

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

EXPLORATION DE L'OPÉRADE DUPLICIALE ET DE SES SOUS-OPÉRADES

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN MATHÉMATIQUES

PAR

KEVIN ABDALLAH

MAI 2024

UNIVERSITÉ DU QUÉBEC À MONTRÉAL
Service des bibliothèques

Avertissement

La diffusion de ce mémoire se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.12-2023). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

REMERCIEMENTS

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont contribué au succès de ce travail.

Tout d'abord, un grand merci à mon directeur de mémoire, Samuele Giraudo, pour son orientation, son soutien et ses précieux conseils tout au long de cette aventure académique. Ses connaissances, sa patience et son expertise ont été d'une aide inestimable pour moi.

Je suis également reconnaissant envers mes amis qui ont été là pour moi dans les moments les plus difficiles. Leur soutien, leurs encouragements et leur compréhension m'ont aidé à surmonter les défis et à rester concentré sur mes objectifs.

Je tiens également à remercier l'Université du Québec à Montréal (UQAM) et le Laboratoire de Combinatoire et d'Informatique Mathématique (LACIM) pour m'avoir offert cette formidable opportunité de travailler et de me développer dans un environnement académique stimulant. Leur soutien et leurs ressources ont joué un rôle crucial dans ma recherche et mon développement professionnel.

Je ne saurais oublier ma famille, pour leur amour inconditionnel et leur soutien constant. Ils ont été ma source d'inspiration et de motivation tout au long de ce parcours. Leur foi inébranlable en moi et leurs sacrifices ont été le pilier de ma réussite.

Enfin, je souhaite remercier tous ceux qui, de près ou de loin, ont contribué à l'élaboration de ce travail. Chacun de vous a joué un rôle essentiel dans la réalisation de cette mémoire.

Kevin Abdallah

TABLE DES MATIÈRES

| | |
|--|----|
| RÉSUMÉ | vi |
| INTRODUCTION | 1 |
| CHAPITRE 1 COMBINATOIRE | 5 |
| 1.1 Séries génératrices | 5 |
| 1.1.1 Séries génératrices ordinaires | 5 |
| 1.1.2 Séries génératrices exponentielles | 6 |
| 1.2 Espèces de structures | 6 |
| 1.2.1 Des espèces aux séries | 7 |
| 1.2.2 Morphismes d'espèces | 8 |
| 1.2.3 Opérations sur les espèces | 8 |
| 1.2.4 Structures monoidales sur les espèces | 10 |
| 1.3 Collections structurées | 11 |
| 1.3.1 Collections générales | 12 |
| 1.3.2 Collections graduées | 13 |
| 1.3.3 Opérations sur les collections graduées | 14 |
| 1.3.4 Séries génératrices sur les collections graduées | 15 |
| 1.4 Arbres et structures similaires | 16 |
| 1.4.1 Arbres enracinés | 16 |
| 1.4.2 Arbres ordonnés | 18 |
| 1.4.3 Collection des arbres enracinés | 18 |
| 1.4.4 Arbres syntaxiques | 20 |
| 1.4.5 Arbres binaires | 20 |

| | | |
|--|---|----|
| 1.4.6 | Arbres comme arbres binaires | 22 |
| 1.4.7 | Partition des arbres binaires | 23 |
| 1.4.8 | Arbres k -aires | 24 |
| 1.5 | Langage des arbres | 25 |
| CHAPITRE 2 COMBINATOIRE ALGÈBRIQUE | | 27 |
| 2.1 | Espaces polynomiaux gradués | 27 |
| 2.1.1 | Séries et polynômes sur collections | 27 |
| 2.1.2 | Espaces polynomiaux..... | 28 |
| 2.1.3 | Série de Hilbert | 29 |
| CHAPITRE 3 OPÉRADES | | 30 |
| 3.1 | Opérades non-symétriques | 30 |
| 3.1.1 | Opérades dans la catégorie des espaces vectoriels | 31 |
| 3.1.2 | Composition partielle | 31 |
| 3.1.3 | Opérades en tant qu'espèces de structures | 34 |
| 3.2 | Exemples d'opérades | 35 |
| 3.2.1 | Opérade des endomorphismes | 35 |
| 3.2.2 | Opérades libres | 36 |
| 3.2.3 | Opérade NAP | 36 |
| 3.3 | Présentation par générateurs et relations | 37 |
| 3.4 | Algèbres sur opérades | 38 |
| 3.5 | Morphismes d'opérades | 38 |
| CHAPITRE 4 OPÉRADE DUPLICIALE ET SOUS-OPÉRADES | | 40 |
| 4.1 | Opérade dupliciale | 40 |

| | | |
|-------|---|----|
| 4.1.1 | Indexation des nœuds : ordre infixe | 42 |
| 4.1.2 | Langage des arbres binaires et opérade dupliciale | 43 |
| 4.1.3 | Proposition sur les opérades dupliciales | 45 |
| 4.2 | Une sous-opérade dupliciale | 46 |
| 4.2.1 | Sous-opérades engendrées par un peigne | 48 |
| 4.2.2 | Sous-opérades engendrées par un singleton..... | 49 |
| | PERSPECTIVES ET CONCLUSION | 55 |
| | ANNEXE A ALGORITHME | 58 |
| | BIBLIOGRAPHIE | 61 |

RÉSUMÉ

La combinatoire est un domaine des mathématiques qui se spécialise dans le comptage et l'arrangement d'ensembles finis, utilisant souvent des outils tels que les séries génératrices pour comprendre des modèles complexes. Elle est essentielle pour comprendre les structures en théorie des graphes, en probabilité, et pour aider à optimiser les algorithmes en informatique. En combinatoire, les opérades sont utiles pour découvrir des caractéristiques de structures combinatoires et favoriser de nouvelles découvertes. Ce sont des structures algébriques qui définissent des opérations avec plusieurs entrées et une sortie, permettant la création d'opérations complexes et combinées. Les opérades se présentent sous différentes formes, par exemple, les opérades colorées et les opérades non-symétriques. Ce travail se concentre principalement sur l'opérade dupliciale, une opérade non-symétrique définie par l'insertion d'un nouvel arbre binaire dans un nœud d'un arbre existant. L'opérade dupliciale est engendrée par deux arbres binaires spécifiques et un ensemble de relations. Nous approfondissons cette étude en examinant Dup_G , la sous-opérade engendrée par un ensemble G . Plus spécifiquement, notre attention se porte sur des singletons G contenant un arbre à n nœuds, où la racine doit avoir un enfant gauche et un enfant droit, en utilisant les outils présentés dans les chapitres précédents pour explorer cette structure au sein du cadre dupliciale. Dans ce cadre, notre analyse révèle un aspect intéressant de Dup_G : le nombre $Dup_G(i)$, défini comme le nombre d'arbres binaires ayant i nœuds dans Dup_G , correspond au nombre d'arbres n -aires possédant exactement i feuilles, qui est compté en utilisant les suites de Fuss-Catalan. Ce résultat fournit une compréhension plus approfondie des propriétés énumératives de Dup_G et de sa relation avec la structure des arbres binaires.

INTRODUCTION

En combinatoire, l'accent est mis sur l'étude de groupes spécifiques d'objets, tels que les permutations, les mots, les arbres et les graphes, et sur la détermination de leur nombre en fonction de leur taille. Cela implique des stratégies concrètes telles que l'utilisation de séries génératrices, qui sont des outils pour calculer ces nombres, les bijections et la décomposition d'objets complexes en partie plus simples et plus petits. Le travail de Stanley (Stanley, 1999a) est une ressource complète sur la combinatoire énumérative. Les séries génératrices sont essentielles en combinatoire pour transformer des problèmes complexes de dénombrement en expressions algébriques. En représentant des suites sous forme de sommes dans des séries, ces séries permettent de résoudre des problèmes difficiles, tels que le dénombrement des différentes manières d'arranger des objets ou les issues possibles d'une série (Wilf, 2005). Par exemple, les séries génératrices peuvent résoudre efficacement des problèmes comme déterminer le nombre de chemins différents dans un graphe ou le nombre de façons de partitionner un nombre. De plus, pour comprendre un graphe complexe, Bóna (Bóna, 2011) offre une introduction accessible à la fois à l'énumération et à la théorie des graphes. La combinatoire pourrait impliquer de diviser un graphe en sous-graphes plus simples, puis d'appliquer ces techniques pour compter et analyser chaque pièce séparément, rendant ainsi le problème global plus gérable.

La combinatoire algébrique est une discipline fascinante qui se situe à l'intersection des problèmes combinatoires et des outils algébriques. Ce domaine emploie des méthodes algébriques pour simplifier et résoudre des problèmes combinatoires, comme illustré dans les travaux de Bergeron sur les espaces co-invariants et les fonctions symétriques dans (Bergeron, 2009). Par exemple, les polynômes de Schur ont une interprétation combinatoire en termes de tableaux de Young. Les coefficients dans le développement des polynômes de Schur peuvent être interprétés en termes du nombre de certains types de tableaux de Young, ce qui en fait un outil utile dans l'énumération combinatoire. La combinatoire algébrique crée des ponts entre des concepts mathématiques abstraits et concrets, en codant les résultats multiples d'une opération sous la forme d'une somme, conduisant à des avancées significatives dans le domaine.

Les opérades, en tant que concept en mathématiques, offrent un cadre polyvalent pour comprendre et structurer une large gamme de phénomènes algébriques. Originaires de la topologie algébrique dans les années 1970 pour l'étude des espaces de lacets (Boardman et Vogt, 1973) (May, 1972), les opérades ont

depuis connu une évolution significative. Ce sont des instruments puissants pour décrire les structures algébriques au sein des catégories monoidales symétriques. Au-delà de leur rôle fondamental dans la description des structures algébriques, les opérades peuvent être conceptualisées comme des mécanismes sophistiqués pour des opérations d'insertion. Au cœur de celles-ci, les opérades permettent l'insertion d'une opération dans une autre. Chaque opération ou élément est vu comme un nœud avec une certaine arité, indiquant le nombre d'entrées qu'il peut accueillir. Les applications de composition, centrales à la structure de l'opérade, permettent l'insertion de ces opérations les unes dans les autres. Cela est semblable à greffer un arbre dans un autre à une feuille spécifiée, un processus qui reflète la composition fonctionnelle en programmation ou les opérations algébriques en mathématiques, capturant l'essence de la façon dont des opérations complexes peuvent être construites à partir d'éléments plus simples. Les opérades existent sous différentes formes. Par exemple, les opérades cycliques, les opérades colorées et les opérades non-symétriques. Les opérades cycliques, par exemple, jouent un rôle essentiel dans l'étude des structures où les entrées et les sorties sont traitées de manière cyclique, ce qui permet de mieux comprendre des domaines tels que la topologie (Jean-Louis Loday, 2012). Les opérades colorées, quant à elles, étendent le concept en permettant des opérations de différents "types" ou "couleurs", élargissant leur applicabilité dans des domaines tels que la théorie des catégories et l'algèbre à plus haute dimension (Yau, 2016). Dans ce mémoire, nous parlerons principalement des opérades non-symétriques. Elles sont importantes en combinatoire, car elles traitent des opérations où l'ordre des éléments est important. Cela nous aide à mieux comprendre et résoudre les problèmes où la manière dont les choses sont arrangées est essentielle.

Un élément pivot de notre exploration dans ce mémoire est l'opérade dupliciale (également appelé l'opérade overunder) liée aux algèbres dupliciales. Une algèbre dupliciale, comme référencée dans (Loday et Ronco, 2002) et (Brouder et Frabetti, 2001), est un espace vectoriel doté de deux opérations bilinéaires et associatives, notées par \llcorner et \lrcorner . Ces opérations sont définies de sorte que l'équation $(x \lrcorner y) \llcorner z = x \lrcorner (y \llcorner z)$ soit vraie. Il est établi que l'algèbre dupliciale libre D , engendrée par un seul élément, a sa base représentée par des arbres binaires. Dans ce contexte, les opérations \llcorner et \lrcorner correspondent respectivement aux opérations binaires \setminus (under) et $/$ (over), comme détaillé dans (Loday et Ronco, 2002). Par conséquent, les tailles des composants homogènes dans cette structure algébrique suivent la série des nombres de Catalan, qui commence par 1, 2, 5, 14.

L'opérade dupliciale se définit à partir d'une opération de greffe particulière sur les arbres binaires. Elle se

concentre spécifiquement sur le processus d'insertion d'un nouvel arbre binaire dans un nœud d'un arbre existant. Le sous-arbre gauche du nœud remplacé doit être attaché à la feuille la plus à gauche du nouvel arbre greffé, et le sous-arbre droit être attaché à la feuille la plus à droite. Dans des contextes combinatoires, l'opéade dupliciale offre une perspective unique sur l'énumération et la catégorisation des structures arborescentes, améliorant notre compréhension de leurs propriétés intrinsèques et interrelations.

Dans ce mémoire, nous plongeons dans les subtilités des sous-opéades dupliciales, un domaine qui reste relativement inexploré. En appliquant itérativement la procédure de greffe avec des arbres binaires spécifiques comme éléments de base, nous pouvons engendrer n'importe quel arbre binaire. Cependant, la sélection de différents ensembles d'arbres binaires initiaux, nos générateurs, nous conduit à des sous-ensembles spécifiques d'arbres binaires, chacun avec des caractéristiques uniques. Ces sous-ensembles, connus sous le nom de sous-opéades dupliciales, forment le cœur de notre exploration.

Notre recherche vise à démêler les complexités de ces sous-opéades dupliciales, cherchant à comprendre les propriétés et l'énumération des arbres binaires qu'ils produisent, en fonction du choix des générateurs. Étant donné l'escalade rapide en complexité et en nombre de ces arbres, l'analyse manuelle devient impraticable. Pour résoudre ce défi, nous avons développé un algorithme, détaillé dans l'annexe A, qui aide à calculer la série génératrice pour divers ensembles de générateurs. Notre recherche utilise l'ordinateur de manière intensive et une implantation de cet algorithme en Python. Nos principales découvertes, présentées au chapitre 4, reposent fortement sur les suites de Fuss-Catalan. Nous avons réussi à déterminer le nombre et la structure des arbres engendrés à partir d'un singleton, où la racine a à la fois des enfants gauches et droits. De manière intéressante, des séries de Fuss-Catalan sont également apparues dans l'étude d'autres ensembles de générateurs, bien qu'une caractérisation précise des arbres correspondants reste insaisissable. De plus, nous avons rencontré des séries divergeant du type Fuss-Catalan, que nous discutons vers la fin de notre mémoire.

Cette exploration des sous-opéades dupliciales a non seulement éclairé nos questions initiales concernant la description et l'énumération des éléments au sein de ces opéades, mais a également dévoilé la vaste portée de notre investigation.

Ces questions marquent le début d'une recherche plus approfondie dans le domaine des sous-opéades

dupliciales, promettant d'enrichir notre compréhension de ce paysage mathématique fascinant.

Le mémoire se compose de quatre chapitres, chacun se basant sur le matériel du précédent. La structure du mémoire est la suivante :

1. Le chapitre 1 introduit les notions fondamentales telles que les séries génératrices, les arbres enracinés, les arbres binaires et les collections graduées,
2. le chapitre 2 utilise les notions du chapitre 1 de collections graduées pour créer un espace vectoriel et introduire l'espace polynomial gradué, un outil utilisé en combinatoire algébrique,
3. le chapitre 3 introduit le lecteur à la théorie des opérades, principalement les opérades non-symétriques,
4. le chapitre 4 donne un aperçu de l'opérade dupliciale, une opérade définie sur les arbres binaires. Ensuite, celui-ci portera sur les sous-opérades de l'opérade dupliciale. À ma connaissance, les méthodes développées et les résultats obtenus dans les sections suivant la définition 4.2 de ce chapitre sont originaux.

CHAPITRE 1

COMBINATOIRE

Ce chapitre initial présente des concepts fondamentaux de la combinatoire qui sont essentiels pour les parties suivantes de notre travail. Nous commençons par parler des séries génératrices, puis nous présentons le concept de collection combinatoire avec différents exemples basés sur des structures arborescentes.

1.1 Séries génératrices

Les séries génératrices sont un concept mathématique utile pour aborder des problèmes complexes d'énumération. En substance, elles représentent des séries de nombres, où chaque terme de la série correspond au nombre d'objets de taille n dans un problème d'énumération spécifique. Cette approche transforme des séries, comme les permutations ou les séries binaires, en expressions algébriques, encapsulant des problèmes de comptage complexes sous des formes plus simples. En transformant un problème de comptage en une équation algébrique, les fonctions génératrices dévoilent des relations dans des séries qui sont autrement cachées dans les méthodes de comptage différentes. Plus d'informations à ce sujet peuvent être trouvées dans (Flajolet et Sedgewick, 2009).

Exemple. Pour illustrer, considérons le dénombrement des façons d'allouer des objets identiques dans des boîtes distinctes. Avec k objets identiques et n boîtes distinctes, la série génératrice correspondante est

$$G(x) = (1 + x + x^2 + x^3 + \dots)^n = \left(\frac{1}{1-x} \right)^n,$$

où le coefficient de x^k dans la forme développée de $G(x)$ indique le nombre de distributions de k objets dans n boîtes. Cet exemple démontre l'utilité des séries génératrices pour encapsuler et calculer efficacement des solutions à des problèmes combinatoires.

1.1.1 Séries génératrices ordinaires

La série génératrice ordinaire d'une suite $\{a_n\}_{n=0}^{\infty}$ est la série entière formelle définie par

$$A(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \dots = \sum_{n=0}^{\infty} a_n x^n,$$

où a_n est le n^e terme de la série, et x est une indéterminée.

