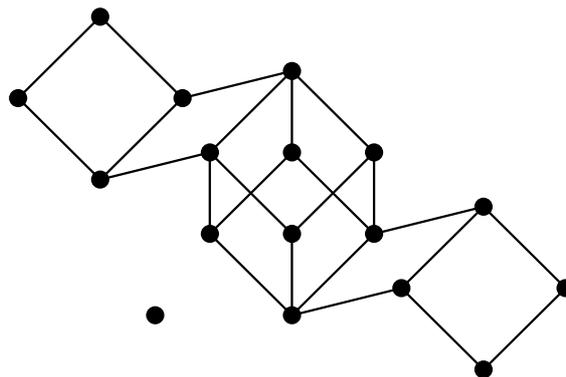


Université de Rouen
Département d'informatique
Mémoire de Master 2

Maître de stage : Florent Hivert

LES ARBRES ÉQUILIBRÉS DANS L'ORDRE DE TAMARI

Samuele Giraud



mercredi 1^{er} juillet 2009

Remerciements

Il va de soit que ce travail n'aurait jamais eu lieu sans l'encadrement exemplaire de mon maître de stage, Florent Hivert. Toujours généreux en temps, en anecdotes, en explications, en liberté de travail, ces quatre mois que j'ai passé a travailler avec lui ont été pour moi une expérience extrêmement agréable et enrichissante. Il est l'unique responsable de mon goût actuel pour la combinatoire. Je l'en remercie pour tout ce qu'il a fait.

Bien entendu, je remercie sincèrement l'ensemble des enseignants du Master 2 d'informatique théorique de l'université de Rouen. Leur implication et leur investissement dans ma formation n'ont pas besoin d'être démontrés. Ils sont totalement responsables de mon niveau actuel en informatique théorique et le sont en partie de ma progression future.

Pour finir, je remercie mes camarades de promotion pour leurs encouragements, les discussions intéressantes, et tout simplement pour leur compagnie. Les intéressés se reconnaîtront.

Table des matières

Introduction	ix
1 Généralités sur les arbres binaires	1
1.1 Les arbres binaires	1
1.1.1 Définition	1
1.1.2 Un peu de terminologie et quelques propriétés simples	2
1.1.3 Les différents parcours d'un arbre binaire	6
1.1.4 Un peu de dénombrement	7
1.2 Les arbres binaires étiquetés	8
1.2.1 Définition	8
1.2.2 Arbre binaire de recherche	8
1.3 Les arbres binaires équilibrés et l'opération de rotation	10
1.3.1 Définition	10
1.3.2 Quelques propriétés des arbres binaires équilibrés	11
1.3.3 L'opération de rotation	13
2 Quelques codages des arbres binaires	15
2.1 Codage par mots de Dyck	15
2.1.1 Définition	15
2.1.2 La bijection avec les arbres binaires	16
2.1.3 Comportements avec les opérations des arbres binaires	18
2.2 Codage par langages des feuilles	20
2.2.1 Classe des langages des feuilles	20
2.2.2 Comportements avec les opérations des arbres binaires	21
2.3 Codage par vecteurs de poids	22
2.3.1 Définition	22
2.3.2 La bijection avec les arbres binaires	23
2.3.3 Comportements avec les opérations des arbres binaires	24
3 Le treillis de Tamari	29
3.1 Définitions et propriétés de base	29
3.1.1 La relation d'ordre de Tamari	29
3.1.2 La structure de treillis	30
3.2 Quelques propriétés et notes bibliographiques	33
3.2.1 Quelques propriétés	33
3.2.2 Manifestations de l'ordre de Tamari et remarques	34

4	Les arbres équilibrés et l'ordre de Tamari	37
4.1	Application et motif de déséquilibre	37
4.1.1	Application de déséquilibre	37
4.1.2	Motifs de déséquilibre	38
4.2	Modification des déséquilibres pour la relation de couverture	40
4.2.1	Modification des déséquilibres locaux	40
4.2.2	Propagation des modifications des déséquilibres	42
4.3	Une classification des arbres équilibrés dans le treillis de Tamari	46
4.3.1	Les classes d'arbres équilibrés	47
4.3.2	Dénombrement des classes d'arbres équilibrés	48
4.4	Les intervalles d'arbres équilibrés	50
4.4.1	Description des intervalles d'arbres équilibrés	50
4.4.2	Dénombrement des intervalles d'arbres équilibrés	56
	Conclusion	59

Liste des symboles

$[x, y]$	intervalle où x (resp. y) est le plus petit (resp. le plus grand) élément
$\#E$	cardinal de l'ensemble E
\mathbb{N}	ensemble des entiers naturels
\mathbb{Z}	ensemble des entiers relatifs
$\text{Fact}(u)$	ensemble des facteurs du mot u
$\text{ham}(u, v)$	distance de Hamming entre les mots u et v
$\text{inf}(E)$	borne inférieure de l'ensemble ordonné E
$[x^n] f(x)$	coefficient de x^n dans la série génératrice $f(x)$
\log_a	logarithme en base a
$\max_{\text{lex}} L$	plus grand élément du langage L pour l'ordre lexicographique
$\min_{\text{lex}} L$	plus petit élément du langage L pour l'ordre lexicographique
$\text{Pref}(u)$	ensemble des préfixes du mot u
\sqcup	union disjointe
$\text{Suff}(u)$	ensemble des suffixes du mot u
$ i $	valeur absolue de l'entier i
$ u $	longueur du mot u
$ u _a$	nombre d'occurrences de la lettre a dans le mot u
L^*	étoile de Kleene du langage L
L^+	raccourci pour $L.L^*$
$u.v$	concaténation des mots u et v
$u.v^{-1}$	quotient à droite du mot u par le mot v
$u[i..j]$	facteur du mot u de position gauche i et de position droite j
$u[i]$	i ème lettre du mot u où $1 \leq i \leq u $
$v^{-1}.u$	quotient à gauche du mot u par le mot v

Chapitre 0

Introduction

Ce mémoire s'inscrit dans le domaine de la combinatoire, et plus précisément à mi-chemin entre la combinatoire algébrique et la combinatoire énumérative. La question de départ est très simple :

Il existe d'un côté un objet mathématique introduit depuis bientôt un demi-siècle connu sous le nom de treillis de Tamari. Une relation d'ordre partielle est définie sur les objets combinatoires catalans, dont les arbres binaires, les mots de Dyck, les triangulations, pour ne donner qu'une courte liste, sont les représentants[23, 24]. Si l'on considère les arbres binaires comme représentants, l'arbre T_0 est déclaré inférieur à l'arbre T_1 s'il est possible de transformer l'arbre T_0 par des opérations de rotation à droite en l'arbre T_1 . Beaucoup de propriétés intéressantes sur ce treillis sont déjà connues[8, 10, 16, 18, 7].

D'autre part, il existe un autre ensemble d'objets, de nature plutôt algorithmique, connus sous le nom d'arbres binaires équilibrés, étonnamment introduits presque en même temps que le treillis de Tamari, mais de manière totalement indépendante[2]. L'utilité de cet objet vient du fait que lorsque l'on attribue des étiquettes à chacun de ses nœuds, il peut être utilisé comme structure de donnée pour représenter un ensemble dynamique offrant des complexités temporelles logarithmiques en fonction de son nombre de nœuds pour la plupart des opérations. La rotation est utilisée pour conserver l'équilibre de l'arbre lors d'ajouts et de suppressions d'éléments.

La question devient maintenant presque naturelle : les arbres équilibrés jouent-ils un rôle particulier dans le treillis de Tamari ? À notre connaissance, cette question n'a jamais été posée, ce qui n'est pas si étonnant à la lumière du fait que les experts du treillis de Tamari ne sont pas forcément au courant du vaste champ que recouvre l'algorithmique, et ne connaissent donc pas nécessairement le concept d'arbre équilibré. Le fil conducteur de cette recherche se dessine autour de la conjecture suivante :

Conjecture 1.

L'ensemble des arbres équilibrés à n nœuds est clos par intervalle dans le treillis de Tamari.

Cette conjecture semble être vraie puisque nous l'avons confirmée expérimentalement pour les arbres binaires de moins de 22 nœuds.

Le travail de recherche résumé dans ce mémoire s'est déroulé en plusieurs parties. Commencé au début du mois de mars, nous nous sommes plongés dans la littérature abondante du treillis de Tamari pendant environ un mois. Cette étude bibliographique nous a permis de nous familiariser avec cet objet. Il nous a rapidement semblé naturel, pour résoudre la conjecture précédente, de travailler sur un codage des arbres binaires qui, simultanément, permet de décider si deux arbres

sont comparables pour l'ordre de Tamari et de décider si l'arbre encodé est équilibré. Nous avons étudié pour cela divers objets catalans mais sans succès[23, 24, 25]. Nous en sommes arrivés à la conclusion que nous ne pouvons pas traduire ces deux aspects dans un même codage. L'avantage de cette étude est que nous avons isolé des objets catalans déjà connus mais peu mentionnés : les vecteurs de poids[12, 13]. Ces vecteurs possèdent plusieurs propriétés intéressantes et donnent en particulier une réponse immédiate au problème consistant à décider si deux arbres sont comparables. Fort des connaissances que nous avons acquises au cours de ces premières semaines de travail, nous avons essayé de démontrer la conjecture. Pendant environ trois semaines, nous avons tenté divers schémas de preuve mais sans succès. Nous avons conclu que ce problème est difficile et nous l'avons donc laissé de côté. Naturellement, notre intérêt s'est ensuite porté sur la restriction de l'ordre de Tamari aux arbres équilibrés et nous avons obtenus plusieurs résultats nouveaux. Il nous a fallu définir un outillage pour obtenir un cadre formel à l'établissement de nos résultats.

La démarche que nous avons adoptée est en très grande partie expérimentale. Chacune des questions que nous nous sommes posées ont été vérifiées par la puissance de calcul offerte par l'outil informatique[1]. De cette façon, avant même de tenter de construire une démonstration d'une propriété intuitive qui semble vraie à première vue, pour se rendre compte au bout de longues lignes de manipulation de symboles et de longues heures de réflexion qu'elle est totalement fautive, la machine, servant de garde-fou, peut construire en quelques secondes un contre-exemple infirmant la conjecture et nous faire épargner ainsi temps et énergie.

Notre travail est organisé comme suit. Le **chapitre 1** est consacré aux définitions de base sur les arbres binaires et les arbres équilibrés. Nous rappelons quelques propriétés fondamentales ainsi que les opérations usuelles définies sur l'ensemble des arbres binaires. Les notions importantes de ce chapitre sont l'opération de rotation et l'adressage des nœuds.

