$T_B(w)$	$T_{RB}(w)$	$T_{BR}(w)$
1223				
1232				
1322				
2123				
2132				
2213				
2312				

TABLE A.4 – Packed words of size 4 and associated forests (part 3).

w	$T_R(w)$	$T_B(w)$	$T_{RB}(w)$	$T_{BR}(w)$
1123				
1132				
1213				
1231				
1312				
1321				
2113				
2131				

TABLE A.5 – Packed words of size 4 and associated forests (part 4).

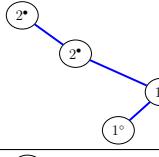
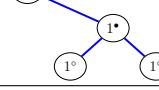
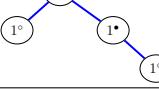
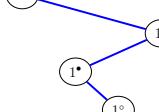
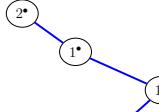
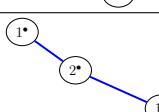
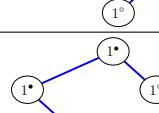
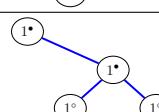
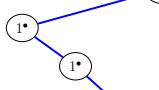
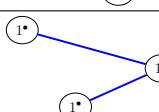
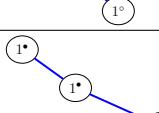
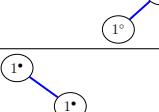
w	$T_R(w)$	$T_B(w)$	$T_{RB}(w)$	$T_{BR}(w)$
1222				
2122				
2212				
1122				
1212				
1221				
2112				
2121				
1112				
1121				
1211				
1111				

TABLE A.6 – Packed words of size 4 and associated forests (part 5).

\mathbb{R}_1	$\mathbb{P}_{(1)}$	$\mathbb{O}_{(1)}$	\mathbb{Q}_1
$\mathbb{R}_{12} - \mathbb{R}_{21}$	$\mathbb{P}_{(1)} \circ (1)$	$\mathbb{O}_{(1)} \circ (1)$	$\mathbb{Q}_{12} - \mathbb{Q}_{21}$
\mathbb{R}_{21}	$\mathbb{P}_{(1, 1)}$	$\mathbb{O}_{(1, 1)}$	\mathbb{Q}_{21}
\mathbb{R}_{11}	$\mathbb{P}_{(11)}$	$\mathbb{O}_{(11)}$	\mathbb{Q}_{11}

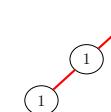
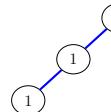
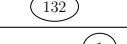
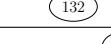
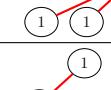
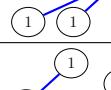
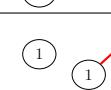
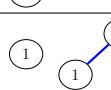
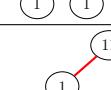
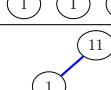
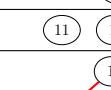
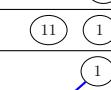
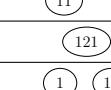
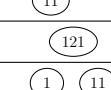
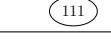
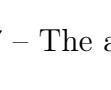
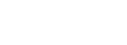
\mathbb{R}	\mathbb{P}	\mathbb{O}	\mathbb{Q}
$123 - 213 - 231 + 321$			$123 - 213 - 312 + 321$
$132 - 231$			$132 - 312$
$213 - 312 + 231 - 132$			$213 + 312 - 231 - 132$
$231 - 321$			$312 - 321$
$132 + 312 - 231 - 321$			$231 + 132 - 321 - 312$
321			321
$122 - 121 + 212 - 211$			$122 - 221$
212			$122 - 212$
221			211
$112 - 221$			$112 - 211$
121			$121 - 211$
$121 + 211$			221
111			111

TABLE A.7 – The automorphism of $\mathbf{WQSym}_{\leq 3}$.

1234	1234	1233	3123	1223	2123	1123	1123
1243	1243	1323	1323	1232	1232	1132	1132
1324	1324	1332	1332	1322	1322	1213	1213
1342	1342	2133	3213	2123	1223	1231	1231
1423	1423	2313	2313	2132	2132	1312	1312
1432	1432	2331	3212	2213	2113	1321	1321
2134	2134	3123	1233	2231	3112	2113	2213
2143	2143	3132	3132	2312	2131	2131	2312
2314	3124	3213	2133	2321	3121	2311	3312
2341	4123	3231	3122	3122	3231	3112	2231
2413	3142	3312	2311	3212	2331	3121	2321
2431	4132	3321	3211	3221	3221	3211	3321
3124	2314						
3142	2413						
3214	3214						
3241	4213						
3412	3412						
3421	4312						
4123	2341	1122	2112	1222	2212	1112	1112
4132	2431	1212	1212	2122	2122	1121	1121
4213	3241	1221	1221	2212	1222	1211	1211
4231	4231	2112	1122	2221	2111	2111	2221
4312	3421	2121	2121				
4321	4321	2211	2211	1111	1111		

TABLE A.8 – The involution $w \mapsto \hat{w}$ on packed words of size 4.