1.1.2 Séries génératrices exponentielles

La série génératrice exponentielle d'une suite $\{a_n\}_{n=0}^\infty$ est la série définie par

$$A(x) = a_0 + a_1x + \frac{a_2x^2}{2!} + \frac{a_3x^3}{3!} + \cdots = \sum_{n=0}^{\infty} \frac{a_nx^n}{n!}.$$

Les séries génératrices exponentielles sont particulièrement utiles pour dénombrer des objets étiquetés.

1.2 Espèces de structures

Bien que nous n'allons pas travailler de manière approfondie avec les espèces de structures, nous introduirons brièvement les notions de ce concept telles que présentées dans (Bergeron *et al.*, 1998). Il est essentiel de comprendre que les espèces occupent une place importante en combinatoire et peuvent s'avérer utiles pour notre sujet principal.

Définition 1.1 (Espèce) Une espèce de structure P consiste en ce qui suit.

- Pour chaque ensemble fini I , un ensemble $P[I]$.
- Pour chaque bijection $\sigma : I \rightarrow J$, une application $P[\sigma] : P[I] \rightarrow P[J]$ telle que

$$P[\sigma \circ \tau] = P[\sigma] \circ P[\tau] \text{ et } P[id] = id$$

où id est la fonction d'identité.

Un élément $x \in P[I]$ est appelé une P -structure sur I . L'application $P[\sigma]$ est le transport de P -structures le long de σ ou simplement une application de réétiquetage sur P . Notons que les $P[\sigma]$ sont nécessairement des bijections. Nous écrivons simplement "espèce" au lieu "d'espèce de structure".

Par exemple, considérons l'espèce des partitions Par . Pour un ensemble fini A , la collection $Par[A]$ consiste en toutes les manières de partitionner A en sous-ensembles non vides.

Soit $A = \{1, 2, 3\}$. Les Par -structures de $Par[A]$ sont

- $\{\{1\}, \{2\}, \{3\}\}$

- $\{\{1, 2\}, \{3\}\}, \{\{1, 3\}, \{2\}\}, \{\{2, 3\}, \{1\}\}$
- $\{\{1, 2, 3\}\}$

La fonction de transport pour une bijection $\sigma : A \rightarrow B$ est définie par $Par[\sigma](\pi) = \{\sigma(S) : S \in \pi\}$ pour chaque partition $\pi \in Par[A]$. Ainsi, si $\sigma : \{1, 2, 3\} \rightarrow \{a, b, c\}$ est une bijection telle que $\sigma(1) = a$, $\sigma(2) = b$ et $\sigma(3) = c$. Alors σ transforme la partition $\{\{1, 2\}, \{3\}\}$ en $\{\{a, b\}, \{c\}\}$.

Définition 1.2 L'espèce caractéristique des singletons X est définie par

$$X(I) = \begin{cases} \{I\} & \text{si } |I| = 1, \\ \emptyset & \text{sinon.} \end{cases}$$

Définition 1.3 L'espèce $\mathbb{1}$ est définie par

$$\mathbb{1}(I) = \begin{cases} \{*\} & \text{si } I = \emptyset, \\ \emptyset & \text{sinon,} \end{cases}$$

où $*$ est un objet quelconque.

La fonction de transport pour une bijection $\sigma : I \rightarrow J$ est définie par

$$\mathbb{1}[\sigma] = \begin{cases} \{id\} & \text{si } I = \emptyset, \\ \emptyset & \text{sinon.} \end{cases}$$

Définition 1.4 (Espèce positive) : une espèce P est dite **positive** si elle n'attribue aucune structure à l'ensemble vide. Autrement dit, $P(\emptyset) = \emptyset$.

1.2.1 Des espèces aux séries

Les espèces en combinatoire servent à structurer des ensembles finis de manière systématique. Leur série génératrice est définie par

$$P(x) = \sum_{n \geq 0} P_n \frac{x^n}{n!}$$

où P_n est le nombre de structures P sur un ensemble de n éléments, soit $P_n := |P[\{1, 2, \dots, n\}]|$. Cette série encapsule les caractéristiques combinatoires de l'espèce P de manière élégante et concise.

Exemple : considérons l'espèce des permutations. Une permutation sur un ensemble fini E est une bijection de E dans E . La série génératrice des permutations est donnée par

$$P(x) = \sum_{n \geq 0} n! \frac{x^n}{n!} = \sum_{n \geq 0} x^n = \frac{1}{1-x}.$$

Cette expression résulte du fait que les éléments d'un ensemble à n éléments peuvent être ordonnés de $n!$ façons différentes.

1.2.2 Morphismes d'espèces

Les morphismes d'espèces offrent un moyen de relier et de transformer des espèces, enrichissant ainsi notre compréhension des structures combinatoires. Un morphisme d'espèces $\alpha : P \rightarrow Q$ est constitué de fonctions $\alpha_I : P[I] \rightarrow Q[I]$ qui préservent la structure des bijections entre les ensembles :

$$Q[\sigma] \circ \alpha_I = \alpha_J \circ P[\sigma].$$

Cela assure la cohérence des structures à travers les bijections comme illustrée dans le diagramme commutatif

$$\begin{array}{ccc} P[I] & \xrightarrow{P[\sigma]} & P[J] \\ \alpha_I \downarrow & & \downarrow \alpha_J \\ Q[I] & \xrightarrow{Q[\sigma]} & Q[J] \end{array}$$

Exemple : un morphisme entre une espèce P de graphes et l'espèce Par , peut être défini de manière telle que l'image d'un graphe est partitionnée en deux blocs, la première composante étant les sommets qui sont connectés à d'autres sommets, la seconde composante étant les sommets qui ne sont connectés à rien.

Si toutes les fonctions α_I sont bijectives, α est alors un isomorphisme, indiquant une correspondance entre les structures des espèces P et Q . Nous disons alors que P et Q sont isomorphes en tant qu'espèces.

1.2.3 Opérations sur les espèces

Nous avons déjà exploré comment les espèces constituent un outil essentiel en combinatoire, permettant la création de structures spécifiques sur des ensembles finis. Dans la continuité de cette exploration, nous

nous pencherons sur la manière dont différentes espèces peuvent interagir et être manipulées conjointement. Cette section vise à introduire et à détailler des opérations sur les espèces qui, par leur nature, se traduisent en des opérations classiques sur les séries génératrices. Cela inclut des opérations telles que la somme, le produit de Cauchy et la substitution, chacune jouant un rôle clé dans leurs applications potentielles en combinatoire.

- Somme : soit P et Q deux espèces. La somme de ces espèces, notée $P + Q$, est définie par

$$(P + Q)[I] := P[I] \sqcup Q[I]$$

où \sqcup est l'union disjointe.

Une $P + Q$ -structure sur I est soit une P -structure sur I , soit une Q -structure sur I . De plus, si $x \in (P + Q)[I]$, alors $(P + Q)[\sigma](x) \in (P + Q)[J]$ et

$$(P + Q)[\sigma](x) = \begin{cases} P[\sigma](x) & \text{si } x \in P[I] \\ Q[\sigma](x) & \text{si } x \in Q[I]. \end{cases}$$

Soient $P(x)$ et $Q(x)$ les séries génératrices associées de P et Q . Il est clair que

$$(P + Q)(x) = P(x) + Q(x).$$

- Produit de Cauchy : soit P et Q deux espèces. Le produit de Cauchy de P et Q , noté $P.Q$ est défini par

$$\begin{aligned} (P.Q)[I] &= \bigsqcup_{S \subseteq I} P[S] \times Q[I \setminus S] \\ &= \bigsqcup_{I = S \sqcup T} P[S] \times Q[T], \end{aligned}$$

où \times est le produit cartésien de deux ensembles.

Ainsi, une $P.Q$ -structure sur I est un couple (p, q) où :

- p est une P -structure sur $S \subseteq I$
- q est une Q -structure sur $T \subseteq I$
- $I = S \sqcup T$ (c'est-à-dire que (S, T) est une décomposition de I)

De plus, l'application de réétiquetage d'une $P.Q$ -structure, étant donné une bijection $\sigma : I \rightarrow J$, est définie comme suit :

$$\begin{aligned} (P.Q)[\sigma] : (P.Q)[I] &\longrightarrow (P.Q)[J] \\ (p, q) &\longrightarrow (P.Q)[\sigma](p, q) = (P[\sigma|_S](p), Q[\sigma|_T](q)) \end{aligned}$$

où la notation $\pi|_H$ signifie la restriction de la fonction π sur H .

De plus, si $P(x)$ est la série génératrice de P et $Q(x)$ la série génératrice de Q , alors

$$(P.Q)(x) = P(x).Q(x).$$

- Substitution : soit P et Q deux espèces, où Q est une espèce positive. L'espèce substitution, notée $P \circ Q$, est définie par

$$(P \circ Q)[I] = \bigsqcup_{\pi \vdash I} P[\pi] \times \prod_{A \in \pi} Q[A].$$

Ainsi, une $P \circ Q$ -structure sur I est un triplet (π, p, q) , où :

- $\pi \vdash I$ signifie que π est une partition de I ,
- p est une P -structure sur π ,
- $q = (q_A)_{A \in \pi}$ et q_A est une Q -structure sur A .

De plus, l'application de réétiquetage de $P \circ Q$, étant donné une bijection $\sigma : I \rightarrow J$, est définie comme suit :

$$(P \circ Q)[\sigma](\pi, p, q) = (\pi', p', q'),$$

où (π', p', q') est définie de la manière suivante :

- $\pi' = \{\sigma(A) : A \in \pi\} \vdash J$,
- $p' = P[\sigma^*](p)$ où $\sigma^* = Par[\sigma](\pi)$,
- $q' = (q'_B)_{B \in \pi'}$ où $B = \sigma(A)$ et $q'_B = Q[\sigma|_A](q_A)$.

Si $P(x)$ est la série génératrice de P et $Q(x)$ est la série génératrice de Q , alors

$$(P \circ Q)(x) = P(Q(x)).$$

1.2.4 Structures monoidales sur les espèces

Définition 1.5 Une espèce **Monoïdale** est une espèce M accompagnée de morphismes d'espèces

$$\mu : M.M \longrightarrow M$$

et

$$i : \mathbf{1} \longrightarrow M,$$

tel que le diagramme ci-dessous commute

$$\begin{array}{ccccc}
 \mathbb{1}.M & \xrightarrow{i \times id} & M.M & \xleftarrow{id \times i} & M.\mathbb{1} \\
 & \searrow & \downarrow \mu & \swarrow & \\
 & & M & &
 \end{array}$$

Le morphisme μ consiste en des applications :

$$(M.M)[I] = \bigsqcup_{I=S \sqcup T} M[S] \times M[T] \longrightarrow M[I].$$

C'est-à-dire,

$$\begin{aligned}
 \mu_{S,T} : M[S] \times M[T] &\longrightarrow M[I] \\
 (x, y) &\longrightarrow \mu_{S,T}(x, y)
 \end{aligned}$$

pour tous S, T et $I = S \sqcup T$, telle que, en notant $\mu_{S,T}(x, y)$ par $x.y$,

- pour tous ensembles finis I , décompositions $I = S \sqcup T$, et M -structures $x \in M[S]$ et $y \in M[T]$:

$$M[\sigma](\mu_{S,T}(x, y)) = \mu_{\sigma(S), \sigma(T)}(M[\sigma|_S](x), M[\sigma|_T](y)). \quad (\text{Naturalité})$$

- Pour tous ensembles finis I , décompositions $I = S \sqcup T \sqcup R$, et M -structures $x \in M[S]$, $y \in M[T]$ et $z \in M[R]$:

$$(x.y).z = x.(y.z) \quad (\text{Associativité})$$

- Pour tous ensembles finis I , et M -structures $x \in M[I]$:

$$1.x = x = x.1 \quad (\text{Unitarité})$$

Exemple : si I est un ensemble tel que $I = S \sqcup T$, $\pi \vdash S$ et $\pi' \vdash T$, Par est un monoïde avec l'opération

$$\mu_{S,T}(\pi, \pi') = \pi \cup \pi'.$$

1.3 Collections structurées

Cette section, ainsi que la suite du chapitre 1, le chapitre 2 et le chapitre 3 de ce mémoire, s'appuie largement sur l'ouvrage *Nonsymmetric Operads in Combinatorics* de Giraudo Samuele (Giraudo, 2018) et sur (Giraudo, 2017). La représentation des arbres et leur langage nous ont permis de coder ces structures dans notre algorithme et de faciliter la présentation et les démonstrations des théorèmes.

1.3.1 Collections générales

Considérons un ensemble I appelé index I . Avec cet index, nous pouvons construire ce qu'on appelle une I -collection, symbolisée par C . Pensez à C comme une union de sous-ensembles, où chaque sous-ensemble est associé de manière unique à un élément de I :

$$C = \bigsqcup_{i \in I} C(i).$$

Chaque sous-ensemble $C(i)$ contient des objets associés à l'index i . Lorsque nous choisissons un objet x de C qui appartient à $C(i)$, nous disons que x est un i -objet. L'index de cet objet, noté $\text{ind}(x)$, est naturellement i .

Si chaque sous-ensemble $C(i)$ est fini, alors notre collection C est *combinatoire*. Si C lui-même a un nombre fini d'objets, c'est simplement une collection finie. La collection la plus minimale, celle sans objet, est l'ensemble vide, \emptyset . Si notre ensemble d'indices I est un singleton, alors C est une collection *simple*. Cela signifie que chaque ensemble habituel peut être vue comme une collection simple et l'inverse est également vrai.

Par exemple, nous proposons une méthode pour classifier l'ensemble des permutations de n éléments, fondée sur la longueur du cycle le plus long dans leur décomposition cyclique. Cette classification est organisée comme suit : pour chaque $i \in [n]$, le sous-ensemble $C(i)$ regroupe toutes les permutations dont le cycle le plus long est de i éléments. Pour $n = 4$, nous classons les permutations de l'ensemble $I = \{1, 2, 3, 4\}$ dans les sous-ensembles tels que :

- $C(1) = \{(1)(2)(3)(4)\}$.
- $C(2) = \{(12)(3)(4), (13)(2)(4), (14)(2)(3), (23)(1)(4), (24)(1)(3), (34)(1)(2), (12)(34), (13)(24), (14)(23)\}$.
- $C(3) = \{(123)(4), (132)(4), (124)(3), (142)(3), (134)(2), (143)(2), (234)(1), (243)(1)\}$.
- $C(4) = \{(1234), (1243), (1324), (1342), (1423), (1432)\}$.

Une relation binaire R sur C assure que pour tous deux objets x et y dans C tels que xRy , $\text{ind}(x)=\text{ind}(y)$. Si C_1 et C_2 sont deux I -collections, une fonction $\phi : C_1 \rightarrow C_2$ est un *morphisme de I -collections* si l'index

de chaque élément dans C_1 correspond à l'index de son image par ϕ dans C_2 . La notation $C_1 \cong C_2$ indique l'existence d'un isomorphisme entre C_1 et C_2 . De plus, si pour chaque $i \in I$, $C_1(i) \subseteq C_2(i)$, alors C_1 est une *sous-collection* de C_2 . Pour tout $i \in I$, chaque $C(i)$ peut être considéré comme une sous-collection de C qui se compose de tous ses i -objets.

1.3.2 Collections graduées

Une *collection graduée* est une \mathbb{N} -collection. Dans une telle collection, étant donné un objet x de la collection C , sa taille, notée $|x|$, correspond à l'entier $\text{ind}(x)$. Cette relation entre un objet et sa taille est capturée par la fonction $|\cdot| : C \rightarrow \mathbb{N}$. Considérons C comme une collection combinatoire graduée. La série génératrice de C est $\mathbb{G}_C(t) = \sum_{n \in \mathbb{N}} |C(n)|t^n$, où la notation $|S|$ signifie la cardinalité d'un ensemble S . Cette série représente la suite d'entiers liée à C , spécifiquement la série $(\#C(n))_{n \in \mathbb{N}}$.

Considérons la classification des permutations de $n = 4$ selon la longueur du plus long cycle dans leur décomposition cyclique que nous l'avons fait avant, d'où les ensembles $C(1)$, $C(2)$, $C(3)$ et $C(4)$ contiennent respectivement les permutations dont le plus long cycle est de longueur 1, 2, 3 et 4. Soit x la permutation $(12)(3)(4)$. Puisque le plus long cycle de x est 2, alors $x \in C(2)$, et $\text{ind}(x) = |x| = 2$.

La série génératrice $\mathbb{G}_C(t)$ pour cette collection est définie comme suit :

$$\mathbb{G}_C(t) = \sum_{n \in \mathbb{N}} |C(n)|t^n.$$

Pour cet exemple, la série génératrice s'exprime par :

$$\mathbb{G}_C(t) = |C(1)|t^1 + |C(2)|t^2 + |C(3)|t^3 + |C(4)|t^4,$$

et en substituant les cardinalités, on obtient

$$\mathbb{G}_C(t) = t + 9t^2 + 8t^3 + 6t^4.$$

Une observation intéressante est que deux collections combinatoires graduées C_1 et C_2 sont équivalentes (c'est-à-dire, $C_1 \equiv C_2$) si et seulement si leurs séries génératrices sont égales. Voici quelques autres notions autour des collections graduées

- une collection C est dite *connexe* si son élément de taille zéro est unique.

- C est *augmentée* si elle ne contient pas d'élément de taille zéro.
- La collection représentée par $\{\epsilon\}$, où ϵ est un objet de taille zéro, est connue sous le nom de *collection unité*. Il est à noter que cette collection est intrinsèquement connexe, et toute collection C est connexe s'il existe un unique morphisme $\phi : \epsilon \rightarrow C$ de collections graduées.