Dans le **chapitre 2** nous résumons notre étude des différents codages des arbres binaires. Nous nous sommes concentrés sur trois codages particuliers : le codage par mot de Dyck, par langage des feuilles et par vecteur de poids. Les mots de Dyck sont des objets combinatoires très célèbres et il nous a paru intéressant d'étudier leur comportement vis-à-vis de l'opération de rotation. Nous établissons ensuite une bijection des arbres binaires avec un langage particulier appelé langage des feuilles pour finir par étudier le codage qui nous intéresse le plus : le codage par vecteur de poids.

Dans le **chapitre 3**, nous résumons notre travail sur l'ordre de Tamari. Nous montrons qu'il s'agit d'un treillis et donnons un algorithme pour calculer la borne inférieure d'un couple d'arbres. Nous donnons de plus un encadrement sur les longueurs des chaînes qui incluent le plus petit et le plus grand élément et finissons en donnant quelques propriétés ainsi que leurs références bibliographiques.

Le **chapitre 4** contient la partie originale de ce travail et donne un certain nombre de réponses sur le rôle des arbres équilibrés dans le treillis de Tamari. Nous y introduisons les notions d'application et de motif de déséquilibre. Ces notions sont essentielles pour l'établissement de nos résultats. Nous réalisons ensuite une étude des modifications des déséquilibres provoquées par une opération de rotation. Dans la foulée, nous classifions les arbres équilibrés dans l'ordre de Tamari et donnons une condition nécessaire et suffisante sur les motifs de déséquilibre pour déterminer la classe de l'arbre. Nous comptons ensuite le nombre d'arbres qui appartiennent à ces classes et montrons notre résultat principal qui énonce que les intervalles composés

d'arbres équilibrés dans l'ordre de Tamari sont isomorphes à un treillis hypercube et donnons les équations fonctionnelles des séries génératrices qui comptent le nombre d'intervalles composés d'arbres équilibrés et le nombre d'hypercubes maximaux formés par les intervalles d'arbres équilibrés.

Chapitre 1

Généralités sur les arbres binaires

Les arbres binaires sont des objets informatiques très importants. Ils sont à la base d'algorithmes de traitement de données efficaces. Dans cette partie, nous allons poser la plupart des définitions qui nous seront utiles par la suite en introduisant la notion d'arbre binaire, d'arbre étiqueté, d'arbre binaire de recherche puis d'arbre équilibré. Nous rappelons aussi certaines de leurs propriétés fondamentales.

1.1 Les arbres binaires

1.1.1 Définition

Définition 1.1.1. (Arbre binaire)

Un *arbre binaire* est défini récursivement comme étant :

- ou bien une *feuille* ;
- ou bien un couple composé de deux arbres binaires.

L'arbre binaire réduit à une feuille, également nommé *l'arbre vide*, est noté \perp . Si $T = (G, D)$ est un arbre binaire, alors nous dirons que l'arbre binaire G (resp. l'arbre binaire D) est *l'arbre fils gauche* (resp. *arbre fils droit*) de T . Au lieu de représenter les arbres binaires conformément à la définition 1.1.1 par des couples imbriqués, il est d'usage de les dessiner en suivant la règle suivante : si $T = (G, D)$, alors T est représenté en dessinant un *nœud* (représenté par un cercle) par lequel sortent deux *arcs*, l'un dirigé vers la gauche, l'autre vers la droite pointant vers le fils gauche et le fils droit de T . Si $T = \perp$, alors T est représenté en dessinant un arc terminé par un segment (voir Figure 1.1).

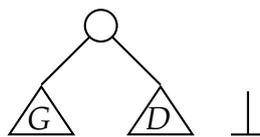
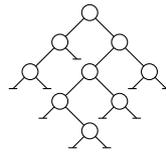


FIG. 1.1 – Représentations graphiques des arbres (G, D) et \perp .

Exemple.

Soit $T_0 = (((\perp, \perp), \perp), ((\perp, (\perp, \perp)), (\perp, \perp)), (\perp, \perp))$. Sa représentation graphique est



1.1.2 Un peu de terminologie et quelques propriétés simples

Dans la suite de ce mémoire, nous considérerons tantôt les arbres comme des objets récursifs tantôt comme des graphes, sans toujours l'évoquer de manière explicite, les deux approches étant équivalentes. En raisonnant sur la représentation graphique des arbres binaires, faisons un peu de terminologie :

- si $T = (G, D)$, alors T est représenté par , et le nœud visible dans cette représentation est appelé *racine* de T . Il est clair qu'un arbre possède une unique racine.
- Si $T = (G, D)$ avec $G \neq \perp$, alors le nœud racine de G est le *fil gauche* du nœud racine de T qui lui-même est son *père*. Symétriquement, si $D \neq \perp$, le nœud racine de D est le *fil droit* du nœud racine de T qui lui-même est son père. Le fil gauche et le fil droit éventuels du nœud racine de T sont nommés simplement *fil* du nœud racine de T si l'on ne désire pas effectuer la distinction entre fil gauche et fil droit. Deux nœuds qui possèdent un même père sont appelés *frères*. Il est clair qu'un nœud peut avoir 0, 1, ou 2 fils et 0 ou 1 frère. Tout nœud d'un arbre possède un unique père, excepté la racine qui n'en possède pas.
- Un *chemin* dans un arbre binaire T est une suite de nœuds (x_1, \dots, x_k) tel que x_i est le père du nœud x_{i+1} pour $1 \leq i \leq k-1$. La *longueur du chemin* est la valeur $k-1$.
- Un nœud y est un *descendant* d'un nœud x dans un arbre binaire T s'il existe un chemin de la forme (x, \dots, y) . Similairement, le nœud x est appelé *ancêtre* du nœud y . Si ce chemin est de longueur supérieure à 1, on parle de *descendant propre* et d'*ancêtre propre*.
- Le *niveau* d'un nœud x dans un arbre binaire T est la longueur de l'unique chemin (r, \dots, x) où r est la racine de T .
- Un *sous-arbre* S d'un arbre binaire T est un arbre construit en sélectionnant un nœud x de T et en considérant uniquement les nœuds descendants de x . L'ensemble des sous-arbres de T est noté $\text{SArbre}(T)$.

Voici une première statistique importante sur les arbres binaires :

Définition 1.1.2. (Nombre de nœuds)

Soit T un arbre binaire. Le *nombre de nœuds* de T , noté $\text{nd}(T)$, est défini récursivement par :

$$\text{nd}(T) = \begin{cases} 0 & \text{si } T = \perp, \\ 1 + \text{nd}(G) + \text{nd}(D) & \text{avec } T = (G, D) \text{ sinon.} \end{cases} \quad (1.1)$$

Muni de cette statistique, nous pouvons partitionner la totalité des arbres binaires en sous-ensembles contenant uniquement les arbres binaires qui possèdent un nombre fixé de nœuds. Nous notons donc \mathcal{A}_n l'ensemble des arbres binaires à n nœuds. L'ensemble $\cup_{n \geq 0} \mathcal{A}_n$, noté \mathcal{A}_∞ , est l'ensemble de tous les arbres binaires.

Il est aussi possible de considérer le nombre de feuilles d'un arbre binaire :

Définition 1.1.3. (Nombre de feuilles)

Soit T un arbre binaire. Le *nombre de feuilles* de T , noté $\text{nf}(T)$, est défini récursivement par :

$$\text{nf}(T) = \begin{cases} 1 & \text{si } T = \perp, \\ \text{nd}(G) + \text{nd}(D) & \text{avec } T = (G, D) \text{ sinon.} \end{cases} \quad (1.2)$$

Les quantités $\text{nd}(T)$ et $\text{nf}(T)$ satisfont la propriété suivante :

Proposition 1.1.1. (Relation nombre de feuilles, nombre de nœuds)

Si T est un arbre binaire alors

$$\text{nd}(T) + 1 = \text{nf}(T). \quad (1.3)$$

Preuve. Par induction structurelle sur l'arbre T .

Si $T = \perp$, nous avons $\text{nd}(\perp) = 0$ et $\text{nf}(\perp) = 1$, ce qui est conforme à la proposition.

Supposons la proposition vraie pour deux arbres binaires G et D et considérons l'arbre $T = (G, D)$. Nous avons

$$\text{nf}(T) = \text{nf}(G) + \text{nf}(D) \text{ (définition 1.1.3)} \quad (1.4)$$

$$= \text{nd}(G) + 1 + \text{nd}(D) + 1 \text{ (hypothèse d'induction)} \quad (1.5)$$

$$= \text{nd}(T) + 1 \text{ (définition 1.1.2)} \quad (1.6)$$

ce qui montre la proposition. \square

Une autre statistique importante des arbres binaires est la hauteur :

Définition 1.1.4. (Hauteur)

Soit T un arbre binaire. La *hauteur* de T , notée $\text{ht}(T)$, est définie récursivement par :

$$\text{ht}(T) = \begin{cases} 0 & \text{si } T = \perp, \\ \max\{\text{ht}(G), \text{ht}(D)\} + 1 & \text{avec } T = (G, D) \text{ sinon.} \end{cases} \quad (1.7)$$

Autrement dit, la hauteur d'un arbre binaire est la longueur du plus long chemin qu'il contient, majorée de 1. Les quantités $\text{nd}(T)$ et $\text{ht}(T)$ satisfont la propriété suivante :

Proposition 1.1.2. (Relation hauteur, nombre de nœuds)

Si T un arbre binaire alors

$$\log_2(\text{nd}(T) + 1) \leq \text{ht}(T). \quad (1.8)$$

Preuve. Il y a au plus 2^i nœuds au niveau i dans T . Comme tout nœud de T est de niveau compris entre 0 et $\text{ht}(T) - 1$ et que l'on a l'identité

$$\sum_{0 \leq i \leq \text{ht}(T)-1} 2^i = 2^{\text{ht}(T)} - 1, \quad (1.9)$$

il y a au plus $2^{\text{ht}(T)} - 1$ nœuds dans T ce qui implique $\text{nd}(T) + 1 \leq 2^{\text{ht}(T)}$ et finalement $\log_2(\text{nd}(T) + 1) \leq \text{ht}(T)$. \square

Nous proposons maintenant un moyen d'adresser un nœud dans un arbre binaire.

Définition 1.1.5. (Adresse d'un sous-arbre)

Soit T un arbre binaire. Une *adresse d'un sous-arbre* de T est un mot sur l'alphabet $\{0, 1\}$ auquel correspond le sous-arbre de T , noté $\text{SArbre}(T, u)$, défini récursivement de la manière suivante :

$$\text{SArbre}(T, u) = \begin{cases} T & \text{si } u = \epsilon, \\ \text{SArbre}(G, u') & \text{pour } T = (G, D) \text{ si } u = 0u' \text{ avec } u' \in \{0, 1\}^*, \\ \text{SArbre}(D, u') & \text{pour } T = (G, D) \text{ si } u = 1u' \text{ avec } u' \in \{0, 1\}^*, \\ \perp & \text{sinon.} \end{cases} \quad (1.10)$$

Une adresse de sous-arbre u dans un arbre binaire T est simplement un chemin, d'origine la racine, déterminé en lisant u de gauche à droite et en se déplaçant du nœud courant vers son fils gauche si la lettre lue est 0, et vers son fils droit si la lettre lue est 1. Une adresse de sous-arbre permet donc de spécifier sans ambiguïté un sous-arbre d'un arbre binaire, et donc en particulier un nœud, la racine du sous-arbre considéré. C'est d'ailleurs principalement pour désigner des nœuds que nous allons utiliser cette notion d'adressage par la suite.