23514	41253	24314	31424	23413	31242	22413	31142
24513	41352	24413	31442	24313	31422	23412	31241
25314	41523	41234	12344	32413	32142	24213	31412
25413	41532	41324	13244	34123	23141	24312	31421
32514	42153	42134	21344	34213	32141		
35124	34152	42314	31244	32313	23133		
35214	43152	42413	34142	33123	12333		
		43124	23144	33213	21333		
		43214	32144				
23213	21313	22312	21131	22212	12222	24113	33142
23312	21331	23212	21311				
31223	21233						
32123	12233	31123	11233	23112	22131	22112	11222
32213	21133	31213	12133				
32312	23131	32113	22133	21112	11122		

TABLE A.9 – The involution $w \mapsto \hat{w}$ on red-irreducible packed words that are not blue-irreducible of size 5.

1234	1	1234	
1243	1	1243	
1324	.	1324	
1423	.	1423	
1432	.	1432	
2134	-1	2134	
2143	-1	2143	
2314	-1	2314	
2341	-1	2341	
2413	-1	2413	
2431	-1	2431	
3124	1	3124	
3142	1	3142	
3214	1	3214	
3412	1	3412	
3421	1	3421	
4123	.	4123	
4132	.	4132	
4213	.	4213	
4231	.	4231	
4312	.	4312	
4321	-1	4321	
1233	.	1233	
1323	.	1323	
1332	.	1332	
2133	.	2133	
2331	.	2331	
3123	.	3123	
3132	.	3132	
3213	.	3213	
3231	.	3231	
3312	.	3312	
3321	.	3321	
1223	.	1223	
1232	.	1232	
1322	.	1322	
2123	.	2123	
2132	.	2132	
2213	.	2213	
2231	.	2231	
2312	.	2312	
2321	.	2321	
3122	.	3122	
3212	.	3212	
3221	.	3221	
1122	.	1122	
1132	.	1132	
1213	.	1213	
1312	.	1312	
1321	.	1321	
2113	.	2113	
2131	.	2131	
2311	.	2311	
3112	.	3112	
3121	.	3121	
3211	.	3211	
1122	.	1122	
1212	.	1212	
2112	.	2112	
2211	.	2211	
1112	.	1112	
1121	.	1121	
2111	.	2111	
1111	.	1111	

FIGURE A.3 – Change-of-basis matrix from \mathbb{P}_4 to \mathbb{R}_4 .

FIGURE A.4 – Change-of-basis matrix from \mathbb{O}_4 to \mathbb{Q}_4 .

1234	1	1234
1243	1	1243
1324	1	1324
1423	1	1423
1432	1	1432
2134	1	2134
2143	1	2143
2314	1	2314
2341	-1 1	2341
2413	1 -1	2413
2431	1 -1	2431
3124	1	3124
3142	1	3142
3214	1	3214
3241	1	3241
3412	1	3412
4123	1	4123
4132	1	4132
4213	1	4213
4231	1	4231
4312	1	4312
4321	1	4321
1233	1 1	1233
1323	1 1	1323
1332	1 1	1332
2133	1 1	2133
2313	-1 1	2313
2331	1	2331
3132	1	3132
3213	1	3213
3231	-1 1	3231
3312	1	3312
3321	1	3321
1223	1	1223
1232	1	1232
1322	1	1322
2123	1	2123
2132	1	2132
2213	1	2213
2231	1	2231
2312	1	2312
2321	1	2321
3122	1	3122
3212	1	3212
3221	1	3221
1123	1	1123
1132	1	1132
1213	1	1213
1231	1 -1 1 -1 1	1231
1312	1 -1	1312
1321	1	1321
2113	1	2113
2131	1 -1 1 -1 1	2131
2311	1 -1 1 -1 1	2311
3112	1	3112
3211	1	3211
1122	1	1122
1212	1	1212
1221	1	1221
2112	1	2112
2211	1	2211
1112	1	1112
1211	1	1211
2111	1	2111
1111	1	1111

FIGURE A.5 – Change-of-basis matrix from \mathbb{Q}_4 to \mathbb{R}_4 .

Jean Dubuffet, Arbre biplan (version I)

août 1968, époxy peint au polyuréthane (1ère épreuve), 72 x 61 x 48 cm,
coll. Fondation Dubuffet/© A.D.A.G.P. Paris