1.3.3 Opérations sur les collections graduées

Les collections graduées, comme nous l'avons discuté, sont un concept fondamental en combinatoire. Cependant, pour vraiment exploiter leur puissance, nous devons les considérer de paire avec les opérations qui peuvent être effectuées sur elles.

- Somme : soient C_1 et C_2 deux collections graduées. La somme $C_1 + C_2$ est définie par

$$(C_1 + C_2)(n) := C_1(n) \sqcup C_2(n).$$

Ainsi, un objet de taille n de $C_1 + C_2$ est soit un objet de taille n de C_1 , soit un objet de taille n de C_2 .

- Produit : soient C_1 et C_2 deux collections graduées. Le produit $C_1 \times C_2$ est défini par :

$$(C_1 \times C_2)(n) := \{(x_1, x_2) : x_1 \in C_1, x_2 \in C_2, \text{ et } |x_1| + |x_2| = n\}.$$

- Suspension : soit C une collection graduée. Étant donné un entier k , la suspension par k d'une collection C est la collection graduée, notée $\text{Sus}_k(C)$, définie pour tout $n \in \mathbb{N}$ par

$$(\text{Sus}_k(C))(n) = \begin{cases} C(n-k) & \text{si } n-k \geq 0, \\ \emptyset & \text{sinon.} \end{cases}$$

- Augmentation : soit C une collection graduée. L'augmentation de C , notée $\text{Aug}(C)$ est définie par $\text{Sus}_1(\text{Sus}_{-1}(C))$. Cela représente la sous-collection de C sans son objet de taille 0. En d'autres termes, $\text{Aug}(C) = C \setminus C(0)$.
- Composition : pour tout entier non négatif k et collections graduées C_1, \dots, C_k , la composition homogène de C avec C_1, \dots, C_k est notée $C \circ [C_1, \dots, C_k]$. Pour chaque $n \in \mathbb{N}$, cette composition est définie comme suit :

$$(C \circ [C_1, \dots, C_k])(n) = \bigsqcup_{x \in C(k)} \left\{ (x, (y_1, \dots, y_k)) : y_i \in C_i, 1 \leq i \leq k, \text{ et } \sum_{i=1}^k |y_i| = n \right\}.$$

Cette définition implique que chaque objet de taille n dans $C \circ [C_1, \dots, C_k]$ est une paire ordonnée $(x, (y_1, \dots, y_k))$, où x est un objet dans C de taille k , et (y_1, \dots, y_k) est un tuple tel que chaque y_i appartient à C_i et que la taille combinée de ces objets y_i est n .

De plus, la composition de deux collections C_1 et C_2 est notée $C_1 \circ C_2$. Pour chaque $n \in \mathbb{N}$, cette composition est exprimée par

$$(C_1 \circ C_2)(n) = \bigsqcup_{k \in \mathbb{N}} C_1(k) \circ [C_2, \dots, C_2]$$

où dans le membre droit de l'égalité, il y a k occurrence de termes C_2 .

- **Liste** : soit C une collection graduée. L'opération de k -liste, notée $T_k(C)$, est définie par

$$(T_k(C))(n) := \{(x_1, \dots, x_k) : x_1, \dots, x_k \in C, \text{ et } |x_1| + \dots + |x_k| = n\}.$$

Cela signifie que tout objet de taille n dans $T_k(C)$ est représenté par un tuple (x_1, \dots, x_k) où $\sum_{i \in [k]} |x_i| = n$. L'opération de liste sur C est définie par

$$T(C) := \bigsqcup_{k \in \mathbb{N}} T_k(C).$$

1.3.4 Séries génératrices sur les collections graduées

Les collections graduées, comme nous l'avons discuté, sont fondamentales en combinatoire. Comprendre les opérations réalisées sur elles se trouvent à être bénéfique, car ces opérations conduisent à de nouvelles collections graduées, qui à leur tour donnent lieu à de nouvelles séries génératrices.

- **Somme** : si C_1 et C_2 sont combinatoires, la somme $C_1 + C_2$ est une collection combinatoire. Sa série génératrice est donnée par

$$\mathbb{G}_{C_1+C_2} = \mathbb{G}_{C_1} + \mathbb{G}_{C_2}.$$

- **Produit** : pour deux collections combinatoires C_1 et C_2 , le produit $C_1 \times C_2$ reste combinatoire. Sa série génératrice est donnée par

$$\mathbb{G}_{C_1 \times C_2} = \mathbb{G}_{C_1} \times \mathbb{G}_{C_2}.$$

- **Composition** : si C_1 et C_2 sont combinatoires, et C_2 est augmentée, la composition $C_1 \circ C_2$ reste combinatoire. La série génératrice pour cette composition est

$$G_{C_1 \circ C_2}(t) = G_{C_1}(G_{C_2}(t)).$$

- **Opérations de liste** : si C est combinatoire, $T_k(C)$ reste combinatoire, et sa série génératrice vérifie

$$\mathbb{G}_{T_k(C)}(t) = \mathbb{G}_C(t)^k.$$

De plus, si C est combinatoire et augmentée, $T(C)$ est combinatoire et sa série génératrice vérifie

$$\mathbb{G}_{T(C)}(t) = \frac{1}{1 - \mathbb{G}_C(t)}.$$

1.4 Arbres et structures similaires

L'arbre est un concept fondamental en combinatoire, dont la définition formelle peut varier considérablement selon le contexte et les propriétés recherchées. Voici quelques formulations variées pour définir un arbre, adaptées à diverses applications en combinatoire.

1.4.1 Arbres enracinés

Cette définition, telle qu'elle est donnée dans le livre de (Even, 2011), se présente comme suit : Considérons $T = (V, E)$, un graphe. T est qualifié d'arbre si et seulement si T est à la fois acyclique et connecté. Notre étude se portera principalement sur les arbres enracinés. Un arbre enraciné est défini par l'existence d'un sommet particulier, nommé racine. Un arbre enraciné est un arbre avec un point de départ spécifique (la racine) qui peut être tracé sur une surface plane sans que ses branches ne se superposent, excepté aux points de connexion. L'ordre des branches émanant d'un nœud donné est crucial et dépend de leur position dans le dessin.

Dans l'étude des graphes, en particulier des arbres, certains théorèmes fondamentaux mettent en lumière des propriétés uniques et définissantes de ces structures. Ces théorèmes offrent une perspective plus profonde sur la nature et les caractéristiques des arbres en combinatoire et en théorie des graphes. Voici deux théorèmes clés qui illustrent ces propriétés :

Théorème 1.6 *Les trois affirmations suivantes sont équivalentes pour un graphe G :*

1. G est un arbre.

2. G est acyclique ; cependant, l'ajout d'une arête supplémentaire à G crée un cycle.
3. G est connexe ; cependant, la suppression d'une arête de G le rend déconnecté.

Ce théorème met en évidence l'équilibre délicat entre acyclicité et connectivité qui définit un arbre.

Théorème 1.7 Pour un graphe fini $G(V, E)$ avec $n = |V|$, les affirmations suivantes sont équivalentes :

1. G est un arbre.
2. G est acyclique et possède $n - 1$ arêtes.
3. G est connexe et possède $n - 1$ arêtes.

Ce théorème relie directement la structure d'un arbre à son nombre d'arêtes, soulignant une propriété fondamentale des arbres en tant que graphes acycliques connectés.

Définition récursive des arbres

Après avoir exploré la première définition et ces propriétés fondamentales des arbres, nous nous penchons maintenant sur une approche plus spécifique et détaillée : la définition récursive des arbres. Cette perspective offre une compréhension approfondie de leur structure interne et de la manière dont chaque composant individuel contribue à l'intégrité globale de l'arbre. Cette définition est plus détaillée dans (Knuth, 2011).

Définition 1.8 Un arbre est caractérisé comme une collection finie T de nœuds ayant les propriétés suivantes :

1. un nœud unique, désigné comme la racine, noté $\text{racine}(T)$, est distinctement identifié au sein de T .
2. À l'exception de la racine, les autres nœuds de T peuvent être partitionnés en m (où $m \geq 0$) sous-ensembles disjoints T_1, \dots, T_m , chacun constituant un arbre. Ces arbres T_1, \dots, T_m sont appelés les sous-arbres de la racine.

Cette approche récursive permet une compréhension profonde de la structure et des propriétés des arbres, essentielle dans des domaines tels que les structures de données et la conception d'algorithmes. Un arbre constitué d'un seul nœud est considéré comme étant uniquement sa racine. Ainsi, chaque nœud dans un arbre fonctionne comme la racine d'un sous-arbre spécifique, formant une structure cohérente. Le nombre de sous-arbres que chaque nœud porte est connu sous le nom d'*arité* ou *degré* du nœud (le même mot peut avoir des significations différentes dans d'autres contextes), un indicateur clé de sa complexité structurelle.

1.4.2 Arbres ordonnés

Un arbre, lorsqu'il est ordonné, constitue une structure fondamentale en combinatoire. Considérons un arbre ordonné noté t . Les nœuds peuvent être catégorisés en deux types :

- **Nœuds internes** : Nœuds ayant un ou plusieurs enfants. Chaque nœud interne peut également être associé à un élément d'un ensemble prédéfini, lui attribuant ainsi une étiquette.
- **Feuilles** : Nœuds sans enfants.

Pour un nœud interne, disons x , dans l'arbre t , ses enfants sont ordonnés de gauche à droite. Si l'arité de x est ℓ , alors les enfants sont indexés de 1 à ℓ . Le i^e sous-arbre de t prend racine chez le i^e enfant de x . De même, les feuilles de l'arbre sont également ordonnées de gauche à droite, indexées de 1 jusqu'au nombre total de feuilles.

Dans les représentations graphiques, l'arbre est dessiné de sorte que la racine est le nœud le plus haut.

1.4.3 Collection des arbres enracinés

Un arbre enraciné (AE) est une collection structurée qui peut être décrite en utilisant une approche récursive. La base de cette structure est l'objet atomique, noté \bullet , qui représente un nœud. Un arbre enraciné peut être décrit comme un appariement ordonné d'un nœud avec un tuple d'autres arbres enracinés. Formellement, la collection des arbres enracinés, représentée par AE, suit la relation :

$$AE = \{\bullet\} \times T(AE)$$

où $T(AE)$ désigne la collection des listes d'arbres enracinés.

Selon cette définition, un arbre enraciné t peut s'exprimer comme

$$t = (\bullet, (t_1, \dots, t_k))$$

où chaque t_i est lui-même un arbre enraciné. Le cas spécial où $t = (\bullet, ())$ est appelé la feuille, notée \perp . La feuille est unique en ce qu'elle a une taille de 1.

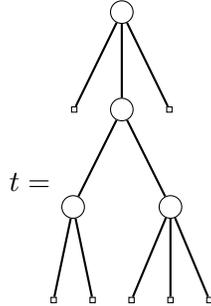
La représentation visuelle d'un arbre enraciné implique de dessiner chaque feuille comme un nœud unique et les autres arbres comme des nœuds reliés à leurs sous-arbres respectifs. L'ordre de connexion suit la séquence dans leur tuple, de gauche à droite.

La taille d'un arbre enraciné, donnée par $|t|$, est déterminée par le nombre d'occurrences de nœuds qu'il contient. Cela peut s'exprimer comme

$$|t| = 1 + \sum_{i \in [k]} |t_i|$$

où k est l'arité de la racine de t .

Par exemple :



on a

$$|t| = 11.$$

La série génératrice de AE, notée $\mathbb{G}_{\text{AE}}(t)$, peut être déduite de sa définition et est donnée par

$$\mathbb{G}_{\text{AE}}(t) = \mathbb{G}_{\bullet}(t) \times \mathbb{G}_{T(\text{AE})}(t).$$

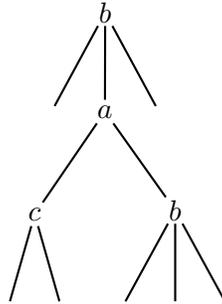
Cela conduit à l'équation quadratique :

$$t - \mathbb{G}_{\text{AE}}(t) + \mathbb{G}_{\text{AE}}(t)^2 = 0.$$

1.4.4 Arbres syntaxiques

Étant donné un ensemble gradué G , noté $G = \sqcup_{n \geq 1} G(n)$, l'arité d'un élément x dans G est n si x appartient à $G(n)$. Un arbre syntaxique sur G , ou un arbre G -syntaxique, est un arbre où chaque nœud interne d'arité n est étiqueté avec un élément de G ayant la même arité. Le degré d'un tel arbre est déterminé par le nombre de ses nœuds internes, tandis que son arité est déterminée par le nombre de ses feuilles.

Par exemple, si G est défini comme l'union de $G(2) = \{a, c\}$ et $G(3) = \{b\}$, un arbre syntaxique t peut être:



où le degré de t est 4 et son arité est 7.

Nous désignons par AE^G la collection graduée de tous les arbres syntaxiques G , où la taille d'un arbre G -syntaxique t est la taille de son arbre enraciné dans AE en enlevant l'étiquette de tous les nœuds. De même, nous désignons par AE_{\perp}^G la collection graduée de tous les arbres syntaxiques G , où la taille d'un arbre G -syntaxique t est la taille de son arité.

Un arbre syntaxique s peut être considéré comme un sous-arbre d'un autre arbre syntaxique t si s peut être superposé sur t à une position spécifique, chevauchant potentiellement les feuilles de s avec les nœuds internes de t . Si c'est le cas, on dit que t a une occurrence de s . Inversement, si t n'a aucune occurrence de s , on dit qu'il évite s .

1.4.5 Arbres binaires

Un arbre binaire est une structure d'arbre courante. Dans un tel arbre, chaque nœud peut posséder jusqu'à deux sous-arbres. Si un nœud a un sous-arbre, il est catégorisé comme sous-arbre gauche ou droit. Pour le définir plus précisément, un arbre binaire peut être vu comme une collection finie de nœuds qui est soit vide, soit composée d'une racine accompagnée de deux arbres binaires distincts, appelés sous-arbres gauches et droits de cette racine. Il convient de noter que bien qu'un arbre binaire partage

des similitudes avec les arbres généraux, il représente une structure distincte en soi.

Soit AB la collection combinatoire graduée vérifiant la relation

$$AB = \perp + [\{\bullet\}, [AB, AB]].$$

Si t est un arbre binaire qui n'est pas une feuille, alors $t = (\bullet, (t_1, t_2))$ où t_1 et t_2 sont deux arbres binaires.

Dans ce cas, t_1 est un sous-arbre gauche de t , et t_2 est un sous-arbre droit de t .

Cette définition est intrinsèquement récursive. Par exemple, les structures :

$$\perp, (\bullet, (\perp, \perp)), (\bullet, ((\bullet, (\perp, \perp)), \perp)), \dots$$

sont toutes des arbres binaires valides. La taille d'un arbre binaire, notée $|t|$, est définie récursivement de la manière suivante.

- Si $t = \perp$, alors $|t| = 1$.
- Si $t = (\bullet, (t_1, t_2))$, alors $|t| = |t_1| + |t_2|$.

Ainsi, par sa définition, nous pouvons déduire que sa série génératrice est la suivante :

$$\mathbb{G}_{AB}(t) = \mathbb{G}_{(\perp)}(t) + \mathbb{G}_{AB}^2(t),$$

$$\mathbb{G}_{AB}(t) = t + \mathbb{G}_{AB}^2(t).$$

Ce qui conduit à l'équation quadratique

$$t - \mathbb{G}_{AB}(t) + \mathbb{G}_{AB}^2(t) = 0.$$

La solution de l'équation ci-dessus est

$$\mathbb{G}_{AB}(t) = \frac{1 - \sqrt{1 - 4t}}{2t} = \sum_{n \in \mathbb{N}} \frac{1}{n+1} \binom{2n}{n} t^n$$

dont les premiers termes sont

$$1, 1, 2, 5, 14, 42, 132, 429, 1430.$$

Ces nombres célèbres en combinatoire sont appelés les nombres de Catalan. Ils portent le nom d'Eugène Charles Catalan, un mathématicien du 19^e siècle. Cependant, Euler et Segner les avaient déjà mentionnés dans leurs travaux au 18^e siècle. Dans le livre (Stanley, 1999b), Stanley énumère plusieurs différentes applications de ces nombres en combinatoire, démontrant leur importance dans ce domaine.

1.4.6 Arbres comme arbres binaires

Nous passons maintenant de l'étude des arbres binaires aux arbres généraux. Il est essentiel de souligner les distinctions fondamentales entre ces deux structures :

1. un arbre est toujours enraciné, ce qui implique qu'il n'est jamais vide. Chaque nœud dans un arbre peut posséder un nombre arbitraire d'enfants non négatifs.
2. En revanche, un arbre binaire peut être vide. Les nœuds dans un arbre binaire peuvent avoir 0, 1 ou 2 enfants. De plus, comme expliqué précédemment, lorsque deux enfants sont présents, l'un est désigné comme l'*enfant de gauche* et l'autre comme l'*enfant de droite*.

La série génératrice des arbres enracinés (AE) suit la même relation algébrique que celle de la collection graduée d'arbres binaires, où la taille d'un arbre binaire est déterminée par son nombre de feuilles. Cela implique que AE et AB , lorsqu'ils sont considérés comme des collections graduées, sont isomorphes.