Nous notons

$$\mathbb{A}_N(T) = \{u \in \{0, 1\}^* \mid \text{SArbre}(T, u) \neq \perp\}, \quad (1.11)$$

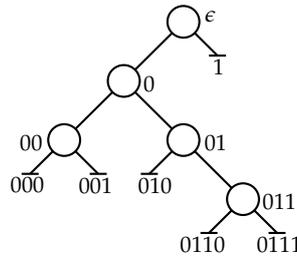
l'ensemble des adresses des nœuds de T et

$$\mathbb{A}_F(T) = \{u \in \{0, 1\}^* \mid \text{SArbre}(T, u) = \perp \text{ et } u.\{0, 1\}^{-1} \in \mathbb{A}_N(T)\}, \quad (1.12)$$

l'ensemble des adresses des feuilles de T .

Exemple.

Considérons l'arbre binaire T :



Nous avons $\mathbb{A}_N(T) = \{\epsilon, 0, 00, 01, 011\}$ et $\mathbb{A}_F(T) = \{000, 001, 010, 0110, 0111, 1\}$.

Posons ce premier lemme, très simple :

Lemme 1.1.1. (Adresse du père d'un nœud)

Soit T un arbre binaire et $u \in \{0, 1\}^+$ l'adresse d'un nœud x différent de la racine. L'adresse du père du nœud x est $u.\{0, 1\}^{-1}$.

Preuve. Le nœud x admet un unique père puisque il n'est pas racine de T . On pose donc y le père de x . Il existe dans T un unique chemin (r, \dots, y, x) avec r racine de T . En posant v l'adresse du nœud y , si x est fils gauche de y , alors l'adresse de x est $v.0$ et s'il est fils droit de y , alors son adresse est $v.1$. Nous avons donc bien $v = v.\{0, 1\}.\{0, 1\}^{-1}$. \square

Voici une caractéristique intéressante du langage $\mathbb{A}_N(T)$:

Proposition 1.1.3. (Langage des adresses des nœuds)

Si T est un arbre binaire, alors $\mathbb{A}_N(T) = \text{Pref}(\mathbb{A}_N(T))$.

Preuve. Montrons l'inclusion réciproque de ces deux langages. Il est clair que $\mathbb{A}_N(T) \subseteq \text{Pref}(\mathbb{A}_N(T))$. Pour montrer l'inclusion inverse, raisonnons par l'absurde en supposant qu'il existe un mot $uv \in \mathbb{A}_N(T)$ avec $u \in \{0, 1\}^*$ et $v \in \{0, 1\}^+$ tel que $u \notin \mathbb{A}_N(T)$. Nous pouvons toujours nous ramener au cas où $|v| = 1$, et il devient clair que u est l'adresse d'un nœud de T qui est, d'après le Lemme 1.1.1, le père du nœud d'adresse uv et donc $u \in \mathbb{A}_N(T)$, ce qui est contradictoire avec notre hypothèse. \square

Rappelons quatre lois de composition internes dans l'ensemble \mathcal{A}_∞ introduites par J.-L. Loday[15].

Définition 1.1.6. (Lois de composition sur \mathcal{A}_∞)

Soient T_1 et T_2 deux arbres binaires.

1. L'opérateur \wedge , *opérateur de greffe* construit l'arbre $T_3 = T_1 \wedge T_2$ comme étant l'arbre qui possède T_1 comme fils gauche et T_2 comme fils droit.
2. L'opérateur $/$, *opérateur au-dessus*, construit l'arbre $T_3 = T_1/T_2$ comme étant l'arbre T_2 qui possède T_1 comme sous-arbre d'adresse u , avec $u = 0^k$ et $u \in \mathbb{A}_F(T_2)$ avec k maximal.
3. L'opérateur \setminus , *opérateur en-dessous*, construit l'arbre $T_3 = T_1 \setminus T_2$ comme étant l'arbre T_1 qui possède T_2 comme sous-arbre d'adresse u , avec $u = 1^k$ et $u \in \mathbb{A}_F(T_1)$ avec k maximal.
4. L'opérateur \leftrightarrow , *opérateur de symétrie*, possède un unique arbre binaire comme opérande, construit l'arbre $T_3 = \leftrightarrow(T_1)$ défini récursivement par :

$$\leftrightarrow(T) = \begin{cases} \perp & \text{si } T = \perp, \\ \leftrightarrow(D) \wedge \leftrightarrow(G) & \text{pour } T = (G, D) \text{ sinon.} \end{cases} \quad (1.13)$$

Dans l'opération T_1/T_2 (resp. $T_1 \setminus T_2$), la feuille la plus à gauche de T_2 (resp. la feuille la plus à droite de T_1), ou autrement dit, la feuille qui possède la plus petite adresse (resp. la plus grande) pour l'ordre lexicographique en posant $0 < 1$, est remplacée par l'arbre T_1 (resp. par l'arbre T_2).

L'opération de greffe n'est ni associative, ni commutative. Les opérations au-dessus et en-dessous sont associatives et non commutatives.

Nous avons de plus les relations suivantes, facilement vérifiables. Pour tout arbres T_1, T_2, T_3 :

$$\text{nd}(T_1 \wedge T_2) = \text{nd}(T_1) + \text{nd}(T_2) + 1; \quad (1.14)$$

$$\text{nd}(T_1/T_2) = \text{nd}(T_1) + \text{nd}(T_2); \quad (1.15)$$

$$\text{nd}(T_1 \setminus T_2) = \text{nd}(T_1) + \text{nd}(T_2); \quad (1.16)$$

$$\text{nd}(\leftrightarrow(T_1)) = \text{nd}(T_1); \quad (1.17)$$

$$\text{ht}(T_1 \wedge T_2) = \max\{\text{ht}(T_1), \text{ht}(T_2)\} + 1; \quad (1.18)$$

$$\text{ht}(T_1/T_2) = \max\{\text{ht}(T_2), \text{ht}(T_1) + |u|\} \text{ où } u = \min_{\text{lex}} \mathbb{A}_F(T_2); \quad (1.19)$$

$$\text{ht}(T_1 \setminus T_2) = \max\{\text{ht}(T_1), \text{ht}(T_2) + |u|\} \text{ où } u = \max_{\text{lex}} \mathbb{A}_F(T_1); \quad (1.20)$$

$$\text{ht}(\leftrightarrow(T_1)) = \text{ht}(T_1); \quad (1.21)$$

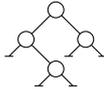
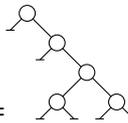
$$T_1 / \perp = \perp / T_1 = T_1; \quad (1.22)$$

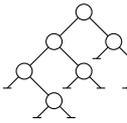
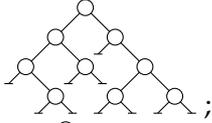
$$T_1 \setminus \perp = \perp \setminus T_1 = T_1; \quad (1.23)$$

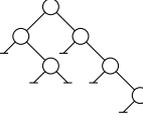
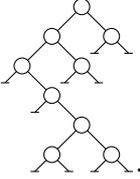
$$T_1 / (T_2 \wedge T_3) = (T_1 / T_2) \wedge T_3; \quad (1.24)$$

$$T_1 \wedge (T_2 \setminus T_3) = (T_1 \wedge T_2) \setminus T_3. \quad (1.25)$$

Exemple.

Soient $T_0 =$  et $T_1 =$ . Nous avons alors :

$T_0 \wedge T_1 =$  ; $T_0 / T_1 =$  ;

$T_0 \setminus T_1 =$  ; $T_1 / T_0 =$ .

1.1.3 Les différents parcours d'un arbre binaire

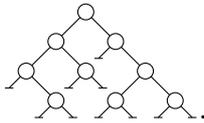
Étant donné un arbre binaire, il est intéressant de pouvoir visiter chacun de ses nœuds selon un ordre bien défini. Il existe pour cela divers types de parcours rappelés ci-dessous :

Définition 1.1.7. (Parcours d'arbre)

Soit T un arbre binaire.

- Le *parcours préfixe gauche droite* (resp. *droite gauche*) de T consiste à visiter récursivement la racine de T , puis son sous-arbre gauche (resp. sous-arbre droit), et enfin son sous-arbre droit (resp. sous-arbre gauche).
- Le *parcours infix gauche droite* (resp. *droite gauche*) de T consiste à visiter récursivement le sous-arbre gauche (resp. sous-arbre droit) de T , puis sa racine, et enfin son sous-arbre droit (resp. sous-arbre gauche).
- Le *parcours suffixe gauche droite* (resp. *droite gauche*) de T consiste à visiter récursivement le sous-arbre gauche (resp. sous-arbre droit) de T , puis son sous-arbre droit (resp. sous-arbre gauche), et enfin sa racine.

Exemple.



Considérons l'arbre

Voici les suites des adresses des nœuds de cet arbre visités par les parcours préfixe, infix et suffixe gauche droite.

Type de parcours	liste des adresses des nœuds parcourus
préfixe gauche droite	$(\epsilon, 0, 00, 001, 01, 1, 11, 110, 111)$
infix gauche droite	$(00, 001, 0, 01, \epsilon, 1, 110, 11, 111)$
suffixe gauche droite	$(001, 00, 01, 0, 110, 111, 11, 1, \epsilon)$

Dans la suite, sauf mention contraire, lorsque nous parlerons de l'un de ces parcours sans préciser le sens droite gauche ou gauche droite, cela signifie qu'il s'agira du parcours gauche droite, plus naturel que son homologue.

1.1.4 Un peu de dénombrement

Intéressons-nous au comptage des arbres binaires selon leur nombre de nœuds.

Théorème 1.1.1. (Cardinal de \mathcal{A}_n)

Pour tout $n \geq 0$,

$$\#\mathcal{A}_n = \frac{1}{n+1} \binom{2n}{n} \quad (1.26)$$

Preuve. Pour compter le nombre d'arbres de \mathcal{A}_n , nous devons compter les arbres binaires selon leur nombre de nœuds. Nous savons qu'un arbre binaire est soit une feuille ayant 0 nœuds soit un couple (G, D) composé de deux arbres binaires G et D dont le nombre de nœuds est 1 plus le nombre de nœuds des arbres G et D .

Par la méthode symbolique, en posant A la classe combinatoire de l'ensemble des arbres binaires, nous avons

$$A = \{\perp\} \sqcup A \times \{\circ\} \times A \quad (1.27)$$

où \perp est une feuille de taille 0 et \circ un nœud de taille 1. En posant

$$A(z) = \sum_{n \geq 0} \#\mathcal{A}_n z^n = \sum_{n \geq 0} a_n z^n \quad (1.28)$$

la série génératrice des arbres binaires comptés selon leur nombre de nœuds, nous avons

$$A(z) = 1 + zA(z)^2, \quad (1.29)$$

qui admet $A(z) = \frac{1 - \sqrt{1-4z}}{2z}$ comme solution. Les coefficients de cette série génératrice vérifient bien $a_n = \frac{1}{n+1} \binom{2n}{n}$. \square

Cette suite a_0, a_1, a_2, \dots dont les premières valeurs sont 1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, ... est bien connue. C'est la fameuse suite de Catalan. En plus des arbres binaires, elle compte de très nombreux objets combinatoires[23, 24]. En vertu de ce fait, ces objets sont qualifiés de *catalans*.

Il est aussi possible de compter les arbres binaires selon leur hauteur, mais cela est beaucoup plus difficile avec l'outillage fourni par les séries génératrices et la méthode symbolique. La difficulté réside dans le fait que la hauteur d'un arbre binaire est définie par le biais de la fonction max rendant l'écriture de sa série génératrice par la méthode symbolique très ardue. Il est néanmoins possible de façon relativement simple de donner une formule de récurrence pour la suite a_h égale au nombre d'arbres binaires de hauteur h . En effet :

$$a_h = \begin{cases} 1 & \text{si } h = 0, \\ a_{h-1}^2 + 2a_{h-1} \sum_{0 \leq i \leq h-2} a_i & \text{sinon.} \end{cases} \quad (1.30)$$

Cette relation de récurrence s'explique facilement de part le fait que pour construire un arbre binaire de hauteur h , il suffit de combiner un arbre binaire de hauteur $h-1$ avec un arbre binaire de hauteur comprise entre 0 et $h-2$, ce qui est possible de faire de deux façons différentes à chaque fois, suivant le fait que l'arbre de hauteur $h-1$ est sous-arbre gauche ou sous-arbre droit de l'arbre construit, ou en combinant deux arbres binaires de hauteur $h-1$. Les premières valeurs de cette suite sont 1, 1, 3, 21, 651, 457653, 210065930571, ...

1.2 Les arbres binaires étiquetés

Les arbres binaires étiquetés sont des objets combinatoires très semblables aux arbres binaires non étiquetés que nous venons de décrire. Contrairement à ces derniers, à chaque nœud d'un arbre binaire étiqueté est associé un élément d'un ensemble arbitraire. En pratique, utilisés dans des algorithmes de gestion d'ensembles dynamiques, ces arbres binaires sont étiquetés par des clés qui correspondent à des pointeurs vers des données.

1.2.1 Définition

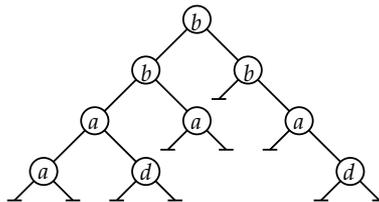
Définition 1.2.1. (Arbre binaire étiqueté)

Un *arbre binaire étiqueté* est un triplet (T, α, Σ) composé d'un arbre binaire et d'une application α qui associe à chaque nœud de l'arbre un élément d'un ensemble Σ .

Bien entendu, la terminologie propre aux arbres binaires qui est décrite dans la section précédente s'étend naturellement aux arbres binaires étiquetés. L'arbre binaire non étiqueté sous-jacent à un arbre binaire étiqueté (T, α, Σ) est noté $\text{Forme}((T, \alpha, \Sigma))$.

Exemple.

Soit l'arbre binaire étiqueté $(T, \alpha, \Sigma) =$



avec $\Sigma = \{a, b, c, d\}$. Les étiquettes associées à chacun des nœuds de T sont représentées à l'intérieur des cercles qui les représentent. Par un léger abus de notation, en confondant nœud et adresse d'un nœud, l'application α vérifie $\alpha(\epsilon) = b$, $\alpha(0) = b$, $\alpha(00) = a$, $\alpha(000) = a$, $\alpha(001) = d$, $\alpha(01) = a$, $\alpha(1) = b$, $\alpha(11) = a$ et $\alpha(111) = d$.

Dans la suite, par soucis de concision, nous n'allons pas faire référence explicitement à l'application d'étiquetage α et à l'ensemble des étiquettes Σ d'un arbre binaire étiqueté, sauf si cela est indispensable. Nous parlerons simplement d'un arbre étiqueté par le biais de sa forme, en évoquant les étiquettes de ses nœuds si nécessaire.

1.2.2 Arbre binaire de recherche

Les arbres binaires de recherche sont des arbres binaires étiquetés utilisés comme structure de donnée pouvant supporter les opérations propres à un ensemble dynamique comme :

- l'ajout d'un élément ;
- la suppression d'un élément ;
- la recherche d'un élément ;
- la recherche de l'élément maximal (resp. minimal) ;
- étant donné un élément, la recherche de son successeur (resp. son prédécesseur).

L'ensemble représenté doit être muni d'une relation d'ordre totale.

Définition 1.2.2. (Arbre binaire de recherche)

Un *arbre binaire de recherche* est un tuple $(T, \alpha, \Sigma, \leq)$ où (T, α, Σ) est un arbre binaire étiqueté et \leq une relation d'ordre totale sur Σ .

De plus, pour tout sous-arbre non vide $S = G \wedge D$ de T , l'étiquette de la racine de S est supérieure aux étiquettes des nœuds de G et est strictement inférieure aux étiquettes des nœuds de D pour la relation \leq (voir Figure 1.2).

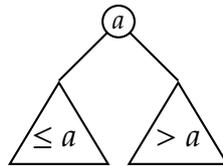


FIG. 1.2 – Condition d'arbre binaire de recherche.

Nous n'allons pas présenter les différents algorithmes de manipulation des arbres binaires de recherche. Ceci est déjà présenté par divers auteurs[9, 12, 3].

Les étiquettes d'un arbre binaire de recherche vérifient cependant une propriété remarquable :

Proposition 1.2.1. (Parcours infixe d'un arbre binaire de recherche)

Soit $(T, \alpha, \Sigma, \leq)$ un arbre binaire de recherche et $(\alpha(x_1), \dots, \alpha(x_n))$ la suite des étiquettes des nœuds de T obtenus par le parcours infixe de T . Alors, $\alpha(x_i) \leq \alpha(x_{i+1})$ pour $1 \leq i \leq n - 1$.

Preuve. Par induction structurelle sur l'arbre T .

Si T est l'arbre vide, la proposition est trivialement vraie.

Supposons que $T = G \wedge D$ pour deux arbres binaires de recherche G et D tels que le parcours infixe des nœuds de G (resp. des nœuds de D) visite les nœuds dans l'ordre croissant de leurs étiquettes. Lors du parcours infixe de l'arbre T , il est visité d'abord son sous-arbre gauche, c'est à dire G , puis sa racine, et enfin son sous-arbre droit, D . Comme T est un arbre binaire de recherche, l'étiquette de sa racine est supérieure aux étiquettes de G et strictement inférieure aux étiquettes de D . La suite des étiquettes des nœuds parcourus dans T est donc croissante. \square

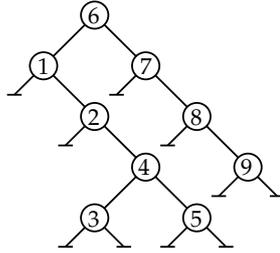
Nous pouvons maintenant définir l'étiquetage canonique d'un arbre binaire de recherche :

Définition 1.2.3. (Étiquetage canonique)

Soit $(T, \alpha, \Sigma, \leq)$ un arbre binaire de recherche. L'arbre T est *étiqueté de manière canonique* si $\Sigma = [1, n]$ où $n = \text{nd}(T)$ et \leq la relation d'ordre habituelle sur \mathbb{N} avec l'application α bijective.

Cet étiquetage est, comme son appellation le suggère, unique : en partant d'un arbre binaire non étiqueté, il existe une unique façon de l'étiqueter de sorte qu'il le soit de manière canonique. Dans la suite, même lorsque nous manipulerons des arbres binaires non étiquetés, nous allons utiliser l'étiquetage canonique pour désigner de manière non ambiguë un nœud par l'étiquette qui lui est correspondante si l'arbre était un arbre binaire étiqueté de manière canonique. Nous dirons que l'entier ainsi associée à un nœud est son *indice canonique*. Cela nous fourni, en plus des adresses, une seconde manière pour repérer un nœud dans un arbre binaire.

Exemple.



L'arbre $T =$ est un arbre binaire de recherche avec comme ensemble d'étiquettes $\Sigma = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, muni de la relation d'ordre habituelle sur les entiers. Il est de plus étiqueté canoniquement.

1.3 Les arbres binaires équilibrés et l'opération de rotation

Le concept d'arbre binaire équilibré et l'opération de rotation sont au centre de ce mémoire. Nous commençons par rappeler la définition d'arbre binaire équilibré ainsi qu'un théorème bien connu qui donne un encadrement pour la hauteur des arbres équilibrés en fonction de leur nombre de nœuds. Pour montrer ce théorème, nous avons besoin de considérer les arbres de Fibonacci. Nous définissons enfin l'opération de rotation sur les arbres binaire et la terminologie adjacente, utilisée dans la suite de ce mémoire.

1.3.1 Définition

Définition 1.3.1. (Arbre binaire équilibré)

Un *arbre binaire équilibré* est un arbre binaire T tel que pour tout $G \wedge D \in \text{SArbre}(T)$ nous avons $|\text{ht}(G) - \text{ht}(D)| \leq 1$.

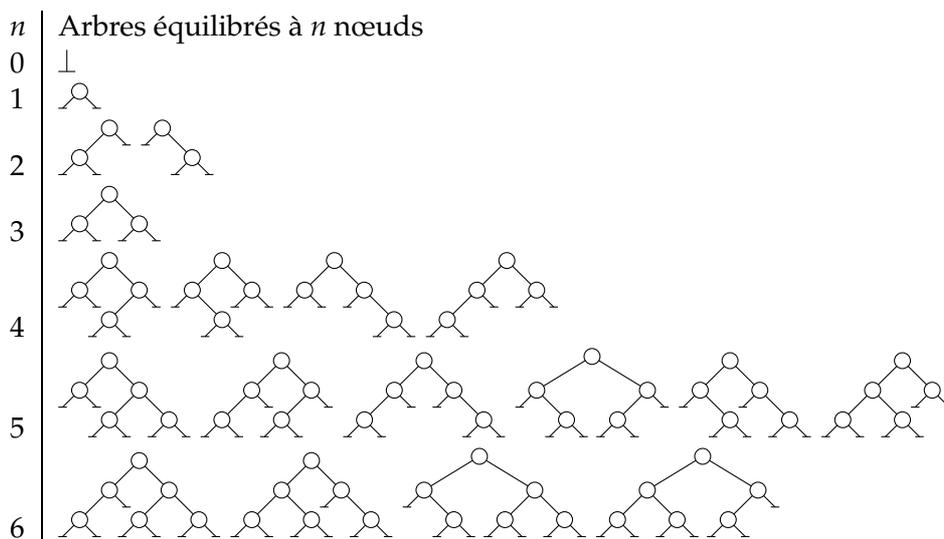


FIG. 1.3 – Les premiers arbres équilibrés.