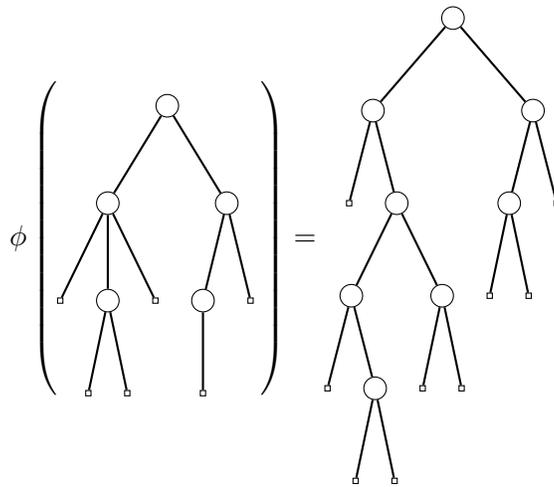
Il existe un isomorphisme explicite entre AE et AB , appelée correspondance de rotation, et attribué à Knuth (Knuth, 2011).

Pour illustrer cette relation dans nos notations, considérons l'application $\phi : AE \rightarrow AB$ définie de manière récursive pour tout arbre enraciné planaire t comme suit :

$$\phi(t) = \begin{cases} \perp \in AB & \text{si } t = \perp, \\ (\bullet, (\phi(t_1), \phi((\bullet, (t_2, \dots, t_k)))))) & \text{sinon, pour } t = (\bullet, (t_1, t_2, \dots, t_k)) \text{ avec } k \geq 1. \end{cases}$$

Les collections graduées AE et AB sont isomorphes. L'application ϕ , telle que définie ci-dessus, établit un isomorphisme entre ces deux collections.

Par exemple,



1.4.7 Partition des arbres binaires

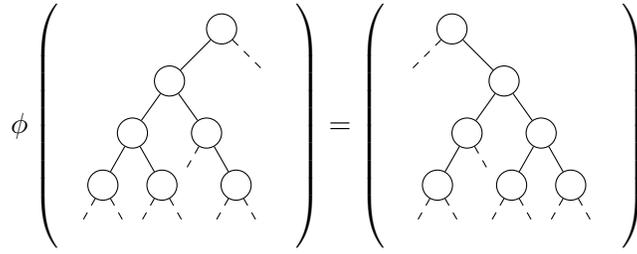
Dans le contexte des arbres binaires, il est essentiel d'introduire des sous-ensembles spécifiques qui, ensemble, partitionnent l'ensemble AB . Cette partition est formulée comme suit :

$$AB = \underset{\swarrow}{\mathbb{B}} \sqcup \underset{\sim}{\mathbb{B}} \sqcup \underset{\searrow}{\mathbb{B}}$$

où chaque sous-ensemble est caractérisé par la configuration des enfants de la racine comme suit :

- $\underset{\swarrow}{\mathbb{B}}$ représente l'ensemble des arbres binaires dont la racine possède à la fois un enfant gauche et un enfant droit.
- $\underset{\sim}{\mathbb{B}}$ désigne l'ensemble des arbres binaires où la racine a seulement un enfant gauche.
- $\underset{\searrow}{\mathbb{B}}$ concerne l'ensemble des arbres binaires où la racine a seulement un enfant droit.

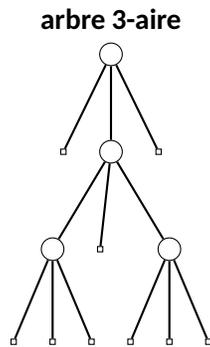
L'ensemble $\underset{\swarrow}{\mathbb{B}}, \underset{\sim}{\mathbb{B}}, \underset{\searrow}{\mathbb{B}}$ est une partition de AB . En outre, il existe un isomorphisme $\phi : \underset{\swarrow}{\mathbb{B}} \rightarrow \underset{\searrow}{\mathbb{B}}$, qui établit un isomorphisme en envoyant chaque arbre d'un ensemble sur son image miroir dans l'autre. Cette relation bijective souligne la symétrie inhérente dans la structure des arbres binaires.



La plupart de nos discussions se feront sur \mathbb{B} , ce qui veut dire qu'elles sont les mêmes sur \mathbb{B} .

1.4.8 Arbres k -aires

Étant donné un nombre naturel $k \geq 1$, un *arbre k -aires* est défini comme un arbre enraciné où chaque nœud interne possède une arité de k . Par exemple, pour $k = 3$, on a



La collection des arbres k -aires, notée $\text{Ary}^{(k)}$, peut être exprimée de manière récursive comme suit :

$$\text{Ary}^{(k)} = \{\perp\} + \{\bullet\} \times \text{Ary}^{(k)k}$$

où \perp et \bullet sont des entités atomiques.

Les collections graduées $\text{Ary}_{\perp}^{(k)}$ et $\text{Ary}_{\bullet}^{(k)}$ de tous les arbres k -aires, où la taille d'un arbre dans $\text{Ary}_{\perp}^{(k)}$ correspond à son arité et la taille d'un arbre dans $\text{Ary}_{\bullet}^{(k)}$ à son degré, sont combinatoires.

De plus, la série génératrice de $\text{Ary}_{\bullet}^{(k)}$ suit l'équation algébrique

$$1 - G_{\text{Ary}_{\bullet}^{(k)}}(t) + tG_{\text{Ary}_{\bullet}^{(k)}}(t)^k = 0.$$

Un résultat connu indique que le nombre d'éléments dans $\text{Ary}_{\bullet}^{(k)}$ de taille n est donné par

$$\#\text{Ary}_{\bullet}^{(k)}(n) = \frac{1}{(k-1)n+1} \binom{kn}{n}.$$

Ces nombres sont connus sous le nom de nombre de Fuss-Catalan.

Dans le contexte des arbres binaires, un arbre binaire est un arbre 2-aire.

De plus, comme l'arité est déterminée par le nombre de nœuds internes, nous pouvons voir que

$$\#Ary_{\perp}^{(k)}(i) = \begin{cases} \#Ary_{\bullet}^{(k)}\left(\frac{i-1}{k-1}\right) & \text{si } i = k + s(k-1) \text{ où } s \in \mathbb{N}, \\ 0 & \text{sinon} \end{cases}$$

pour tout $k \geq 2$.

Ce sont également des nombres de Fuss-Catalan mais avec des zéros dans la série. Par exemple, la série de nombres associée à $Ary_{\perp}^{(k)}$ commence par :

$$1, 0, 1, 0, 3, 0, 12, 0, 55, 0, 273 \quad \text{pour } k = 3,$$

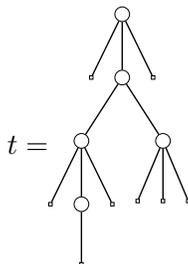
$$1, 0, 0, 1, 0, 0, 4, 0, 0, 22, 0, 0, 140, 0, 0, 969, 0, 0, 7084 \quad \text{pour } k = 4.$$

1.5 Langage des arbres

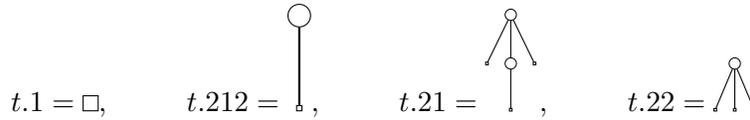
Dans l'étude des arbres enracinés, nous introduisons un langage spécialisé basé sur l'alphabet $\mathbb{N}_{\geq 1}$. Ce langage sert à identifier précisément les nœuds au sein de ces arbres (Giraud, 2017). Nous définissons à présent une action à droite partielle de monoïde sur $AE \times \mathbb{N}_{\geq 1}^*$. Cette action, également notée \cdot , est définie de manière récursive comme suit :

$$(\bullet, (t_1, \dots, t_k)) \cdot u = \begin{cases} (\bullet, (t_1, \dots, t_k)) & \text{si } u = \epsilon, \\ t_i \cdot v & \text{sinon (où } u = iv \text{ et } i \in \mathbb{N}_{\geq 1}), \end{cases} \quad (1.1)$$

pour tout $(\bullet, (t_1, \dots, t_k)) \in AE$ et $u \in \mathbb{N}_{\geq 1}^*$. Il est important de noter que cette action est partielle, car chaque t_i n'est défini que si i ne dépasse pas l'arité de la racine de t . Le langage de l'arbre $N(t)$ d'un arbre t est alors défini comme l'ensemble fini de tous les mots u dans $\mathbb{N}_{\geq 1}$ tels que l'action $t \cdot u$ aboutit à un arbre enraciné planaire bien défini. *Par exemple:*



On a



Soit t un arbre enraciné. Nous définissons les termes associés comme suit :

- **Nœuds** : Chaque mot de $N(t)$ représente un nœud de t . Nous l'appellerons adresse du nœud ainsi représenté
- **Nœuds internes** : Un nœud u de t est un nœud interne s'il vérifie la condition qu'il existe un $i \in \mathbb{N}_{\geq 1}$ tel que ui est également un nœud de t . L'ensemble de tous les nœuds internes est noté $N_{\bullet}(t)$.
- **Feuilles** : Un nœud u de t est une feuille s'il n'est pas un nœud interne. L'ensemble de toutes les feuilles est noté $N_{\perp}(t)$.
- **Racine** : La racine de t a ε pour adresse, qui peut être soit un nœud interne, soit une feuille.
- **Degré et arité** : Le degré de t , noté $\deg(t)$, est le nombre total de nœuds internes, c'est-à-dire $\#N_{\bullet}(t)$. L'arité de t , notée $\text{ari}(t)$, est le nombre total de feuilles, c'est-à-dire $\#N_{\perp}(t)$.
- **Ancêtres et enfants** : Un nœud u est un ancêtre d'un nœud v si $u \neq v$ et $u \preceq_{\text{pref}} v$. Un nœud v est le i^{e} enfant de u si $v = ui$ pour $i \in \mathbb{N}_{\geq 1}$, faisant de u le père unique de v . L'arité d'un nœud est le nombre d'enfants qu'il a.
- **Nœuds Frères** : Deux nœuds v et v' sont frères s'ils partagent un parent commun u et sont des enfants distincts, c'est-à-dire v est le i^{e} enfant et v' est le i'^{e} enfant de u pour $i \neq i'$.
- **Ordre et Chemin** : Les nœuds sont ordonnés en utilisant l'ordre lexicographique de leurs adresses. Un chemin dans t est une suite de nœuds (u_1, \dots, u_k) telle que chaque nœud u_j est le père de u_{j+1} . Un chemin est maximal s'il commence par la racine et se termine par une feuille.
- **Hauteur** : La hauteur de t , $\text{ht}(t)$, est définie comme la longueur maximale de ses chemins maximaux moins 1. Cela correspond également à la longueur du mot le plus long de $N(t)$.

CHAPITRE 2

COMBINATOIRE ALGÈBRIQUE

La combinatoire algébrique est aujourd'hui un domaine essentiel des mathématiques qui combine l'algèbre et la combinatoire. Ce domaine nous donne accès à des outils algébriques puissants, qui améliorent notre compréhension de divers sujets. Elle nous permet par exemple d'exprimer les nombreux résultats d'une opération par une somme. Par exemple, le produit pré-Lie sur les arbres ou produit de mélange sur les mots donne une somme formelle d'objets. Ce chapitre traite des espaces vectoriels obtenus à partir de collections graduées.

2.1 Espaces polynomiaux gradués

Lorsque nous parlons d'une série (resp. d'un polynôme) associée à une collection C , nous faisons référence à une sommation formelle (ou dans le cas des polynômes, une sommation formelle finie) d'objets au sein de C , chacun étant pondéré par des coefficients d'un corps K . Pour le contexte de notre discussion, le corps K est supposé de caractéristique 0.

2.1.1 Séries et polynômes sur collections

Définition 2.1 Soit C une I -collection.

1. Une **série** sur C , où C -série, est une application $f : C \rightarrow K$.
2. Le **coefficient** d'un élément x dans la série f est noté $\langle x, f \rangle$.
3. Le **support** d'une série f est défini comme $\text{Supp}(f) = \{x \in C : \langle x, f \rangle \neq 0\}$, où 0 est le zéro de K .
4. Un **polynôme** sur C , ou C -polynôme, est une C -série qui a un support fini.
5. Une C -série f est **monômiale** si son support est un singleton.
6. Une C -série f est **homogène** s'il existe un indice $i \in I$ tel que tous les éléments de son support appartiennent à $C(i)$.

7. Pour tout sous-ensemble X de C , la **série caractéristique** de X est la C -série vérifiant, pour tout $x \in C$,

$$\langle x, ch(X) \rangle = \begin{cases} 1 & \text{si } x \in X, \\ 0 & \text{sinon.} \end{cases}$$

2.1.2 Espaces polynomiaux

Soit $K\langle C \rangle$ l'ensemble des C -polynômes, appelé espace des C -polynômes. L'espace $K\langle C \rangle$ est équipé de deux opérations principales :

1. **Addition** $+$: $K\langle C \rangle \times K\langle C \rangle \rightarrow K\langle C \rangle$. Pour deux éléments $f_1, f_2 \in K\langle C \rangle$ et un élément $x \in C$, l'addition est définie par

$$\langle x, f_1 + f_2 \rangle = \langle x, f_1 \rangle + \langle x, f_2 \rangle.$$

2. **Multiplication scalaire** \cdot : $K \times K\langle C \rangle \rightarrow K\langle C \rangle$. Pour tout élément $f \in K\langle C \rangle$, scalaire $\lambda \in K$, et $x \in C$, la multiplication de f par le scalaire λ est définie par

$$\langle x, \lambda f \rangle = \lambda \langle x, f \rangle.$$

Avec ces opérations, $K\langle C \rangle$ forme un espace vectoriel sur K . De plus, $K\langle C \rangle$ peut être exprimé comme une somme directe

$$K\langle C \rangle = \bigoplus_{i \in I} K\langle C(i) \rangle.$$

Chaque sous-espace $K\langle C(i) \rangle$ représente la i^e composante homogène de $K\langle C(i) \rangle$.

2.1.2.1 Morphisme d'espace polynomial

Considérons deux espaces polynomiaux, $K\langle C_1 \rangle$ et $K\langle C_2 \rangle$, où C_1 et C_2 sont des I -collections. Une fonction

$$\varphi : K\langle C_1 \rangle \rightarrow K\langle C_2 \rangle$$

est dite être un *morphisme d'espace polynomial* lorsqu'elle vérifie la condition que pour chaque i dans I et chaque x de $C_1(i)$, nous avons $\varphi(x)$ appartenant à $K\langle C_2(i) \rangle$. Il est important de noter qu'un morphisme de collections combinatoires, noté

$$\psi : C_1 \rightarrow C_2$$

peut être étendu linéairement pour produire un morphisme d'espace polynomial

$$\varphi : K\langle C_1 \rangle \rightarrow K\langle C_2 \rangle.$$

Notons également que $K\langle C_2 \rangle$ est un *sous-espace* de $K\langle C_1 \rangle$ s'il existe un morphisme d'espace polynomial injectif de $K\langle C_2 \rangle$ à $K\langle C_1 \rangle$.

2.1.2.2 Espaces polynomiaux combinatoires gradués

Lorsque nous avons une collection combinatoire graduée C , l'espace polynomial $K\langle C \rangle$ peut être exprimé comme une somme directe sur les nombres naturels

$$K\langle C \rangle = \bigoplus_{n \in \mathbb{N}} K\langle C \rangle(n). \quad (2.1)$$

Cette décomposition signifie que l'espace polynomial est construit à partir des composantes graduées de la collection C .

2.1.3 Série de Hilbert

lorsque C est combinatoire, chaque composante $K\langle C(n) \rangle$ pour $n \in \mathbb{N}$ a une dimension finie. La série de Hilbert pour $K\langle C \rangle$ est alors définie par

$$H_{K\langle C \rangle}(t) = \sum_{n \in \mathbb{N}} \dim K\langle C \rangle(n) t^n. \quad (2.2)$$

Il convient de noter que la série de Hilbert $H_{K\langle C \rangle}(t)$ et la série génératrice $\mathbb{G}_C(t)$ pour C représentent la même série de puissance.

CHAPITRE 3

OPÉRADES

Ce chapitre est dédié à comprendre les opérades, des structures algébriques qui jouent un rôle crucial dans l'articulation des opérations et de leurs compositions. Les opérades permettent de modéliser des structures complexes à partir d'éléments plus simples et de comprendre comment ces éléments interagissent entre eux. Nous explorerons divers aspects de cette théorie. Ce chapitre s'appuiera sur des travaux fondamentaux dans ce domaine, tel que ceux présentés dans les références (Chapoton, 2008), (Foissy, 2017b), (Jean-Louis Loday, 2012). L'objectif est de fournir une compréhension des opérades et de leur puissance en tant qu'outil mathématique, tout en préparant le terrain pour l'application spécifique dans le chapitre suivant.

3.1 Opérades non-symétriques

Définition 3.1 Une **opérade non-symétrique** \mathcal{O} est une structure mathématique définie comme suit :

1. elle est composée d'une collection \mathcal{O} d'ensembles (ou d'espaces vectoriels, ou d'autres objets dans une autre catégorie appropriée) indexée sur \mathbb{N} (ou $\mathbb{N} \setminus \{0\}$). On note $\mathcal{O}(n)$ l'ensemble d'indice n de \mathcal{O} .