Bien entendu, les trois types d'arbres que nous avons vu jusqu'à présent peuvent avoir la qualité d'être équilibrés. La propriété pour un arbre d'être équilibré joue un grand rôle en pratique. Il est en effet souhaitable de maintenir un arbre binaire de

recherche équilibré de manière à pouvoir accéder à chacun des nœuds rapidement depuis la racine. C'est à partir de cette idée que sont bâtis les arbres AVL[2].

Nous notons \mathcal{E}_n l'ensemble des arbres binaires équilibrés à n nœuds (voir Figure 1.3 qui illustre les premiers arbres équilibrés) et $\mathcal{E}_\infty = \cup_{n \geq 0} \mathcal{E}_n$ l'ensemble des arbres binaires équilibrés.

1.3.2 Quelques propriétés des arbres binaires équilibrés

Il existe une famille d'arbres binaires qui possède des propriétés remarquables :

Définition 1.3.2. (Arbre de Fibonacci)

Le i -ème arbre de Fibonacci $T_f(i)$ est un arbre binaire non étiqueté défini par la récurrence suivante :

$$T_f(i) = \begin{cases} \perp & \text{si } i = 0, \\ \perp \wedge \perp & \text{si } i = 1, \\ T_f(i-1) \wedge T_f(i-2) & \text{sinon.} \end{cases} \quad (1.31)$$

Ce n'est pas la définition habituelle des arbres de Fibonacci, ils sont en effet usuellement définis comme des arbres binaires étiquetés[12]. Néanmoins, pour l'utilisation que nous allons en faire, seule la forme de ces arbres nous intéresse, ce qui explique notre définition (voir Figure 1.4).

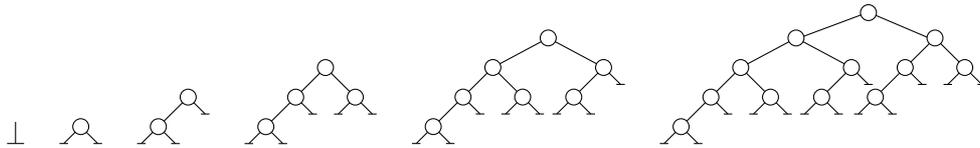


FIG. 1.4 – Les premiers arbres de Fibonacci.

Nous montrons maintenant une série de lemmes et une proposition pour dégager certaines propriétés de ces arbres dans le but d'établir le Théoreme 1.3.1.

Lemme 1.3.1. (Hauteur d'un arbre de Fibonacci)

Pour tout $i \geq 0$,

$$\text{ht}(T_f(i)) = i. \quad (1.32)$$

Preuve. Par récurrence sur i .

Si $i = 0$, alors $T_f(0) = \perp$ et on a bien $\text{ht}(\perp) = 0$.

Si $i = 1$, alors $T_f(1) = \perp \wedge \perp$ et on a bien $\text{ht}(\perp \wedge \perp) = 1$.

Supposons le lemme vrai pour tous les arbres $T_f(i)$ avec $i < k$ pour un entier $k \geq 2$ donné. Considérons l'arbre $T_f(k)$. nous avons

$$\text{ht}(T_f(k)) = \text{ht}(T_f(k-1) \wedge T_f(k-2)) \quad (\text{définition 1.3.2}) \quad (1.33)$$

$$= 1 + \max\{\text{ht}(T_f(k-1)), \text{ht}(T_f(k-2))\} \quad (\text{définition 1.1.4}) \quad (1.34)$$

$$= 1 + \max\{k-1, k-2\} \quad (\text{hypothèse de récurrence}) \quad (1.35)$$

$$= 1 + k - 1 \quad (1.36)$$

$$= k \quad (1.37)$$

ce qui montre le lemme. □

Proposition 1.3.1. (Équilibre des arbres de Fibonacci)

Pour tout $i \geq 0$, $T_f(i)$ est un arbre équilibré.

Preuve. Par récurrence sur i .

Si $i \in \{0, 1\}$, l'arbre $T_f(i)$ est bien équilibré.

Supposons la proposition vraie pour tous les arbres $T_f(i)$ avec $i < k$ pour un entier $k \geq 2$ donné. Considérons l'arbre $T_f(k)$. On a $T_f(k) = T_f(k-1) \wedge T_f(k-2)$ et d'après le Lemme 1.3.1, $\text{ht}(T_f(k-1)) = k-1$ et $\text{ht}(T_f(k-2)) = k-2$. De plus, par hypothèse de récurrence, les arbres $T_f(k-1)$ et $T_f(k-2)$ sont équilibrés. L'arbre $T_f(k)$ est donc équilibré. \square

Lemme 1.3.2. (Nombre de nœuds d'un arbre de Fibonacci)

Pour tout $i \geq 0$,

$$\text{nd}(T_f(i)) = F(i+2) - 1 \quad (1.38)$$

où $(F(i))_{i \geq 0}$ est la suite de Fibonacci, définie par $F(0) = 0$, $F(1) = 1$ et $F(i+2) = F(i+1) + F(i)$.

Preuve. Par récurrence sur i .

Si $i = 0$, alors $T_f(0) = \perp$ et $\text{nd}(\perp) = 0 = F(2) - 1$.

Si $i = 1$, alors $T_f(1) = \perp \wedge \perp$ et $\text{nd}(\perp \wedge \perp) = 1 = F(3) - 1$.

Supposons le lemme vrai pour tous les arbres $T_f(i)$ avec $i < k$ pour un entier $k \geq 2$ donné. Considérons l'arbre $T_f(k)$. Nous avons

$$\text{nd}(T_f(k)) = \text{nd}(T_f(k-1) \wedge T_f(k-2)) \quad (\text{définition 1.3.2}) \quad (1.39)$$

$$= 1 + \text{nd}(T_f(k-1)) + \text{nd}(T_f(k-2)) \quad (\text{définition 1.1.2}) \quad (1.40)$$

$$= 1 + F(k+1) - 1 + F(k) - 1 \quad (\text{hypothèse de récurrence}) \quad (1.41)$$

$$= F(k+2) - 1 \quad (1.42)$$

ce qui montre le lemme. \square

Il est facile de voir que les arbres de Fibonacci possèdent le moins de nœuds possible parmi tous les arbres équilibrés pour une hauteur donnée. Autrement dit, ce sont les arbres équilibrés qui possèdent la plus grande hauteur pour un nombre de nœud de la forme $F(i) - 1$ donné. Nous pouvons maintenant énoncer un théorème[12] qui donne un encadrement sur la hauteur d'un arbre équilibré en fonction du nombre de ses nœuds :

Théoreme 1.3.1. (Hauteur et nombre de nœuds d'un arbre équilibré)

Soit T un arbre équilibré. En posant $n = \text{nd}(T)$ et $h = \text{ht}(T)$, nous avons

$$\log_2(n+1) \leq h \leq C \log_2(n+2) + D \quad (1.43)$$

avec $C \approx 1.4404$ et $D \approx -0.3277$.

Preuve. D'après la Proposition 1.1.2, la borne inférieure de l'encadrement est vraie pour les arbres binaires en général, et donc aussi pour les arbres équilibrés.

Comme les arbres de Fibonacci sont les arbres binaires équilibrés qui possèdent la plus grande hauteur pour un nombre de nœuds minimal et, d'après le Lemme 1.3.2, $\text{nd}(T_f(h)) = F(h+2) - 1$, tout arbre binaire équilibré de hauteur h possède au minimum $F(h+2) - 1$ nœuds. Posons n le nombre de nœuds d'un arbre équilibré de hauteur h . Nous avons donc

$$n \geq F(h+2) - 1, \quad (1.44)$$

ce qui, par la formule de Binet donne

$$n \geq \frac{1}{\sqrt{5}}\phi^{h+2} - \frac{1}{\sqrt{5}}\bar{\phi}^{h+2} - 1, \quad (1.45)$$

avec $\phi = \frac{1+\sqrt{5}}{2}$ et $\bar{\phi} = \frac{1-\sqrt{5}}{2}$. Comme pour tout $h \geq 0$ la valeur de $\frac{1}{\sqrt{5}}\bar{\phi}^{h+2}$ est inférieure à 1, l'inégalité suivante, plus large, est établie :

$$n \geq \frac{1}{\sqrt{5}}\phi^{h+2} - 2 \quad (1.46)$$

On a donc :

$$\sqrt{5}(n+2) \geq \phi^{h+2} \quad (1.47)$$

ce qui, en passant par le logarithme en base ϕ , donne

$$\log_{\phi}(\sqrt{5}(n+2)) \geq h+2 \quad (1.48)$$

On obtient

$$\log_{\phi}(\sqrt{5}) + \log_{\phi}(n+2) - 2 \geq h \quad (1.49)$$

On a maintenant $\log_{\phi}(n+2) = C \log_2(n+2)$ avec $C = \frac{\ln(2)}{\ln(\phi)} \approx 1.4404$ et $\log_{\phi}(\sqrt{5}) - 2 = D \approx -0.3277$.

Nous avons donc bien $h \leq C \log_2(n+2) + D$. □

1.3.3 L'opération de rotation

L'opération de rotation est une opération qui transforme la forme d'un arbre binaire. Cette opération a été introduite historiquement pour maintenir les arbres binaires de recherche équilibrés[2] lors d'insertions et de suppressions successives de nœuds par les algorithmes de gestion d'ensembles dynamiques.

Définition 1.3.3. (Opération de rotation)

Soit T_0 un arbre binaire et $S_0 = (A \wedge B) \wedge C$ un de ses sous-arbres d'adresse u . Si T_1 est l'arbre binaire obtenu en remplaçant S_0 par $S_1 = A \wedge (B \wedge C)$ dans T_0 , on dit alors que T_1 est l'arbre obtenu à partir de T_0 par une *opération de rotation droite* d'adresse u (voir Figure 1.5).

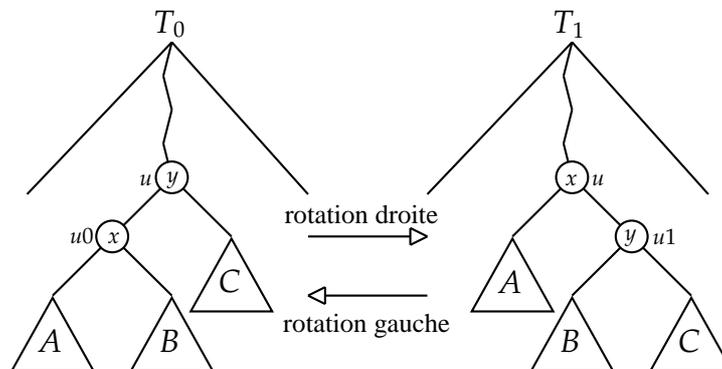


FIG. 1.5 – L'opération de rotation.

L'opération de rotation gauche est définie de manière analogue, c'est son opération inverse. Lorsque une rotation est appliquée à un arbre étiqueté, le nœud d'adresse u (resp. $u0$) dans T_0 a la même étiquette que le nœud d'adresse $u1$ (resp. u) dans T_1 conformément à la Figure 1.5. Remarquons que l'opération de rotation préserve l'ordre canonique des nœuds, ou autrement dit, si T_1 est l'arbre obtenu à partir d'une rotation depuis l'arbre T_0 , et si T_0 est étiqueté canoniquement, alors l'étiquetage de T_1 est encore canonique.

On note $\text{Rot}(T_0, u)$ l'arbre binaire obtenu par la rotation droite d'adresse u dans T_0 et

$$\text{Rot}(T_0) = \{T_1 \in \mathcal{A}_\infty \mid T_1 = \text{Rot}(T_0, u), u \in \{0, 1\}^*\} \quad (1.50)$$

l'ensemble des arbres que l'on peut obtenir par une rotation droite à partir de l'arbre T_0 . On dit de l'arbre $\text{SArbre}(T_0, u)$ qu'il est le *sous-arbre impliqué dans la rotation*, et on appelle l'objet (dans T_0) composé du nœud y et de son fils gauche, le nœud x , le *pivot initial* de la rotation. L'objet (dans T_1) composé du nœud x et de son fils droit, le nœud y , est appelé *pivot final* de la rotation. Il est clair que l'on peut effectuer une rotation droite dans un arbre T_0 d'adresse u seulement si $u0$ est l'adresse d'un nœud de T_0 et aussi que

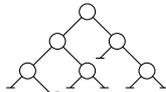
$$(\text{Rot}(T_0, u_1) = \text{Rot}(T_0, u_2)) \Rightarrow (u_1 = u_2). \quad (1.51)$$

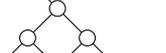
Nous avons donc

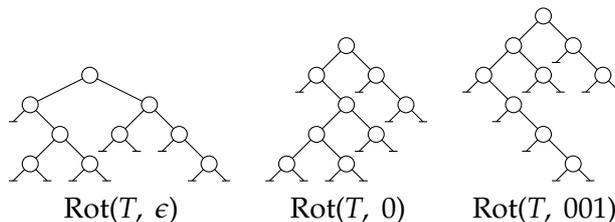
$$\#\text{Rot}(T_0) = \#\{u \in \{0, 1\}^* \mid u0 \in \mathbb{A}_N(T)\}. \quad (1.52)$$

Dans la suite, nous nous intéresserons plus particulièrement aux rotations droites et dirons donc simplement *rotation* pour parler de rotation droite.

Exemple.



Soit $T =$ . Comme $\mathbb{A}_N(T) = \{\epsilon, 0, 00, 001, 0010, 0011, 01, 1, 11\}$, cet arbre admet trois adresses de rotation possibles qui sont $\epsilon, 0$ et 001 . Les trois arbres résultants de ces rotations sont



Chapitre 2

Quelques codages des arbres binaires

Dans cette partie, nous allons étudier divers codages des arbres binaires non étiquetés. Le but de cette partie n'est pas de réaliser une liste exhaustive des objets combinatoires en bijection avec les arbres binaires[23, 24, 25] mais plutôt de donner diverses représentations qui se prêtent bien à certaines opérations sur les arbres comme la rotation, la greffe, les opérations / et \ qui peuvent avoir par ailleurs quelques propriétés intéressantes.

Nous commençons par étudier le codage par mot de Dyck, un des objets catalans les plus célèbres. Nous regardons ensuite un codage qui consiste à représenter un arbre binaire par les adresses de ses feuilles. Ce codage possède quelques propriétés notables et est utilisé par P. Dehornoy[10] pour montrer certaines propriétés sur les arbres et les rotations. En dernière partie, nous rappelons un codage beaucoup moins usuel[12, 13], codage que nous avons appelé *par vecteur de poids*. Ce codage des arbres binaires, en plus de permettre un calcul simple pour les opérations / et \ donne un moyen immédiat de dire si un premier arbre binaire peut être obtenu à partir d'un second en effectuant une suite de rotations.

2.1 Codage par mots de Dyck

Il est presque obligatoire de rappeler et de définir ce codage très connu des arbres binaires.

2.1.1 Définition

Rappelons ce qu'est un mot de Dyck :

Définition 2.1.1. (Mot de Dyck)

Un *mot de Dyck* u est un mot sur l'alphabet $\{a, b\}$ qui satisfait les deux conditions suivantes :

$$(C_1) \quad \forall v \in \text{Pref}(u), |v|_a \geq |v|_b ;$$

$$(C_2) \quad |u|_a = |u|_b.$$

On note \mathcal{D}_∞ l'ensemble des mots de Dyck et $\mathcal{D}_n = \{u \in \mathcal{D}_\infty \mid \frac{1}{2}|u| = n\}$ l'ensemble des mots de Dyck de demi-longueur n .

Il est usuel de représenter les mots de Dyck par un graphique : en partant de l'origine, et en lisant le mot de la gauche vers la droite, un *pas nord-est* est dessiné

à la lecture de la lettre a et un *pas sud-est* est dessiné à la lecture de la lettre b . Cette représentation est nommée *chemin de Dyck*.

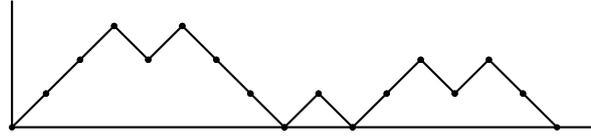


FIG. 2.1 – Représentation graphique du mot de Dyck $aaababbbabababb$.

La condition (C_1) de la définition 2.1.1 est respectée si le chemin reste au dessus de l'axe horizontal. La condition (C_2) quant à elle est satisfaite si l'extrémité droite du chemin se confond avec l'axe horizontal (voir Figure 2.1).

Posons à présent quelques notations :

- pour tout mot $u \in \{a, b\}^*$, on note $ht(u) = |u|_a - |u|_b$. En particulier, si $u \in \mathcal{D}_\infty$, alors $\forall v \in \text{Pref}(u)$, $ht(v) \geq 0$ et $ht(u) = 0$;
- pour tout mot $u \in \mathcal{D}_\infty$, on note $\text{praz}(u) = \min\{i \geq 1 \mid ht(u[1..i]) = 0\}$.

2.1.2 La bijection avec les arbres binaires

Nous avons besoin du lemme suivant, utile pour établir la bijection entre mots de Dyck et arbres binaires :

Lemme 2.1.1. (Décomposition unique d'un mot de Dyck)

Soit u un mot de Dyck. Il existe un unique couple $(v, w) \in \mathcal{D}_\infty \times \mathcal{D}_\infty$ tel que $u = avbw$.

Preuve. Posons $c = \text{praz}(u)$, $v = u[2..c-1]$ et $w = u[c+1..|u|]$. Commençons par montrer que les facteurs v et w de u sont bien des mots de Dyck. En ce qui concerne v :

$$ht(v) = ht(u[2..c-1]) \quad (2.1)$$

$$= ht(u[1..c]) - ht(u[1]) - ht(u[c]) \quad (2.2)$$

$$= 0 - ht(a) - ht(b) \quad (2.3)$$

$$= 0 \quad (2.4)$$

De plus, par hypothèse, nous avons $ht(u[1..i]) > 0$ pour $1 \leq i < c$. Les inégalités suivantes sont donc établies :

$$ht(u[1]) + ht(u[2..i]) > 0 \quad (2.5)$$

$$ht(a) + ht(u[2..i]) > 0 \quad (2.6)$$

$$1 + ht(u[2..i]) > 0 \quad (2.7)$$

$$ht(u[2..i]) \geq 0 \quad (2.8)$$

Le mot v respecte donc les conditions (C_1) et (C_2) . C'est donc un mot de Dyck. En ce qui concerne w :

$$ht(u) = ht(a) + ht(v) + ht(b) + ht(w) \quad (2.9)$$

$$= 1 + 0 - 1 + ht(w) \quad (2.10)$$

$$= ht(w) \quad (2.11)$$

et on en déduit $\text{ht}(w) = 0$ et donc w respecte la condition (C_2) .

De plus, par la condition (C_1) que respecte le mot u , on a pour $c + 1 \leq i \leq |u|$:

$$\text{ht}(avb) + \text{ht}(u[c + 1 .. i]) \geq 0 \quad (2.12)$$

$$\text{ht}(u[c + 1 .. i]) \geq 0, \quad (2.13)$$

qui montre que w respecte aussi la condition (C_1) et est ainsi un mot de Dyck.

Pour montrer l'unicité du couple (v, w) , supposons par l'absurde qu'il existe des mots de Dyck v' et w' tels que $u = av'bw'$ avec $(v', w') \neq (v, w)$. Supposons en premier lieu $v' \in \text{Pref}(v) \setminus \{v\}$. Comme v' est de Dyck, on a $\text{ht}(av'b) = 0$. On a donc $\text{ht}(u[1 .. |v'| + 2]) = 0$ mais ceci est contradictoire avec la minimalité de l'entier c car comme $|v| > |v'|$, on a $c > |v'| + 2$.

Supposons maintenant $v \in \text{Pref}(v') \setminus \{v'\}$. On a donc $vb \in \text{Pref}(v')$ et comme $\text{ht}(vb) = \text{ht}(v) + \text{ht}(b) = 0 - 1 = -1$, il existe un préfixe de v' de hauteur strictement négative, ce qui montre que v' n'est pas un mot de Dyck car il ne respecte pas la condition (C_1) .

Le couple (v, w) existe et est unique. \square

Nous pouvons maintenant énoncer la bijection entre arbres binaires et mots de Dyck :

Proposition 2.1.1. (Bijection entre arbres binaires et mots de Dyck)

L'application $D : \mathcal{A}_n \rightarrow \mathcal{D}_n$ définie par

$$D(T) = \begin{cases} \epsilon & \text{si } T = \perp, \\ a D(G) b D(D) & \text{pour } T = G \wedge D \text{ sinon,} \end{cases} \quad (2.14)$$

est une bijection.