2. Il existe une application

$$\begin{aligned} \gamma : \mathcal{O}(n) \times \mathcal{O}(k_1) \times \cdots \times \mathcal{O}(k_n) &\rightarrow \mathcal{O}(k_1 + \cdots + k_n) \\ (o, o_1, \dots, o_k) &\mapsto o \gamma(o_1, \dots, o_k). \end{aligned}$$

3. Il existe un élément distingué $1 \in \mathcal{O}(1)$, appelé **unité**.

Les axiomes d'une opérade non-symétrique sont les suivants :

1. **Associativité** : pour tous éléments $o \in \mathcal{O}(n)$, $o_i \in \mathcal{O}(k_i)$ (pour $1 \leq i \leq n$) et $p_{ij} \in \mathcal{O}(l_{ij})$ (pour $1 \leq j \leq k_i$), l'équation suivante est vérifiée :

$$\begin{aligned} &o \gamma(o_1 \gamma(p_{11}, \dots, p_{1k_1}), \dots, o_n \gamma(p_{n1}, \dots, p_{nk_n})) \\ &= (o \gamma(o_1, \dots, o_n)) \gamma(p_{11}, \dots, p_{1k_1}, \dots, p_{n1}, \dots, p_{nk_n}) \end{aligned}$$

2. **Unitalité** : pour tout $o \in \mathcal{O}(n)$, la composition avec l'élément identité donne :

$$\gamma(o, 1, \dots, 1) = o$$

où l'identité est utilisée n fois, et

$$1 \gamma(o) = o.$$

Cette définition fournit un cadre pour comprendre la façon dont les opérades non-symétriques structurent les opérations et les compositions dans divers domaines mathématiques.

3.1.1 Opérades dans la catégorie des espaces vectoriels

Bien que de nombreuses opérades conventionnelles soient initialement définies dans la catégorie des ensembles, elles peuvent être étendues à la catégorie des espaces vectoriels tout en préservant leurs propriétés essentielles. Les considérations clés pour cette extension sont les suivantes :

- remplacer les ensembles $\mathcal{O}(n)$ par des espaces vectoriels $K\langle\mathcal{O}(n)\rangle$.
- Remplacer le produit cartésien par le produit tensoriel des espaces vectoriels.
- Remplacer l'union disjointe par la somme directe des espaces vectoriels.
- Étendre par linéarité les applications ensemblistes.

Pour décrire cette transition d'une opérade non-symétrique dans la catégorie des ensembles en une opérade dans la catégorie espace vectoriel, un outil utile est le foncteur espace vectoriel libre sur un ensemble. Cette approche nous permet de maintenir la nature combinatoire de l'opérade, à condition que chaque espace vectoriel $\mathcal{O}(n)$ soit équipé d'une base, et que les applications de composition au sein de ces bases soient rigoureusement définies et décrites.

3.1.2 Composition partielle

Nous introduisons une perspective alternative sur les opérades non-symétriques, s'éloignant de l'utilisation conventionnelle des applications γ . Dans cette approche, les applications \circ_i sont centrales pour définir la composition. L'application \circ_i vérifie

$$\begin{aligned}\circ_i : \mathcal{O}(n) \times \mathcal{O}(m) &\rightarrow \mathcal{O}(n + m - 1) \\ (x, y) &\mapsto x \circ_i y.\end{aligned}$$

Les compositions γ sont généralement associées à la composition de plusieurs éléments à la fois. En revanche, les compositions partielles \circ_i se concentrent sur une composition plus directe, en mettant l'accent sur l'insertion d'un seul objet dans un autre. Cette approche remplace les compositions γ par une suite de compositions \circ_i pour toutes les valeurs possibles de m, n . Ces compositions doivent adhérer à des axiomes spécifiques :

Soient x, y, z dans $\mathcal{O}(m), \mathcal{O}(n), \mathcal{O}(p)$ respectivement.

1. D'abord, pour tout x dans $\mathcal{O}(m), i \geq m + 1$, et y dans \mathcal{O} :

$$x \circ_i y = 0$$

2. **Règle de composition disjointe** : Pour tous x, y, z et $i < j$ dans $[m]$, la composition vérifie $(x \circ_i y) \circ_{j+n-1} z = (x \circ_j z) \circ_i y$.
3. **Règle de composition imbriquée** : La composition vérifie $(x \circ_i y) \circ_{j+i-1} z = x \circ_i (y \circ_j z)$ pour tout x, y, z , et $i \in [m]$, et $j \in [n]$.
4. **Principe d'unité** : L'unité vérifie $1 \circ_1 x = x = x \circ_i 1$ pour toutes les valeurs possibles de x et i .

Les compositions \circ_i peuvent être exprimées en termes de compositions γ et vice versa, offrant un pont entre les deux perspectives. La connexion entre la présentation traditionnelle et la présentation alternative est établie en utilisant l'unité par

$$x \circ_i y = x \gamma(\underbrace{1, \dots, 1}_{i-1}, y, \underbrace{1, \dots, 1}_{|x|-i}).$$

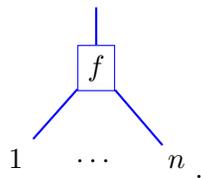
Une opérade non-symétrique, comme nous l'avons dit précédemment, peut être considérée comme un espace polynomial gradué augmenté, noté $\mathbb{K}\langle C \rangle$. L'application \circ_i sera présentée par

$$\circ_i : \mathbb{K}\langle C \rangle(n) \otimes \mathbb{K}\langle C \rangle(m) \rightarrow \mathbb{K}\langle C \rangle(n + m - 1).$$

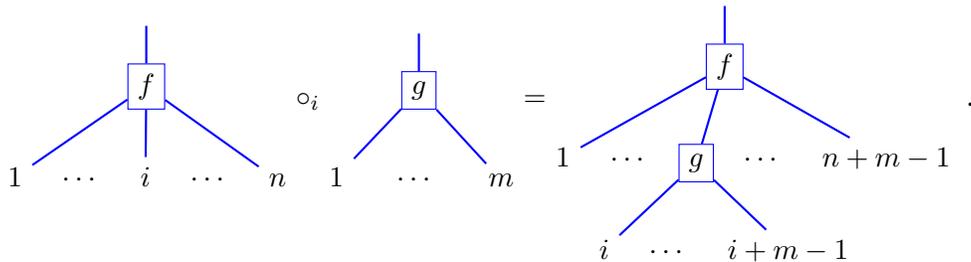
Pour offrir une compréhension plus intuitive, considérons chaque élément f de $\mathbb{K}\langle C \rangle(n)$ comme un produit d'arité n . Les opérations \circ_i fonctionnent en fusionnant ces produits. Plus précisément, pour tout f dans $\mathbb{K}\langle C \rangle(n)$, g dans $\mathbb{K}\langle C \rangle(m)$, et i dans $[n]$, le résultat $f \circ_i g$ est obtenu en connectant la sortie de g à la i^{me} entrée de f .

Cette structure joue un rôle crucial dans divers contextes mathématiques et informatiques, nous permettant de modéliser et de manipuler des opérations de manière non-symétrique. Les opérades non-symétriques trouvent des applications en topologie algébrique, en algèbre homologique et en combinatoire, entre autres. Elles fournissent un cadre polyvalent pour décrire des opérations où l'ordre des opérandes importe et sont un concept fondamental dans l'étude des structures algébriques. Illustrons le concept de composition partielle sous une forme visuelle.

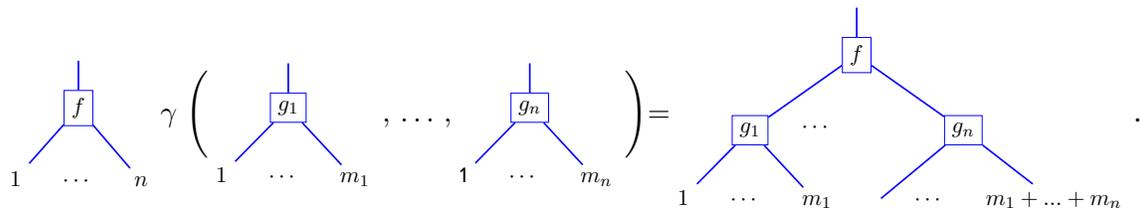
Tout élément $f \in \mathbb{K}\langle C \rangle(n)$ est représenté comme une opération à n entrées et une sortie :



La composition partielle peut être visualisée par



Une composition complète peut être visualisée comme suit :



Sous-opérades Soit $K\langle C \rangle$ une opérade. Une sous-opérade $K\langle C' \rangle$ de $K\langle C \rangle$ est un sous-espace polynomial gradué augmenté. Elle contient l'unité de $K\langle C \rangle$ et est stable sous l'opération γ .

3.1.3 Opérades en tant qu'espèces de structures

Une espèce opéradique P , telle que définie dans (Foissy, 2017a), est une espèce avec des compositions $\circ_a : P(A) \times P(B) \longrightarrow P(A \sqcup B \setminus \{a\})$ définies pour tout ensembles finis non vides A et B tels que $a \in A$.

Les compositions d'une espèce opéradique sont associatives dans le sens que :

- pour tous ensembles A, B, C finis, $a, b \in A$ distincts, $p \in P(A)$, $q \in P(B)$ et $r \in P(C)$

$$(p \circ_a q) \circ_b r = (p \circ_b r) \circ_a q.$$

- Pour tous ensembles A, B, C finis, $a \in A$, $b \in B$, $p \in P(A)$, $q \in P(B)$ et $r \in P(C)$

$$(p \circ_a q) \circ_b r = p \circ_a (q \circ_b r).$$

De plus, pour chaque singleton $\{a\}$ il existe un élément identité $I_a \in P(\{a\})$ tel que pour tout $p \in P(A)$, $p \circ_a I_a = p$ et $I_a \circ_a p = p$.

Par exemple, soit As l'espèce telle que pour tout ensemble fini non vide A , $As(A)$ est l'ensemble des mots possibles formés à partir des lettres de A , où chaque lettre apparaît exactement une fois. L'espèce opéradique sur As est telle que \circ_a est définie comme suit : pour $a_1 \cdots a_n \in As(A)$ et $a \in A$ tel que $a_i = a$ pour un certain $i \in [n]$ et pour un autre mot $w \in As$,

$$a_1 \cdots a_n \circ_a w = a_1 \cdots a_{i-1} w a_{i+1} \cdots a_n.$$

L'identité I_a est simplement le mot a de longueur 1.

À partir d'une espèce opéradique P , on peut déduire une opérade, également notée par P , structurée de la façon suivante :

- pour tout $n \geq 1$, l'ensemble $P(n)$ est défini comme étant l'ensemble $P([n])$.
- La composition partielle des éléments $p \in P(n)$ et $q \in P(m)$ pour $i \in [n]$ est

$$p \circ_i q = P(\sigma_{n,m}^{(i)})(p \circ_i q),$$

où $\sigma_{n,m}^{(i)} : ([n] \setminus \{i\}) \sqcup [m] \longrightarrow [n + m - 1]$ est la bijection définie par

- $\sigma_{n,m}^{(i)}(k) = k$ si $k \in [n]$ et $k \leq i - 1$,
- $\sigma_{n,m}^{(i)}(\dot{k}) = k + i - 1$ si $\dot{k} \in [m]$,
- $\sigma_{n,m}^{(i)}(k) = k + m - 1$ si $k \in [n]$ et $k \geq i$

où les éléments de $[n]$ sont notés par $1, \dots, n$ et les éléments de $[m]$ sont notés par $\dot{1}, \dots, \dot{m}$.

- L'unité de P est $I_1 \in P(\{1\})$.

De plus, une opérade peut être vue comme un monoïde d'espèces positives avec l'opération associative de substitution et ayant l'espèce X comme identité. La structure fondamentale des opérades symétriques, sur lesquelles nous ne travaillerons pas dans ce mémoire, est basée sur les espèces. Les opérades symétriques présentent une structure quelque peu plus complexe par rapport aux opérades non-symétriques. Pour une exploration approfondie des opérades en tant que monoïdes, y compris des exemples illustratifs sur lesquels se basent nos exemples, le lecteur est invité à consulter l'ouvrage complet de Méndez (Méndez, 2015), (Loday et Vallette, 2012) ou bien (Giraud, 2015) pour une approche différente.

3.2 Exemples d'opérades

Les opérades apparaissent dans divers contextes algébriques et combinatoires, chacune capturant différentes structures de composition. Nous explorons ici quelques exemples classiques pour mieux comprendre leur diversité et leurs applications.

3.2.1 Opérade des endomorphismes

Soit A un ensemble. L'opérade d'endomorphisme sur A , notée End_A , est définie comme suit :

- $\text{End}_A(n) = \{A^n \rightarrow A\}$
- La composition partielle de $f \in \text{End}_A(n)$ et de $g \in \text{End}_A(m)$ à la position i vérifie

$$(f \circ_i g)(a_1, \dots, a_{n+m-1}) = f(a_1, \dots, g(a_i, \dots, a_{i+n-1}), \dots, a_{n+m-1})$$

pour tous $a_1, \dots, a_{n+m-1} \in A$.

- L'unité est la fonction d'identité dans $\text{End}_A(1)$.

Cette opérade joue un rôle important en algèbre et en topologie, notamment dans l'étude des structures algébriques. Elle illustre comment les concepts de composition et de fonctionnalité peuvent être encadrés dans le langage des opérades, offrant un moyen puissant de décomposer et d'analyser des opérations complexes.

3.2.2 Opérades libres

Soit G une collection graduée d'ensembles. L'opérade libre sur G , notée Free_G , est définie comme suit :

- $\text{Free}_G(n)$ est l'ensemble des arbres G -syntaxiques d'arité n .
- La composition partielle $s \circ_i t$ consiste à greffer la racine de t sur la i^{e} feuille de s .
- L'unité \perp est l'arbre constitué d'une seule feuille.

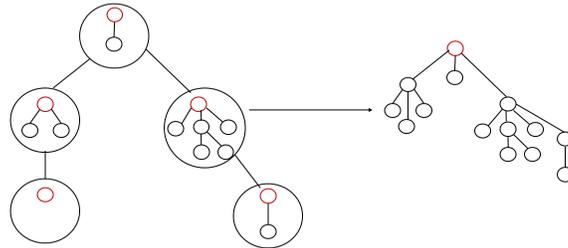
Cette construction met en évidence la nature algébrique des arbres syntaxiques et leur utilisation dans l'élaboration d'opérades. L'opérade libre fournit un cadre pour étudier la composition et la structure des arbres dans divers domaines mathématiques, y compris la théorie des catégories, l'algèbre homologique et la combinatoire. Elle illustre également comment des opérations complexes peuvent être construites de manière itérative à partir de composants de base.

3.2.3 Opérade NAP

L'opérade NAP (abréviation de *non-associative permutative*) a été introduite dans (Livernet, 2006) et se base sur l'espèce des arbres enracinés, notée NAE . Cette espèce NAE est munie d'un produit $\mu(\pi, t, a) = t_V$, où t_V représente un arbre formé en reliant les racines des arbres au sein d'un assemblage $a = \{t_B\}_{B \in \pi}$. Cette liaison se produit lorsque les blocs correspondants dans π sont interconnectés au sein d'un arbre externe. La racine de l'arbre nouvellement formé t_V est déterminée pour être la racine de t_{B_0} , B_0 étant le bloc racine de l'arbre externe (π, t) .

L'opérade NAP capture l'essence de la composition non-associative et permutative des arbres enracinés, offrant une approche pour étudier les propriétés combinatoires et structurelles de ces arbres. Elle joue un

rôle significatif dans la compréhension des opérations algébriques et topologiques dans diverses structures arborescentes, notamment dans l'établissement de certaines propriétés de l'opérade pre-Lie.



Le diagramme ci-dessus illustre la manière dont l'opérade NAP fonctionne. Il démontre la fusion des racines des arbres individuels pour former un nouvel arbre composite, en respectant la structure et l'ordre imposé par l'opérade.

3.3 Présentation par générateurs et relations

Considérons une opérade non-symétrique $K\langle C \rangle$. Une présentation de cette opérade peut être décrite par un couple (G, R) , où G est une collection graduée augmentée et R est un sous-espace de Free_G . Cela conduit à l'isomorphisme

$$K\langle C \rangle \cong \frac{\text{Free}_G}{\langle R \rangle}. \quad (3.1)$$

Ici, $\langle R \rangle$ désigne l'idéal opéradique non-symétrique engendré par R dans Free_G . Dans ce contexte, G représente l'ensemble des générateurs, et R , l'espace des relations de $K\langle C \rangle$.

Une présentation (G, R) est dite *quadratique* si R est un sous-espace homogène de Free_G , composé uniquement d'arbres syntaxiques de degré 2. De plus, (G, R) est *binaire* si tous les éléments de G sont d'arité 2. Par conséquent, $K\langle C \rangle$ est également classifiée comme quadratique ou binaire en fonction de son type de présentation.

Le premier exemple à considérer est celui de l'opérade associative (Ass) en tant qu'opérade non-symétrique. Cette opérade est définie comme suit. L'ensemble $\text{Ass}(n)$ est le singleton $\{M_n\}$ et l'unité M_1 . La composition

est définie par

$$M_m \circ_i M_n = M_{m+n-1}.$$

L'opérade Ass est caractérisée par un générateur et une relation, étant engendrée par M_2 avec une relation singulière exprimée comme

$$M_2 \circ_1 M_2 = M_2 \circ_2 M_2.$$

3.4 Algèbres sur opérades

Étant donnée une opérade non-symétrique $K\langle C \rangle$, elle peut décrire une classe d'algèbres polynomiales graduées, notées comme algèbres sur $K\langle C \rangle$ ou brièvement, $K\langle C \rangle$ -algèbres. Une $K\langle C \rangle$ -algèbre peut être définie comme un espace polynomial gradué $K\langle D \rangle$, équipé d'une action linéaire à gauche :

$$* : K\langle C \rangle(n) \otimes K\langle D \rangle^{\otimes n} \rightarrow K\langle D \rangle.$$

Pour tout élément f dans $K\langle C \rangle$ avec une arité n , il sert de fonction linéaire :

$$f : K\langle D \rangle^{\otimes n} \rightarrow K\langle D \rangle.$$

Cette action respecte les relations imposées par la structure d'opérade non-symétrique de $K\langle C \rangle$. Cette relation s'exprime comme

$$(f \circ_i g) * (a_1 \otimes \cdots \otimes a_{n+m-1}) = f * (a_1 \otimes \cdots \otimes a_{i-1} \otimes g * (a_i \otimes \cdots \otimes a_{i+m-1}) \otimes a_{i+m} \otimes \cdots \otimes a_{n+m-1})$$

pour tout $f \in K\langle C \rangle(n)$, $g \in K\langle C \rangle(m)$, $i \in [n]$ et $a_1 \otimes \cdots \otimes a_{n+m-1} \in K\langle D^{\otimes n+m-1} \rangle$.

Il résulte de cette relation que si G est un ensemble générateur de $K\langle C \rangle$, alors spécifier l'action de chaque x dans G sur $K\langle D \rangle^{\otimes |x|}$ est suffisant pour déterminer $*$.

3.5 Morphismes d'opérades

Les morphismes d'opérades relient différentes structures opéradiques, permettant le transfert et la comparaison de propriétés entre elles. Soit \mathcal{O} et \mathcal{O}' deux opérades. Une application

$$\phi : \mathcal{O} \longrightarrow \mathcal{O}'$$

est un morphisme d'opérades si

- $\phi(1_{\mathcal{O}}) = 1_{\mathcal{O}'}$
- Pour tout x dans \mathcal{O} , $|\phi(x)| = |x|$
- Pour tout x, y dans \mathcal{O} , $\phi(x \circ_i y) = \phi(x) \circ_i \phi(y)$

Dans ce chapitre, nous avons introduit plusieurs concepts de la théorie des opérades. Ces principes constituent la base des méthodes utilisées dans le chapitre 4, où nous explorons un exemple spécifique de ces opérades, l'opérade dupliciale. Comprendre la définition des opérades présentée dans la section 3.1, ainsi que les compositions partielles dans la section 3.1.2 et la présentation par générateurs et relations dans la section 3.3, est crucial pour saisir le contenu du chapitre 4. De plus, une compréhension approfondie l'opérade libre dans la section 3.2.2 peut aider le lecteur à mieux apprécier les preuves des résultats présentés dans le chapitre suivant.

CHAPITRE 4

OPÉRADE DUPLICIALE ET SOUS-OPÉRADES

L'opérade dupliciale est une structure algébrique qui joue un rôle dans la compréhension des opérations binaires et de leurs propriétés associatives. L'opérade dupliciale, définie sur les arbres binaires, se base sur une opération particulière de greffe. Ce processus consiste à insérer un nouvel arbre binaire dans un nœud d'un arbre déjà existant, ce qui permet de construire des structures d'arbres plus complexes.

Alessandra Frabetti a utilisé le concept d'opérade dupliciale dans ses recherches sur la renormalisation (Frabetti, 2008), spécifiquement dans le contexte des arbres binaires. En physique mathématique, la renormalisation est un ensemble de techniques utilisées pour traiter des quantités infinies qui apparaissent dans la formulation de théories physiques, en particulier dans la théorie quantique des champs.

Dans ce mémoire, nous explorons l'opérade dupliciale à travers ses sous-opérades et cherchons à comprendre les propriétés uniques des arbres binaires qu'elle contient. Notre objectif est de comprendre la riche structure algébrique ainsi développée sur les arbres binaires.

4.1 Opérade dupliciale

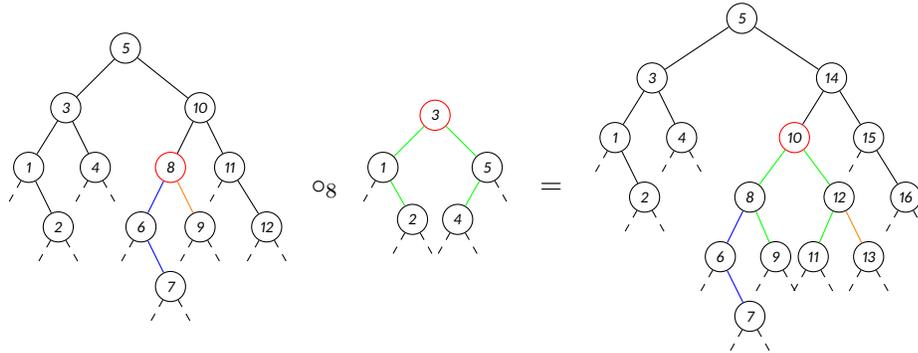
Définition 4.1 *L'opérade dupliciale, également appelée opérade OverUnder et notée Dup , est définie par*

$$Dup = K\langle Aug(AB) \rangle.$$

Où

- *l'ensemble $Dup(n)$ est l'ensemble des arbres binaires planaires avec n nœuds internes.*
- *Pour tous $s \in Dup(n)$, $t \in Dup(m)$ et $i \in [n]$, la composition partielle $w = s \circ_i t$ est définie de la manière suivante. L'arbre binaire w est l'arbre binaire planaire résultant du remplacement du i^e nœud de s par l'arbre t . Le sous-arbre gauche du i^e nœud de s sera greffé sur la feuille la plus à gauche de t dans le nouvel arbre, tandis que le sous-arbre droit du i^e nœud de s sera greffé sur la feuille la plus*

à droite de t . Exemple :



- L'unité est l'arbre binaire constitué d'un seul nœud.

L'opérade Dup est une opérade ensembliste et combinatoire. Sa série de Hilbert est donnée par :

$$H_{Dup}(t) = \frac{1 - \sqrt{1 - 4t}}{2t} = \sum_{n \in \mathbb{N}_{\geq 1}} \binom{2n}{n} \frac{1}{n+1} t^n \quad (4.1)$$

Elle admet la présentation (G, R) où G est l'ensemble de générateurs

$$G = \left(\begin{array}{c} \text{[Diagram 1]} \\ \text{[Diagram 2]} \end{array} \right)$$

et R est un espace de relations :

- $c(\ll) \circ_1 c(\ll) - c(\ll) \circ_2 c(\ll)$
- $c(\gg) \circ_1 c(\ll) - c(\ll) \circ_2 c(\gg)$
- $c(\gg) \circ_1 c(\gg) - c(\gg) \circ_2 c(\gg)$

où \ll (resp. \gg) désigne le premier (resp. le second) arbre de G , et Pour tout $x \in G$, $c(x)$ est l'arbre G -syntaxique étiquetée par x ayant exactement un nœud interne avec $|x|$ feuilles comme enfants.

Les générateurs et les relations sont définis de manière à capturer la structure dupliciale, avec des opérations binaires spécifiques et des conditions d'associativité. En particulier, la 1^{re} (resp. la 3^e) relation dit que \ll (resp. \gg) est associative, et la 2^e est une associativité relative entre \ll et \gg .

Les algèbres sur Dup , connues sous le nom d'algèbres dupliciales, sont des espaces dotés de deux produits binaires qui satisfont certaines conditions d'associativité dérivées des relations dans R . Ces algèbres peuvent être exprimées comme :

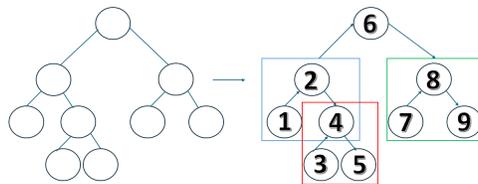
$$(x \ll y) \gg z = x \ll (y \gg z) \quad \text{pour tout } x, y, z \in D \quad (4.2)$$

où \ll et \gg sont les opérations binaires et associative de l'algèbre.

4.1.1 Indexation des nœuds : ordre infixé

Il existe de nombreuses manières d'indexer les nœuds d'un arbre binaire. Toutes peuvent être utilisées dans ce chapitre. La méthode que nous utiliserons est appelée l'ordre infixé.

Définition 4.2 L'ordre infixé induit par un arbre t est un ordre total sur l'ensemble de ses nœuds internes. Cet ordre est défini de manière récursive. Il suppose que tous les nœuds internes du sous-arbre $t.1$ (le sous-arbre gauche) sont considérés comme inférieurs à la racine de t . En même temps, la racine de t est considérée comme inférieure à tous les nœuds internes du sous-arbre $t.2$ (le sous-arbre droit). La convention est d'indexer ainsi les nœuds de 1 à n où n est le nombre de nœuds de t . Par exemple, voici un arbre binaire et ses nœuds ainsi indexés selon l'ordre infixé



De manière récurrente, les nœuds du rectangle rouge sont plus grands que les nœuds du rectangle bleu. Les nœuds du rectangle vert sont plus grands que les nœuds des rectangles bleu et rouge.

4.1.2 Langage des arbres binaires et opérade dupliciale

Notons que dans Dup , l'indexation se fait sur les nœuds et non sur les feuilles. Cela pourrait créer une certaine confusion par rapport à ce que nous avons vu jusqu'à présent, car nous représentons habituellement les entrées d'un élément par les feuilles d'un arbre. Par conséquent, dans cette section, nous allons redéfinir le langage utilisé pour éviter toute confusion.

Considérons $t \in AB$. Comme défini précédemment, le langage de l'arbre t , noté $N(t)$, est constitué de mots d'entiers positifs appelés adresses. Dans les arbres binaires, les adresses dans $N(t)$ sont composées exclusivement de 1 et de 2. De plus, le nombre et la disposition des feuilles dans un arbre binaire sont fixes. Par conséquent, pour tout $N_\bullet(t)$ donné, il existe un unique $N_\perp(t)$ correspondant qui ferait de t un arbre binaire. Ainsi, un arbre binaire peut être défini de manière unique par l'ensemble $N_\bullet(t)$. Dans notre contexte des arbres binaires, nous noterons simplement $N(t)$ au lieu de $N_\bullet(t)$.

Nous utiliserons également la notation $n_n(t)$ pour désigner le nombre de nœuds (qui ne sont pas des feuilles) dans un arbre t , et $n_l(t)$ pour désigner le nombre de feuilles dans t .

4.1.2.1 Branches dans les arbres binaires

Définition 4.3 La **branche gauche** d'un arbre binaire t est la sous-arborescence de t dont les adresses sont composées de toutes les adresses de t constituées uniquement de 1, y compris la racine de t . De manière similaire, la **branche droite** est définie par des adresses composées uniquement de 2.

4.1.2.2 Premier et dernier nœud

Le *premier nœud* de t est identifié par l'adresse avec la plus longue suite de 1. Si k représente le nombre de 1 dans cette adresse, nous désignons cette adresse par 1^k . Le *dernier nœud* de t est le nœud avec la plus longue suite de 2.

4.1.2.3 Langage des arbres et opérade dupliciale

Soit t un arbre et soit x une adresse. Nous définissons $x.N(t)$ par

$$x.N(t) = \{x.i : i \in N(t)\}.$$

Soit t_1, t', t des arbres binaires dans Dup , tels que $t \circ_i t_1 = t'$. Soit x l'adresse du nœud i dans t , 1^k l'adresse du premier nœud et 2^j celle du dernier nœud dans t_1 . L'ensemble $N(t')$ contient exactement

- tous les adresses de $N(t)$ qui ne commencent pas par x .
- Toutes les adresses dans $x.N(t_1)$, ce qui inclut également l'adresse x .
- Tous les adresses de $N(t)$ qui commencent par x , avec la modification suivante :
 - toute adresse de la forme $x.q$ sera changée en $x.1^k.q$ si q commence par 1,
 - toute adresse de la forme $x.q$ sera changée en $x.2^j.q$ si q commence par 2.

Exemple : soit t l'arbre binaire tel que

$$N(t) = \{\epsilon, 1, 11, 112, 121, 1212, 122, 1221, 2, 21, 212, 22\}.$$

Soit t_1 l'arbre binaire tel que

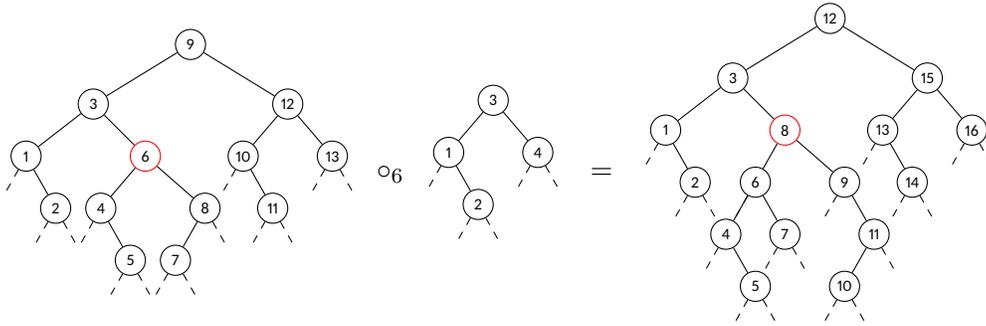
$$N(t_1) = \{\epsilon, 1, 12, 2\}$$

Soit $t' = t \circ_6 t_1$ (Les arbres binaires de t, t_1 et t' sont illustrés ci-dessous). Il est évident d'après $N(t_1)$ que l'adresse du premier nœud est 1, tandis que celle du dernier nœud est 2. L'adresse du nœud d'indice 6 par l'ordre infixe est 12. Ainsi, $N(t')$ sera formé de

- toutes les adresses dans $N(t)$ qui ne commencent pas par 12, qui sont $\epsilon, 1, 11, 112, 122, 2, 21, 22, 212$.
- Toutes les adresses dans $12.N(t_1)$ qui sont $12.\epsilon, 12.1, 12.2, 12.12$ équivalent à $12, 121, 122, 1212$.
- Les adresses 121, 1212 deviennent maintenant 12.1.1, 12.1.12 respectivement.
- Les adresses 122, 1221 deviennent maintenant 12.2.2, 12.2.21 respectivement.

Ainsi, $N(t') = \{\epsilon, 1, 11, 112, 12, 121, 1211, 12112, 1212, 122, 1222, 12221, 2, 21, 212, 22\}$ qui sont

exactement les adresses des arbres représentés par le graphique suivant :



Pour un arbre binaire t , et toute adresse $x \in N(t)$, le nombre de nœuds $n_n(t \cdot x)$ est le nombre d'adresses dans $N(t)$ qui commencent par x . Cela fournit un moyen de quantifier la structure de l'arbre par rapport à un nœud donné.

Considérons un autre arbre binaire t' . Si t et t' sont identiques, alors pour chaque adresse $x \in N(t)$, le nombre de nœuds dans la sous-arborescence de la racine x dans les deux arbres est égal. Formellement, cela peut être exprimé comme suit :

$$t = t' \implies n_n(t \cdot x) = n_n(t' \cdot x) \text{ pour tout } x \in N(t).$$

Cette propriété souligne la cohérence du décompte des nœuds basé sur les adresses à travers des arbres binaires identiques. Cette propriété est bien sûr évidente mais sa contraposée est un outil puissant qui sera utilisé à de nombreuses reprises pour démontrer que deux arbres ne sont pas identiques.

4.1.3 Proposition sur les opérades dupliciales

Dans cette section, nous introduisons une proposition fondamentale qui est essentielle pour les développements qui suivront.

Proposition 4.4 *Considérons des arbres binaires t, t_1, t_2 dans Dup tels que $t_1, t_2 \in \mathbb{B}$ et $n_n(t) = n$. Soient i, j des entiers avec $1 \leq i, j \leq n$. Alors,*

$$t \circ_i t_1 = t \circ_j t_2 \implies i = j \text{ et } t_1 = t_2.$$

Démonstration. Il est immédiatement évident que pour que $t \circ_i t_1 = t \circ_j t_2$ soit vrai, t_1 et t_2 doivent avoir

le même nombre de nœuds. Par conséquent, dans notre démonstration, nous supposons que t_1 et t_2 ont le même nombre de nœuds.

La démonstration est divisée en cas démontrant que pour tout arbre binaire t avec une sous-arborescence gauche ou droite non vide, des nœuds différents i et j aboutiront à des arbres distincts. Nous noterons par (i) l'adresse de i , et par (j) , l'adresse de j . D'abord, considérons que le nœud i est la racine de t . Comme les branches gauche et droite de t contiennent au moins un nœud (puisque $t \in \mathbb{B}$), alors $n_n((t \circ_i t_1).1) > n_n(t.1)$ et $n_n((t \circ_i t_1).2) > n_n(t.2)$. Si $j < i$ (selon l'ordre infixe), alors $n_n((t \circ_j t_2).1) > n_n(t.1)$ mais $n_n((t \circ_j t_2).2) = n_n(t.2)$, ce qui indique que les arbres ne peuvent pas être les mêmes, impliquant $j \neq i$. De même, j ne peut pas être plus grand que i , menant à $j = i$.

Deuxièmement, supposons que les nœuds i et j sont dans des sous-arbres différents de t . Supposons aussi, sans perte de généralité, que i est dans le sous-arbre gauche, et que j est dans le sous-arbre droit de t . Cela entraîne $n_n((t \circ_i t_1).1) > n_n(t.1)$. De plus, $n_n((t \circ_j t_2).1) = n_n(t.1)$ puisque j est dans le sous-arbre droit de t et n'affectera pas le nombre de nœuds de $(t \circ_j t_2).1$, i et j ne peuvent pas être dans des sous-arbres différents.

Ensuite, supposons que i et j sont dans le même sous-arbre de t mais qu'aucun de ces deux nœuds n'est descendant de l'autre. Supposant que i et j sont dans le sous-arbre gauche de t , ils doivent être descendants d'un troisième nœud x dont l'adresse sera notée (x) . Si i est dans le sous-arbre gauche de x , et j est dans le droit, alors $n_n((t \circ_i t_1).(x).1) > n_n((t \circ_j t_1).(x).1)$, ce qui implique que $t \circ_i t_1 \neq t \circ_j t_2$.

Enfin, supposons que i est un descendant de j . Si i appartient au sous-arbre gauche de j , alors $n_n((t \circ_i t_1).(j).2) = n_n(t.(j).2)$ mais $n_n((t \circ_j t_2).(j).2) > n_n(t.(j).2)$ (car $t_1, t_2 \in \mathbb{B}$), impliquant que les arbres sont différents.