Son inverse est l'application $D^{-1} : \mathcal{D}_n \rightarrow \mathcal{A}_n$ définie par

$$D^{-1}(u) = \begin{cases} \perp & \text{si } u = \epsilon, \\ D^{-1}(v) \wedge D^{-1}(w) & \text{pour } u = avbw \text{ avec } (v, w) \in \mathcal{D}_\infty \times \mathcal{D}_\infty \text{ sinon.} \end{cases} \quad (2.15)$$

Preuve. Montrons d'abord que pour tout arbre binaire T , on a $D^{-1}(D(T)) = T$ par induction structurelle sur les arbres.

Si $T = \perp$, on a clairement $D^{-1}(D(\perp)) = \perp$.

Supposons la proposition vraie pour des arbres G et D quelconques et considérons l'arbre $T = G \wedge D$. On a alors

$$D^{-1}(D(T)) = D^{-1}(D(G \wedge D)) \quad (2.16)$$

$$= D^{-1}(a D(G) b D(D)) \quad (2.17)$$

$$= D^{-1}(D(G)) \wedge D^{-1}(D(D)) \quad (2.18)$$

$$= G \wedge D \quad (2.19)$$

$$= T. \quad (2.20)$$

Réciproquement, montrons que pour tout mot de Dyck u , on a $D(D^{-1}(u)) = u$ par récurrence sur la demi longueur de u .

Si $u = \epsilon$, on a clairement $D(D^{-1}(\epsilon)) = \epsilon$.

Supposons la proposition vraie pour tout mot de Dyck u tel que $\frac{1}{2}|u| \leq k$. Soit u un mot de Dyck tel que $\frac{1}{2}|u| = k + 1$. D'après le Lemme 2.1.1, ce mot s'écrit de manière

unique sous la forme $u = avbw$ avec $v, w \in \mathcal{D}_\infty$. Comme $\frac{1}{2}|v| \leq k$ et $\frac{1}{2}|w| \leq k$ on peut appliquer l'hypothèse de récurrence à v et w . On a alors

$$D(D^{-1}(u)) = D(D^{-1}(avbw)) \quad (2.21)$$

$$= D(D^{-1}(v) \wedge D^{-1}(w)) \quad (2.22)$$

$$= a D(D^{-1}(v)) b D(D^{-1}(w)) \quad (2.23)$$

$$= avbw \quad (2.24)$$

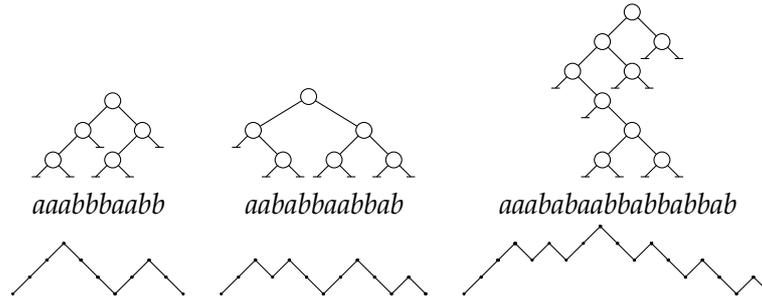
$$= u. \quad (2.25)$$

Les applications D et D^{-1} sont donc bien des bijections, l'une inverse de l'autre. \square

Cette bijection peut se réaliser comme suit : calculer le mot de Dyck correspondant à un arbre binaire revient à réaliser un parcours préfixe de l'arbre en écrivant un a lorsque un nœud est parcouru et un b lorsque c'est une feuille. Le dernier b correspondant à la dernière feuille est omis. La construction inverse est immédiate. La Figure 2.6 illustre la bijection entre les arbres binaires à 4 nœuds et les mots de Dyck.

Exemple.

Voici quelques arbres binaires et leur mot et chemin de Dyck correspondant :



2.1.3 Comportements avec les opérations des arbres binaires

Regardons la manière dont les opérations que nous avons définies sur les arbres se comportent avec les mots de Dyck. Pour les trois propositions suivantes, nous donnons des preuves intuitives.

Proposition 2.1.2. (Opération de greffe et mots de Dyck)

Si T_0 et T_1 sont deux arbres binaires, alors

$$D(T_0 \wedge T_1) = a D(T_0) b D(T_1). \quad (2.26)$$

Preuve. Clair d'après la Proposition 2.1.1. \square

Proposition 2.1.3. (Opération / et mots de Dyck)

Si T_0 et T_1 sont deux arbres binaires, alors

$$D(T_0/T_1) = a^k D(T_0) (a^k)^{-1} \cdot D(T_1) \quad (2.27)$$

avec $a^k b \in \text{Pref}(D(T_1))$ ou $k = 0$ si $T_1 = \perp$.

Preuve. Lors du parcours préfixe de l'arbre T_0/T_1 , l'arbre T_0 est parcouru lorsque une feuille est rencontrée pour la première fois dans T_1 . En effet, la racine de l'arbre T_0 est greffée à la place de la feuille la plus à gauche de l'arbre T_1 . C'est le premier b dans le mot $D(T_1)$ qui symbolise cette feuille de T_1 . Le mot $D(T_0)$ est donc inséré à la place de la première occurrence de ce b . De plus, un dernier b est rajouté à la fin de l'insertion pour marquer le parcours de la dernière feuille de T_0 qui n'existe pas dans le mots $D(T_0)$. \square

Proposition 2.1.4. (Opération \setminus et mots de Dyck)

Si T_0 et T_1 sont deux arbres binaires, alors

$$D(T_0 \setminus T_1) = D(T_0)D(T_1). \quad (2.28)$$

Preuve. Il est facile de voir que lors du parcours préfixe de l'arbre $T_0 \setminus T_1$, l'arbre T_0 est parcouru, puis, comme la racine de T_1 se trouve à la place de la feuille la plus à droite de T_0 , l'arbre T_1 est ensuite parcouru. \square

Pour caractériser l'opération de rotation sur les mots de Dyck, nous avons besoin du lemme suivant qui donne une condition nécessaire et suffisante pour décider, à partir d'un mot de Dyck u , si l'arbre $D^{-1}(v)$ où $v \in \text{Fact}(u)$ est un sous-arbre de l'arbre $D^{-1}(u)$.

Lemme 2.1.2. (Facteurs d'un mot de Dyck qui correspondent à des sous-arbres)

Soient T un arbre binaire, $u = D(T)$ le mot de Dyck qui lui correspond et

$$L = \{v \in \mathcal{D}_\infty \mid v \in \text{Suff}(u) \text{ ou } vb \in \text{Fact}(u)\}. \quad (2.29)$$

On a alors $v \in L$ si et seulement si il existe un sous-arbre S de T tel que $D(S) = v$.

Preuve. (\Rightarrow) : soit $v \in L$. Si $v \in \text{Suff}(u)$ alors $u = wv$ avec $w \in \mathcal{D}_\infty$. D'après la Proposition 2.1.4, nous avons $D^{-1}(u) = D^{-1}(w) \setminus D^{-1}(v)$, ce qui montre que $D^{-1}(v)$ est bien un sous-arbre de $D^{-1}(u)$. Si $vb \in \text{Fact}(u)$, alors d'après la Proposition 2.1.1 on a bien $D^{-1}(v)$ sous-arbre de $D^{-1}(u)$.

(\Leftarrow) : par induction structurelle sur les arbres binaires.

Si $T = \perp$, alors $L = \{\epsilon\}$ ce qui ne contredit pas le lemme puisque T est son seul sous-arbre.

Supposons l'implication vraie pour deux arbres G et D et considérons l'arbre $T = G \wedge D$. Cet arbre possède comme sous-arbres ceux de G , ceux de D et lui-même, T . Nous avons $D(T) = au_Gbu_D$ avec $u_G = D(G)$ et $u_D = D(D)$. Par hypothèse de récurrence, les mots de Dyck qui sont suffixes de u_D et facteurs de u_D suivis d'un b correspondent aux sous-arbres de D . Ces mots appartiennent clairement au langage L . De même, par hypothèse de récurrence, les mots de Dyck qui sont suffixes de u_G et facteurs de u_G suivis d'un b correspondent aux sous-arbres de G . Ces mots appartiennent au langage L car ceux qui étaient suffixes de u_G sont maintenant facteurs de au_Gb et bien suivis d'un b , et ceux qui étaient facteurs de u_G et suivis d'un b le sont encore dans au_Gbu_D . \square

Proposition 2.1.5. (Opération de rotation et mots de Dyck)

Soit T_0 un arbre binaire et T_1 l'arbre binaire obtenu par une rotation à partir de T_0 . Il existe alors un mot

$$v \in \{v \in \mathcal{D}_\infty \mid v \in \text{Suff}(D(T_0)) \text{ ou } vb \in \text{Fact}(D(T_0))\} \quad (2.30)$$

tel que $v = aaAbBbC$ et

$$D(T_0) = XaaAbBbCY \text{ et } D(T_1) = XaAbaBbCY \quad (2.31)$$

avec $A, B, C, X, Y \in \mathcal{D}_\infty$.

Preuve. D'après le Lemme 2.1.2, le facteur v de $D(T_0)$ correspond à un de ses sous-arbres. C'est aussi le cas pour les facteurs A, B et C . Comme

$$D^{-1}(aaAbBbC) = (D^{-1}(A) \wedge D^{-1}(B)) \wedge D^{-1}(C) \quad (2.32)$$

et

$$D^{-1}(aAbaBbC) = D^{-1}(A) \wedge (D^{-1}(B) \wedge D^{-1}(C)), \quad (2.33)$$

et que le facteur $aAbaBbC$ de $D(T_1)$ correspond à l'un de ses sous-arbres, il existe bien un sous-arbre de T_0 modifié dans T_1 par, ailleurs identique, conformément à la définition 1.3.3. \square

2.2 Codage par langages des feuilles

Dans le chapitre précédent, nous avons défini le langage $\mathbb{A}_F(T)$ comme étant le langage contenant les adresses des feuilles de l'arbre T . Nous prouvons maintenant que $\mathbb{A}_F(T)$ définit un codage de l'arbre T et examinons le comportement de ce codage sous les diverses opérations sur les arbres binaires.

2.2.1 Classe des langages des feuilles

Voyons sous quelles conditions un langage de mots sur l'alphabet $\{0, 1\}$ correspond à un arbre binaire. Donnons une première caractéristique de ce langage :

Lemme 2.2.1. (Première condition du langage des feuilles)

Soit T un arbre binaire.

Si $u_1, u_2 \in \mathbb{A}_F(T)$ et $u_1 \in \text{Pref}(u_2)$, alors $u_1 = u_2$.