Ayant établi que $i = j$, ceci mène au fait que t_1 doit être identique à t_2 puisque greffer deux arbres différents à la même position sur le même arbre donne toujours des résultats différents. \square

4.2 Une sous-opérade dupliciale

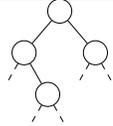
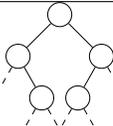
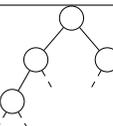
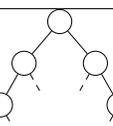
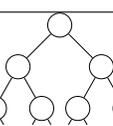
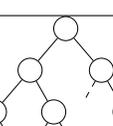
L'exploration des sous-opérades d'une opérade donnée permet d'affirmer la compréhension de la structure et en particulier d'identifier les sous-opérades qu'elle contient. Un exemple notable est l'étude

des sous-opérades dupliciales. Rappelons que l'opérade dupliciale, notée Dup , est engendrée par deux arbres binaires, sous certaines relations. Une question naturelle se pose alors : quelles sont les structures qui émergent lorsque nous modifions l'ensemble des générateurs ?

Définition 4.5 Soit G une collection d'arbres binaires. Nous définissons Dup_G comme l'opérade engendrée par les éléments de G ainsi que par les relations inhérentes à Dup . Il est évident que Dup_G constitue une sous-opérade de Dup , puisque Dup contient tous les arbres binaires.

En utilisant un algorithme, nous avons pu compter le nombre d'arbres distincts dans Dup_G pour différents ensembles G de générateurs. Les résultats dans le tableau ci-dessous ont été notre première inspiration pour notre travail suivant. Nous ne partagerons que ceux que nous avons pu expliquer, mais il y aura plus d'exemples dans la dernière section.

Tableau 4.1 Dimensions de sous-opérades Dup_G où G est un singleton. Ici 0^n est équivalent à la sous-suite $0, 0, \dots, 0$ avec 0 utilisé n fois. Les références OEIS concernent uniquement les sous-suites des nombres strictement positifs.

| G | Suite de dimensions | OEIS référence et formule |
|---|--|--|
|  | 1, 0, 0, 1, 0, 0, 4, 0, 0, 22, 0, 0, 140, 0, 0, 969... | $\frac{\binom{4n}{n}}{3n+1}$: A002293 |
|  | 1, 0 ³ , 1, 0 ³ , 5, 0 ³ , 35, 0 ³ , 285, 0 ³ , 2530... | $\frac{\binom{5n}{n}}{4n+1}$: A002294 |
|  | 1, 0, 0, 1, 0, 0, 4, 0, 0, 22, 0, 0, 140, 0, 0, 969... | $\frac{\binom{4n}{n}}{3n+1}$: A002293 |
|  | 1, 0 ³ , 1, 0 ³ , 5, 0 ³ , 35, 0 ³ , 285, 0 ³ , 2530... | $\frac{\binom{5n}{n}}{4n+1}$: A002294 |
|  | 1, 0 ⁵ , 1, 0 ⁵ , 7, 0 ⁵ , 70, 0 ⁵ , 819, 0 ⁵ , 10472, 0 ⁵ ... | $\frac{\binom{7n}{n}}{6n+1}$: A002296 |
|  | 1, 0 ⁵ , 1, 0 ⁵ , 7, 0 ⁵ , 70, 0 ⁵ , 819, 0 ⁵ , 10472, 0 ⁵ ... | $\frac{\binom{7n}{n}}{6n+1}$: A002296 |

Cette section se penche sur l'investigation des sous-opérades engendrées par un seul arbre binaire. Notre objectif principal est de présenter les propriétés structurelles de ces sous-opérades.

4.2.1 Sous-opérades engendrées par un peigne

Théorème 4.6 Soit G le singleton contenant t de degré n , où t est un peigne gauche ou peigne droit. Alors,

$$\mathbb{H}_{Dup_G}(t) = \sum_{m \in \mathbb{N}} i \times t^m$$

où

$$i = \begin{cases} 1 & \text{si } m = n + k(n - 1) \text{ pour } k \in \mathbb{N}, \\ 0 & \text{sinon.} \end{cases}$$

La démonstration est facilement vue du fait que si t est un peigne gauche, $t \circ_i t = t \circ_j t$ pour tout $1 \leq i, j \leq n$. Il y a donc soit un unique arbre d'un certain degré qui est le peigne gauche, soit aucun arbre du tout. Ceci peut être vu comme une série de k -Fuss-Catalan de $k = 1$ avec des zéros à l'intérieur.

4.2.2 Sous-opérades engendrées par un singleton

Théorème 4.7 Soit G le singleton contenant t de degré n , où $t \in \mathbb{B}$. Alors,

$$|Dup_G(i)| = \#Ary_{\perp}^{(n)}(i).$$

Pour démontrer ce théorème, nous introduirons un morphisme d'opérades

$$\phi : Free_G \longrightarrow Dup_G.$$

L'existence de ce morphisme découle du fait que l'opérade libre satisfait la propriété universelle des opérades. C'est-à-dire, dans ce cas précis, qu'il existe un unique morphisme d'opérade ϕ tel que le diagramme

$$\begin{array}{ccc} G & \longrightarrow & Dup_G \\ \downarrow & \nearrow & \\ Free_G & & \end{array}$$

commute.

Puisque G ne contient qu'un élément de degré n , $Free_G$ consistera en des arbres n -aires. Comme $Free_G(i)$ consiste en des arbres G -syntaxiques d'arité i et comme ils sont tous des arbres n -aires, on peut facilement voir par sa définition que $Free_G(i) = \#Ary_{\perp}^{(n)}(i)$. L'arbre avec 1 nœud et n feuilles sera le générateur de $Free_G$, que nous appellerons g .

Soit t l'unique élément de G et ϕ une fonction telle que

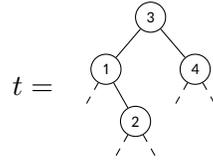
$$\phi : Free_G \longrightarrow Dup_G \tag{4.3}$$

$$g \mapsto \phi(g) = t. \tag{4.4}$$

Une autre façon de voir $\phi(x)$, est que si $x = (\dots((g \circ_{i_1} g) \circ_{i_2} g) \dots \circ_{i_m} g)$ alors $\phi(x) = (\dots((t \circ_{i_1} t) \circ_{i_2} t) \dots \circ_{i_m} t)$.

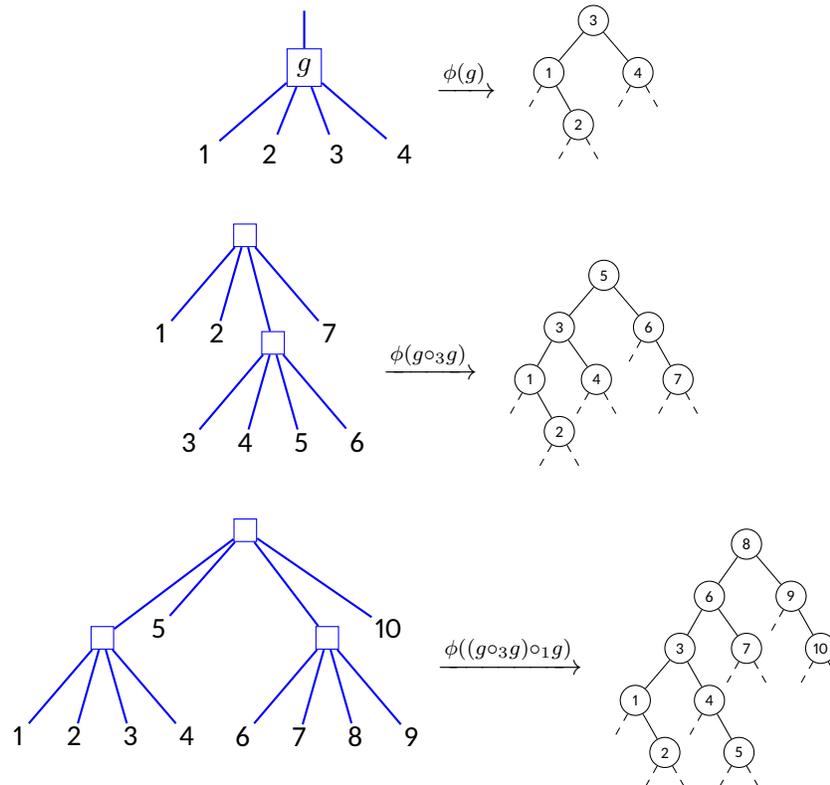
Visualisons cette fonction pour mieux la comprendre avant de démontrer qu'il s'agit d'une bijection.

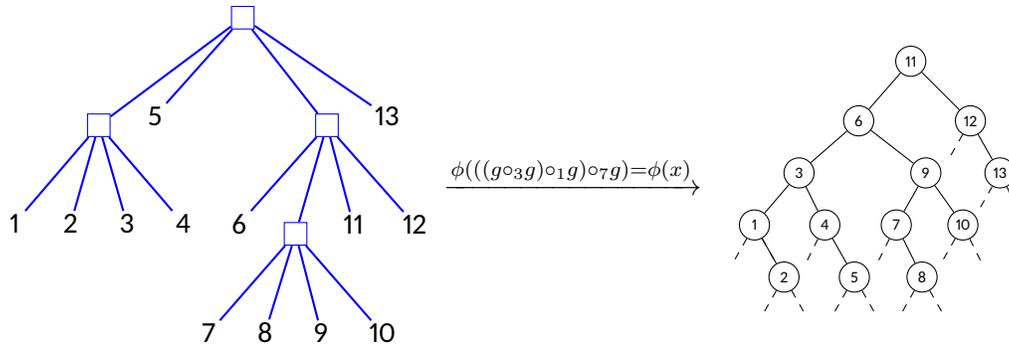
Prenons un exemple simple. Soit



Puisque $|t| = 4$, cela signifie que Free_G consisterait en des arbres 4-aires. Nous appellerons g l'arbre avec 1 racine et 4 feuilles, car il est le générateur de Free_G .

Considérons l'arbre x dans Free_G tel que $x = (((g \circ_3 g) \circ_1 g) \circ_7 g)$. Voici les étapes de $\phi(x)$:





Il est facile de voir que la fonction est surjective.

Soit t_1 un arbre dans Dup_G , cela signifie que t_1 a été construit par de multiples compositions partielles du générateur t , puisqu'il n'y a qu'un seul générateur. Supposons que $n_n(t) = n$. Soit $t_1 = (\dots ((t \circ_{i_1} t) \circ_{i_2} t) \dots \circ_{i_m} t)$. Puisque à chaque composition, la taille augmente de $n - 1$, en partant de n , cela signifie que $x = (\dots ((g \circ_{i_1} g) \circ_{i_2} g) \dots \circ_{i_m} g)$ existe dans $Free_G$ et $\phi(x) = t_1$ par définition.

4.2.2.1 Lemme préliminaire

Avant de commencer la démonstration de l'injectivité, nous établissons d'abord un lemme préliminaire.

Lemme 4.8 Soit G le singleton contenant t de degré n , où $t \in \mathbb{B}$. Considérons deux arbres x et y dans Dup_G tels que $x = t\gamma[t_1, \dots, t_n]$ et $y = t\gamma[t'_1, \dots, t'_n]$, et supposons $x = y$. Cette condition implique que $t_i = t'_i$ pour tout $i \in [n]$.

La démonstration sera faite par induction sur le nombre k d'arbres différents de l'unité (de l'opéradé dupliciale) dans x . (Le nombre de i tels que t_i n'est pas l'unité).

Démonstration. Cas de base . Pour $k = 1$:

supposons que pour $x = t\gamma[t_1, \dots, t_n]$, t_i n'est pas une unité. Si la composition de y se compose d'un seul arbre sur le nœud j^e , alors, on peut écrire $x = t \circ_i t_i$ et $y = t \circ_j t'_j$, et puisque $x = y$, par la proposition 4.4, $i = j$ et $t_i = t'_j$ ce qui implique que $t_i = t'_i$ pour tout $i \in [n]$.

Nous allons maintenant démontrer par contradiction que y ne peut pas être composé de plus d'un arbre.

- Premièrement, supposons que t'_{j_s} n'est pas une unité dans la composition de y pour tous les $s \in S$.

Comme $x = y$, alors $n_n(x) = n_n(y)$, $n_n(x.1) = n_n(y.1)$ et $n_n(x.2) = n_n(y.2)$. Si i est un nœud gauche de t , afin de garder le même nombre de nœuds dans le même arbre, j_s sera un nœud gauche pour tous les $s \in S$. (Idem pour le nœud droit).

- De plus, tout nœud j_s pour $s \in S$ est un descendant de i ou est égal à i car, s'il y a un j_s tel que ni i est un descendant de j_s ni vice versa, alors ils sont les descendants d'un troisième nœud h . Supposons i est sur la gauche de h . Ainsi, $n_n(t.(h).1)$ va augmenter de $n_n(t_i) - 1$ dans x , tandis que dans y , comme j_s n'est pas une unité, alors sa greffe sur le côté droit de h assurera que $y.(h).1$ est strictement inférieur à $x.(h).1$ (Idem si i est sur la droite). De plus, si i est un descendant de l'un des j_s , supposons de son côté droit, alors $n_n(t.(j_s).2)$ augmentera de $n_n(t_i) - 1$ dans x tandis que $n_n(t.(j_s).1)$ n'augmentera pas. Mais, dans y , comme $t'_{j_s} \in \mathbb{B}$, alors $n_n(t.(j_s).1)$ et $n_n(t.(j_s).2)$ augmenteront tous les deux d'une quantité strictement inférieure à $n_n(t_i) - 1$, ce qui conduira à une contradiction.
- Comme nous travaillons dans \mathbb{B} , la greffe de t_i sur i dans x créera de nouveaux nœuds sur ses branches gauche et droite, donc la seule façon de compenser est de choisir un s et un s' tels que j_s et j'_s soient tous les deux descendants de i (ou l'un est égal) et soient sur la branche gauche et droite de $t.(i)$. Ici, nous pouvons toujours choisir le nœud le plus haut sur la branche, et utiliser les axiomes des opérades pour ne travailler qu'avec lui car le reste peut être greffé dessus en créant un nouvel arbre. Donc maintenant, nous considérons seulement le cas de deux nœuds, l'un sur la gauche et l'autre sur la droite de i (ou l'un d'eux est égal à i).
- Supposons que l'un des deux nœuds dans y est égal à i et l'autre non, soit $n_n(t'_i.1) < n_n(t_i.1)$ ou $n_n(t'_i.2) < n_n(t_i.2)$. Notons par 1^{t_i} l'adresse du premier nœud de t_i . Supposons que $n_n(t'_i.1) < n_n(t_i.1)$. Cela impliquerait qu'il existe une adresse dans $x.i.1$ qui a été créée en greffant t_i dans x qui n'a pas été créée en greffant t'_i dans y . Ainsi, il doit exister un $s' \in S$ pour que cette adresse soit créée. Comme elle est dans $x.i.1$, alors j'_s doit être un descendant gauche de i . Supposons $(j'_s) = (i).1^{s'}$. En greffant $t'_{j'_s}$ sur t dans y , nous pouvons voir que $n_n(y.(j'_s)) = n_n(y.(i).1^{s'}) = n_n(t'_{j'_s})$ maintenant greffer t'_i n'affecte pas les nœuds créés par la greffe de $t'_{j'_s}$ mais changera leurs adresses, et ainsi la racine de $y.(j'_s)$ deviendra la racine de $y.i.1^{t_i}.1^{s'}$ et donc en les greffant, nous pouvons toujours conclure que $n_n(y.i.1^{t_i}.1^{s'}) = n_n(t'_{j'_s})$. Maintenant greffer t_i sur t dans x , $x.i.1^{t_i}.1^{s'}$ est un sous-arbre qui doit être compensé par $t'_{j'_s}$, car il ne peut pas être dans t'_i . Mais $t'_{j'_s} \in \mathbb{B}$, donc sa greffe crée un nœud gauche et un nœud droit. Comme $x.i.1^{t_i}.1^{s'}$ est dans le sous-arbre gauche de $t'_{j'_s}$ alors $n_n(x.i.1^{t_i}.1^{s'}) < n_n(t'_{j'_s})$. Ceci est une contradiction car $x = y$.
Donc, $n_n(t'_i.2) < n_n(t_i.2)$, mais avec le même raisonnement, nous pouvons démontrer que c'est

impossible. Donc j_s et j'_s ne peuvent pas être i . Avec le même raisonnement, il ne devrait pas y avoir de nœuds entre j_s et i ou entre j'_s et i car il y aurait une incompatibilité dans le nombre de nœuds sur l'adresse de la greffe de j_s ou de j'_s et donc i doit être le père de j_s et de j'_s .

- Supposons que j_s soit à sa gauche. Maintenant, nous savons que $n_n(t_{i.1}) + 1 = n_n(t'_{j_s})$, et puisque i doit être le père de j_s et de j'_s , nous pouvons voir que la structure de t'_{j_s} avec l'arbre $t.(j_s).1$ greffé à sa gauche et $t.(j_s).2$ greffé à sa droite devrait être la même que $t_{i.1}$ avec $t.(j_s)$ greffé à sa gauche pour. C'est impossible, car la seule façon d'augmenter le nombre de nœuds sur une branche pour qu'ils deviennent similaires est de greffer le générateur sur un nœud de la branche (ou la racine, mais cela a été traité) mais cela créera à son tour une adresse à l'intérieur car le générateur des deux arbres appartient à \mathbb{B} , et comme une adresse qui n'est pas sur la branche ne peut pas être compensée, les arbres ne seront pas les mêmes. Ainsi, nous ne pouvons pas avoir plus d'un t'_{j_s} et il doit être le même que t_i avec $j_s = i$.

Étape inductive : supposons que la propriété est vraie pour k arbres.

Supposons que $k + 1$ est le nombre d'arbres différents de l'unité dans x .

- Supposons maintenant que l soit la racine de t . S'il y a un $i > l$ tel que t_i n'est pas une unité, alors il doit y avoir un $i' > l$ tel que $t'_{i'}$ n'est pas une unité pour avoir le même nombre de nœuds de $x.2$ et $y.2$. Donc, si nous avons des arbres non unitaires greffés avant et après l sur t dans x , alors nous devons aussi avoir des arbres non unitaires greffés avant et après l sur t dans y . Cependant, la greffe d'arbres sur le côté droit de la racine n'affecte pas le côté gauche d'un arbre. Ainsi, nous voyons que si $x' = t\gamma[t_{1_x}, \dots, t_{n_x}]$ où $t_{i_x} = t_i$ si $i < l$ et $t_{i_x} = \text{unité}$ si $i \geq l$, et $y' = t\gamma[t'_{1_y}, \dots, t'_{n_y}]$ où $t'_{i_y} = t'_i$ si $i < l$ et $t'_{i_y} = \text{unité}$ si $i \geq l$, alors $x' = y'$. Par l'hypothèse d'induction, $t_i = t'_i$ pour tous les $i < l$. Nous pouvons faire de même pour prouver que $t_i = t'_i$ pour tous les $i > l$. Et par la proposition 4.4 maintenant, puisque $x = t_l \circ_l t''$ et $y = t'_l \circ_l t''$ pour un certain arbre t'' , alors $t_l = t'_l$.
- Si l'arbre a $k + 1$ arbres dans x qui ne sont pas des unités et sont tous du même côté de la racine, alors y doit avoir tous les arbres qui ne sont pas des racines du même côté, et doivent être les mêmes arbres que ceux de x . Maintenant, si nous choisissons un $o \in [n]$ tel que t_o n'est pas une unité et pour tout $j \in [n]$, tel que t_j n'est pas une unité, le nœud j n'est pas un descendant de o , alors en utilisant les mêmes arguments dans l'induction pour $n = 1$, nous pouvons prouver que $t'_o = t_o$. Maintenant, soit $x' = t\gamma[t_1, \dots, \text{unité}, \dots, t_n]$ où **unité** remplace t_o . Soit $y' = t\gamma[t'_1, \dots, \text{unité}, \dots, t'_n]$ où **unité**

remplace t'_o . Comme $t'_o = t_o$, alors $x' = y'$ et le nombre d'arbres dans x' qui ne sont pas des unités est maintenant k , donc par l'hypothèse d'induction, en plus du fait que $t_o = t'_o$, nous pouvons dire que $t_i = t'_i$ pour tous les $i \in [n]$.

□

4.2.2.2 Démonstration de l'injectivité

Nous procédons maintenant à la démonstration de l'injectivité du morphisme d'opéades

$$\phi : \text{Free}_G \longrightarrow \text{Dup}_G.$$

du théorème 4.7 .

Démonstration. Soient x et y deux arbres dans Free_G tels que $\phi(x) = \phi(y)$.

Cas de base : Pour $\deg(x) = 1 = \deg(y)$, l'assertion est vraie en raison de la nature du morphisme.

Étape inductive : Supposons que l'affirmation est vraie pour $\deg(x) = n$. Considérons x et y tels que $\deg(x) > n$ et $\phi(x) = \phi(y)$. Si $\phi(x) = t\gamma[\phi(x_1), \dots, \phi(x_n)]$ et $\phi(y) = t\gamma[\phi(y_1), \dots, \phi(y_n)]$, alors selon le lemme 4.8, $\phi(x_i) = \phi(y_i)$ pour tout $i \in [n]$. En appliquant l'hypothèse d'induction à chaque couple (x_i, y_i) , nous trouvons que $x_i = y_i$ pour tout $i \in [n]$. Par conséquent, $x = y$. □

PERSPECTIVES ET CONCLUSION

Pour conclure, les opérades sont des structures algébriques qui interagissent avec la combinatoire. Dans ce mémoire, les opérades sont vues comme des triplets $(\mathcal{O}, \circ_i, 1)$, où \mathcal{O} est une collection indexée par \mathbb{N} , \circ_i est une application vérifiant les axiomes mentionnés dans la section 3.1.2, et 1 est l'unité. Intuitivement, une opérade peut être considérée comme un ensemble contenant des opérateurs abstraits ayant de multiples entrées et d'une unique sortie qui peuvent être combinés de plusieurs façons. Pour préciser, considérons f comme un opérateur avec n entrées et g comme un opérateur avec m entrées, alors, $f \circ_i g$ représente l'opérateur qui possède $n + m - 1$ entrées, formé en connectant la sortie de g à la i^e entrée de f . Ceci est illustré dans la section 3.1.2.

Dans l'opérade dupliciale Dup , un élément $f \in Dup(n)$ est un arbre binaire avec n nœuds. Les entrées de f sont les nœuds de son arbre binaire. Si $g \in Dup(m)$, alors $f \circ_i g$ est un arbre binaire résultant de l'insertion de g dans le i^e nœud de f , comme vu dans la section 4.1. Dans notre étude de l'opérade dupliciale, l'une de nos découvertes principales est formulée dans la proposition 4.4. Cette proposition affirme que pour tout arbre binaire t dans $Dup(n)$, et pour tous t_1 et t_2 dans $\hat{\mathbb{B}}$, si les compositions $t \circ_i t_1$ et $t \circ_j t_2$ sont égales, il s'ensuit nécessairement que les indices i et j sont identiques et que les arbres t_1 et t_2 sont les mêmes. La preuve de ce théorème et les résultats utilisent le langage des arbres binaires introduit dans la section 4.1.2.

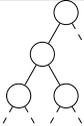
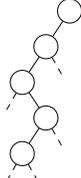
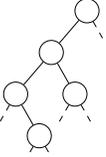
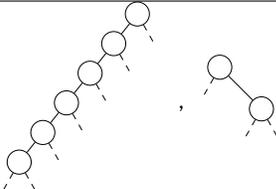
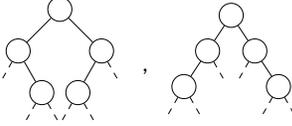
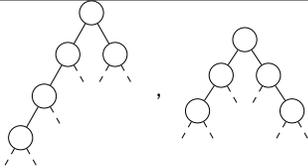
La section 4.1 présente également Dup via sa présentation (G, R) par générateurs. Pensé de cette façon, tout élément de Dup peut être écrit par des compositions partielles des éléments de G . Autrement dit, les deux arbres binaires de G , que nous appelons générateurs, avec la composition partielle de Dup , peuvent engendrer n'importe quel arbre binaire. Puisque G peut engendrer tous les arbres binaires, changer les générateurs résultera en un autre ensemble d'arbres, donc une sous-opérade de Dup que nous appellerons Dup_G . En utilisant l'algorithme de l'annexe A, nous avons pu compter le nombre d'arbres dans Dup_G pour certains G . Le tableau 4.1 fournit une partie des résultats, où la première colonne indique les éléments de G , la deuxième colonne indique le nombre d'arbres de Dup_G , classés par le nombre de nœuds. Dans le théorème 4.7, nous avons pu prouver que le nombre d'arbres de Dup_G , si G est un singleton contenant un arbre binaire de $\hat{\mathbb{B}}$, correspond aux nombres de Fuss-Catalan.

Dans cette section, nous listons certaines de nos découvertes qui n'ont pas été complétées. Cela constitue

une base pour des travaux futurs. Notons que cette section est principalement conjecturale.

Tout d'abord, le tableau ci-dessous répertorie quelques séries qui ont été trouvées en utilisant l'algorithme.

Tableau 4.2 Dimensions de sous-opérades Dup_G . Ici 0^n est équivalent à la sous-suite $0, 0, \dots, 0$ avec 0 utilisé n fois. Les références OEIS concernent uniquement les sous-suites des nombres strictement positifs.

| G | Suite de dimensions | OEIS référence et formule |
|---|--|--|
|  | $1, 0^2, 1, 0^2, 3, 0^2, 12, 0^2, 55, 0^2, 273, 0^2, 1428\dots$ | $\frac{\binom{3n}{n}}{2n+1}$: A001764 |
|  | $1, 0^3, 1, 0^3, 4, 0^3, 22, 0^3, 140, 0^3, 969, 0^3, 7084\dots$ | $\frac{\binom{4n}{n}}{3n+1}$: A002293 |
|  | $1, 0^3, 1, 0^3, 3, 0^3, 12, 0^3, 55, 0^3, 273, 0^3, 1428\dots$ | $\frac{\binom{3n}{n}}{2n+1}$: A001764 |
|  | $1, 1, 1, 2, 7, 22, 57, 128, 268, 573, \dots$ | A212384 |
|  | $1, 0^3, 2, 0^3, 20, 0^3, 280, 0^3, 4560\dots$ | $2^n \frac{\binom{5n}{n}}{4n+1}$: A217364 |
|  | $1, 0^3, 2, 0^3, 19, 0^3, 250, 0^3, 3803$ | nulle |

Ce tableau donne des résultats fascinants. Nous pouvons clairement voir que certains générateurs agissent bien, tandis que d'autres non. Par exemple, les générateurs de la ligne 5 du tableau peuvent suggérer que si G contient deux éléments t, t' tel que $t, t' \in \mathbb{B}$ et $|t| = |t'| = n$, alors $|Dup_G(i)| = \#\text{Ary}_{\perp}^{(n)}(i) \times 2^n$. Cependant, la ligne 6 a les mêmes propriétés mais ne les suit pas. Les lignes 1 et 2 suggèrent initialement

que si G est un singleton contenant t , et que t appartient à \mathbb{B} , alors $|Dup_G(i)| = \#Ary_{\perp}^{(n-1)}(i)$. Mais ce n'est pas vrai pour la ligne 3. Ce n'est pas vrai non plus dans notre expérimentation pour d'autres arbres. De nouvelles questions ont émergé, ouvrant la voie à des recherches plus approfondies :

- quelles sont les descriptions complètes des arbres binaires en tant que générateurs de sous-opéades dupliciales donnant des résultats significatifs ?
- Un partitionnement différent des arbres binaires en tant que générateurs peut-il offrir une description plus raffinée de nos opéades ?
- Quelles sont les caractéristiques des générateurs qui aboutissent à des séries de Fuss-Catalan ?
- Comment évoluent les opéades lorsque nous mélangeons des générateurs avec un nombre égal ou différent de nœuds ?

Une proposition en cours de rédaction, qui n'a pas été mentionnée dans ce mémoire mais qui peut être utile pour les travaux futurs relatifs aux sous-opéades dupliciales, est la suivante.

Proposition 4.9 *Considérons des arbres binaires t, t_1 dans Dup tels que $t, t_1 \in \mathbb{B}$ et $n_n(t) = n$. Soient i, j des entiers avec $1 \leq i, j \leq n$. Alors,*

$$i = 1 \text{ et } j = n \implies t \circ_i t_1 = t \circ_j t_1.$$

ANNEXE A

ALGORITHME

Dans notre étude des sous-opéades dupliciales, nous avons utilisé un algorithme pour trouver le nombre d'arbres distincts correspondant à des générateurs différents. Voici le pseudo-code de l'algorithme

Pseudo-code :

Fonction over(A, B):

Si B est vide,

 Renvoyer A

Sinon,

 Renvoyer (over(A, premier élément de B), reste de B)

Fonction under(B, A):

Si B est vide,

 Renvoyer A

Sinon,

 Renvoyer (premier élément de B, under(reste de B, A))

Fonction nodesnbr(t):

Si t est vide,

 Renvoyer 0

Sinon,

 Renvoyer 1 + nodesnbr(premier enfant de t) + nodesnbr(second enfant de t)

Fonction insroot(A, B):

 Renvoyer under(over(premier enfant de A, B), second enfant de A)

Fonction dup(A, B, n):

k = nodesnbr(premier enfant de A)

Si k est égal à n - 1,

 Renvoyer insroot(A, B)

Si k est supérieur ou égal à n,

 Renvoyer (dup(premier enfant de A, B, n), second enfant de A)

Sinon,

 j = n - k - 1

 Renvoyer (premier enfant de A, dup(second enfant de A, B, j))

Fonction createdup(G, d):

res = [((),())]

Si d est égal à 0,

 Renvoyer res

Sinon,

 Y = createdup(G, d - 1)

 Pour chaque y dans Y,

 Pour chaque g dans G,

 Pour i allant de 1 à nodesnbr(y),

 x = dup(y, g, i)

 Ajouter x à res

 Renvoyer liste des éléments uniques de res

Description :

Cet algorithme construit des arbres en se basant sur les principes de l'opérade dupliciale. Il implique des opérations telles que l'insertion d'un arbre dans un autre à une position spécifiée (fonctions over et under), le comptage des nœuds dans un arbre, et la construction récursive des arbres en appliquant les opérations de l'opérade dupliciale. Un arbre dans cet algorithme sera représenté par un tuple. Par exemple : les générateurs de l'opérade dupliciale seront représentés par $(((),()),())$ et $((),((),()))$

Utilisation :

Voici un exemple de la manière dont nous pouvons obtenir la suite d'entier de la ligne 4 du tableau 4.1. Tout d'abord, nous devons implémenter le pseudo-code en langage Python. Ensuite, pour obtenir le résultat souhaité en Python, nous ajouterons également ce qui suit :

```
from collections import Counter
tree = ((((),()),()),()),(((),((),())))
def count-nodes-summary(G, d):
    generated-trees = createdup(G, d)
    nodes-count = [nodesnbr(tree) for tree in generated-trees]
    count-summary = Counter(nodes-count)
    return count-summary
print(count-summary(tree,5)).
```

Après l'exécution de ce code, nous obtiendrons le résultat

```
Counter({21: 2530, 17: 285, 13: 35, 9: 5, 1: 1, 5: 1}).
```

Ce résultat signifie que nous avons engendré 1 arbre avec 5 nœuds, 1 arbre avec 1 nœud, 5 arbres avec 9 nœuds, 35 arbres avec 13 nœuds, 285 arbres avec 17 nœuds et 2530 arbres avec 21 nœuds.

BIBLIOGRAPHIE

- Bergeron, F. (2009). *Algebraic Combinatorics and Coinvariant Spaces*. CRC Press.
- Bergeron, F., Labelle, G. et Leroux, P. (1998). *Combinatorial species and tree-like structures*, volume 67 de *Encyclopedia of Mathematics and its Applications*. Cambridge: Cambridge University Press.
- Boardman, J. M. et Vogt, R. M. (1973). *Homotopy invariant algebraic structures on topological spaces*, volume 347 de *Lecture Notes in Mathematics*. Berlin: Springer-Verlag.
- Brouder, C. et Frabetti, A. (2001). Qed hopf algebras on planar binary trees. [arXiv:math/0112043].
- Bóna, M. (2011). *A Walk Through Combinatorics: An Introduction to Enumeration and Graph Theory*. World Scientific.
- Chapoton, F. (2008). Operads and algebraic combinatorics of trees. *Sém. Lothar. Combin.* 58, 2008.
- Even, S. (2011). *Graph Algorithms*. Cambridge University Press.
- Flajolet, P. et Sedgewick, R. (2009). *Analytic combinatorics*. Cambridge University Press, Cambridge.
- Foissy, L. (2017a). Algebraic structures associated to operads.
- Foissy, L. (2017b). Une introduction aux opérades. Paper presented at the Seminaire Laboratoire de Mathématiques Pures et Appliquées Joseph Liouville.
- Frabetti, A. (2008). Groups of tree-expanded series. *Journal of Algebra*, 319, 377–413.
- Giraud, S. (2015). Combinatorial operads from monoids. *J. Algebraic Combin.*, 41(2), 493–538.
- Giraud, S. (2017). Operads in algebraic combinatorics. Habilitation à diriger des recherches.
- Giraud, S. (2018). *Nonsymmetric Operads in Combinatorics*. Springer International Publishing.
- Jean-Louis Loday, B. V. a. (2012). *Algebraic Operads* (1 éd.). Grundlehren der mathematischen Wissenschaften №346. Springer.
- Knuth, D. E. (2011). *The art of computer programming. Vol. 4A. Combinatorial algorithms. Part 1*. Addison-Wesley, Upper Saddle River, NJ.
- Livernet, M. (2006). A rigidity theorem for pre-Lie algebras.
- Loday, J.-L. et Ronco, M. O. (2002). Order structure on the algebra of permutations and of planar binary trees. *Journal of Algebraic Combinatorics*, 15(3), 253–270.
- Loday, J.-L. et Vallette, B. (2012). *Algebraic operads*, volume 346 de *Grundlehren der mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer, Heidelberg.
- May, J. P. (1972). *The geometry of iterated loop spaces*, volume 271 de *Lecture Notes in Mathematics*.

Berlin: Springer-Verlag.

Méndez, M. A. (2015). *Set Operads in Combinatorics and Computer Science* (1 éd.). SpringerBriefs in Mathematics. Springer Cham.

Stanley, R. P. (1999a). *Enumerative Combinatorics*, volume 1. Cambridge University Press.

Stanley, R. P. (1999b). *Enumerative combinatorics. Vol. 2*, volume 62 de *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge.

Wilf, H. S. (2005). *Generatingfunctionology: Third Edition* (3 éd.). A K Peters/CRC Press.

Yau, D. (2016). *Colored Operads*. Graduate Studies in Mathematics. American Mathematical Society.