Preuve. En supposant par l'absurde qu'il existe des adresses de feuilles $u_1, u_2 \in \mathbb{A}_F(T)$ telles que $u_1 \in \text{Pref}(u_2) \setminus \{u_2\}$, par induction, il existe un nœud x dans T d'adresse $u_1.a$ avec $a \in \{0, 1\}$. Ceci implique que la feuille d'adresse u_1 est père du nœud x , ce qui est absurde car une feuille n'a pas de descendance. \square

Cette condition n'est bien entendu pas suffisante car, par exemple, en prenant le langage $L = \{0, 10\}$, il est facile de voir qu'il est impossible de construire un arbre binaire qui admet les mots de ce langage comme adresses de ses feuilles. Donnons une seconde caractéristique de $\mathbb{A}_F(T)$:

Lemme 2.2.2. (Seconde condition du langage des feuilles)

Soit T un arbre binaire.

Si $uav \in \mathbb{A}_F(T)$ alors il existe $ubw \in \mathbb{A}_F(T)$ avec $u, v, w \in \{0, 1\}^*$, $a, b \in \{0, 1\}$ et $a \neq b$.

Preuve. Par l'absurde, et sans perte de généralité, supposons qu'il existe un mot $u0v \in \mathbb{A}_F(T)$ et que $u1 \notin \text{Pref}(\mathbb{A}_F(T))$. À l'adresse u correspond un nœud x . La feuille d'adresse $u0v$ fait partie de la descendance du fils gauche du nœud x . Maintenant, x possède obligatoirement un fils droit. Si ce fils droit est vide, alors $u1 \in \mathbb{A}_F(T)$, ce qui est contradictoire et s'il n'est pas vide, il contient au moins une feuille d'adresse préfixée par $u1$, ce qui est également contradictoire. \square

Ces deux caractérisations sont suffisantes pour décrire la classe des langages des feuilles :

Proposition 2.2.1. (Caractérisation des langages des feuilles)

Pour tout langage $L \subseteq \{0, 1\}^*$ tel que

- (i) si $u_1, u_2 \in L$ et $u_1 \in \text{Pref}(u_2)$, alors $u_1 = u_2$;
- (ii) si $uav \in L$ alors il existe $ubw \in L$ avec $u, v, w \in \{0, 1\}^*$, $a, b \in \{0, 1\}$ et $a \neq b$,

il existe un arbre binaire T tel que $\mathbb{A}_F(T) = L$.

Et réciproquement, pour tout arbre binaire T , le langage $L = \mathbb{A}_F(T)$ satisfait les deux conditions (i) et (ii).

Preuve. (\Leftarrow) : d'après le Lemme 2.2.1 et le Lemme 2.2.2, si T est un arbre binaire, le langage $\mathbb{A}_F(T)$ satisfait les deux conditions (i) et (ii).

(\Rightarrow) : réciproquement, supposons que L est un langage qui respecte les deux conditions (i) et (ii). Montrons comment nous pouvons construire un unique arbre binaire T tel que $\mathbb{A}_F(T) = L$. Pour les besoins de cette preuve, l'arbre T est étiqueté sur l'ensemble $\Sigma = \{N, F\}$. L'arbre T est initialement composé d'un unique nœud d'étiquette F . Pour chacun des mots u de L , on suit le chemin d'adresse $u.\{0, 1\}^{-1}$ dans T . Si ce chemin n'existe pas, on crée les nœuds nécessaires, étiquetés par N de façon à ce qu'il existe. On crée ensuite un nœud d'adresse u , étiqueté par F . Montrons que l'objet T' obtenu en supprimant les feuilles de T et en remplaçant les nœuds étiquetés par F par des feuilles est bien un arbre binaire. D'après le Lemme 2.2.2, tout nœud x de T étiqueté par N possède exactement deux fils. En effet, en supposant que l'adresse de x est v , il existe un mot $u \in L$ tel que $u = vau'$ pour $u' \in \{0, 1\}^*$ et $a \in \{0, 1\}$, et donc aussi une feuille d'adresse vbu'' pour $u'' \in \{0, 1\}^*$, $b \in \{0, 1\}$ et $b \neq a$. Ceci montre aussi que pour toute feuille f de T' il existe $w \in L$ tel que w est l'adresse de f . De plus, par le Lemme 2.2.1, nous savons que tout nœud d'étiquette F dans T ne possède pas de descendance. \square

Étant donné un arbre binaire T , nous appellerons maintenant le langage $\mathbb{A}_F(T)$ le langage des feuilles de T . Clairement, d'après la Proposition 1.1.1, T possède $n + 1$ feuilles, et donc

$$\#\mathbb{A}_F(T) = \text{nd}(T) + 1. \quad (2.34)$$

La Figure 2.6 illustre la bijection entre les arbres binaires à 4 nœuds et les langages des feuilles.

2.2.2 Comportements avec les opérations des arbres binaires

Étudions la manière dont la rotation modifie le langage des feuilles d'un arbre.

Proposition 2.2.2. (Opération de rotation et langages des feuilles)

Soient T_0 et T_1 des arbres binaires et u une adresse de rotation tels que $T_1 = \text{Rot}(T_0, u)$. Alors $u00 \in \text{Pref}(\mathbb{A}_F(T_0))$ et le langage $\mathbb{A}_F(T_1)$ est obtenu à partir du langage $\mathbb{A}_F(T_0)$ en effectuant les substitutions suivantes :

$$u00A \longrightarrow u0A \quad (2.35)$$

$$u01B \longrightarrow u10B \quad (2.36)$$

$$u1C \longrightarrow u11C \quad (2.37)$$

pour tout $A, B, C \in \{0, 1\}^*$.

Preuve. Tout d'abord, remarquons que la condition $u00 \in \text{Pref}(\mathbb{A}_F(T))$ est nécessaire parce que, le cas échéant, le nœud d'adresse u aurait l'arbre vide comme fils gauche, et donc la rotation à cette adresse serait impossible.

Posons maintenant $S_0 = \text{SArbre}(T_0, u) = (T_A \wedge T_B) \wedge T_C$. D'après la définition 1.3.3, ce sous-arbre se retrouve modifié en l'arbre $S_1 = T_A \wedge (T_B \wedge T_C)$ dans l'arbre T_1 . En se reportant à la Figure 1.5, il est possible de voir graphiquement que :

- les feuilles de T_A sont d'adresses préfixées par $u00$ dans T_0 et deviennent préfixées par $u0$ dans T_1 ,
- les feuilles de T_B sont d'adresses préfixées par $u01$ dans T_0 et deviennent préfixées par $u10$ dans T_1 ,
- les feuilles de T_C sont d'adresses préfixées par $u1$ dans T_0 et deviennent préfixées par $u11$ dans T_1 .

De plus les feuilles d'adresses non préfixées par u dans T_0 gardent la même adresse dans T_1 . Ceci montre que la substitution est valide. \square

L'avantage de ce codage par langage des feuilles, contrairement au codage par mots de Dyck est clair dans le contexte de l'opération de rotation : il est facile de calculer l'arbre T_1 résultant d'une opération de rotation depuis l'arbre T_0 étant donnée une adresse de rotation u .

2.3 Codage par vecteurs de poids

Le codage par langages des feuilles présente le désavantage majeur, étant donné deux langages L_0 et L_1 , qu'il est difficile de dire si l'arbre correspondant à L_1 peut être obtenu par une suite de rotations à partir de l'arbre correspondant à L_0 . Nous définissons ici un codage adapté à ce problème de décision.

2.3.1 Définition

Définition 2.3.1. (Vecteur de poids)

Un vecteur de poids v de longueur n est une suite d'entiers tels que :

$$(C_1) \quad \forall 1 \leq i \leq n, 1 \leq v[i] \leq n - i + 1;$$

$$(C_2) \quad \forall 1 \leq i \leq n, \forall 1 \leq r \leq v[i] - 1, v[i+r] \leq v[i] - r.$$

Tout comme les mots de Dyck, les vecteurs de poids possèdent leur représentation graphique : pour chaque composante du vecteur $v[i]$, traité de la gauche vers la droite, on dessine une colonne de $v[i]$ petits carrés (voir Figure 2.2).

De manière visuelle et intuitive, en considérant la représentation graphique d'un vecteur de poids, la condition (C_1) signifie que la représentation est contenue dans

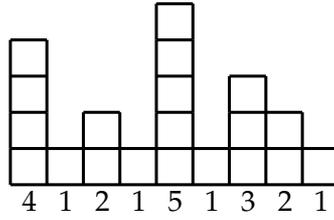


FIG. 2.2 – Représentation graphique du vecteur de poids 412151321.

le triangle rectangle de hauteur n et de longueur n , et la conditions (C_2) traduit le fait que si l'on choisit une colonne, en considérant le triangle rectangle formé par cette colonne à droite, aucune colonne ne dépasse de ce triangle.

S'il n'y a pas d'ambiguïté, les vecteurs de poids seront notés en juxtaposant simplement les composantes et en omettant les virgules. Nous allons en effet voir les vecteurs de poids simplement comme des mots sur l'alphabet $\mathbb{N} \setminus \{0\}$.

On note \mathcal{V}_n l'ensemble des vecteurs de poids de longueur n et $\mathcal{V}_\infty = \cup_{n \geq 0} \mathcal{V}_n$ l'ensemble des vecteurs de poids de longueur quelconque.

2.3.2 La bijection avec les arbres binaires

De manière informelle, ce codage revient à effectuer un parcours infixe d'un arbre binaire en inscrivant pour chaque nœud visité le nombre de nœuds présents dans son sous-arbre droit, incrémenté de 1. Plus formellement :

Proposition 2.3.1. (Bijection entre arbres binaires et vecteurs de poids)

L'application $V : \mathcal{A}_n \rightarrow \mathcal{V}_n$ définie par

$$V(T) = \begin{cases} \epsilon & \text{si } T = \perp, \\ V(G).nd(D) + 1.V(D) & \text{pour } T = G \wedge D \text{ sinon,} \end{cases} \quad (2.38)$$

est une bijection.

Son inverse est l'application $V^{-1} : \mathcal{V}_n \rightarrow \mathcal{A}_n$ définie par

$$V^{-1}(v) = \begin{cases} \perp & \text{si } v = \epsilon, \\ V^{-1}(g) \wedge V^{-1}(d) & \text{pour } v = gad \text{ avec } a = |d| + 1 \text{ et } |g| \text{ minimal.} \end{cases} \quad (2.39)$$

Preuve. Montrons $V^{-1}(V(T)) = T$ pour tout arbre binaire T par induction structurelle sur les arbres binaires.

Si $T = \perp$, $V^{-1}(V(\perp)) = V^{-1}(\epsilon) = \perp$.

Supposons la proposition vraie pour deux arbres binaires G et D et montrons qu'elle est encore vraie pour l'arbre $T = G \wedge D$. Nous avons

$$V^{-1}(V(T)) = V^{-1}(V(G \wedge D)) \quad (2.40)$$

$$= V^{-1}(V(G).nd(D) + 1.V(D)) \quad (2.41)$$

Maintenant, comme $g = V(G)$ est par hypothèse d'induction un vecteur de poids, toutes ses composantes sont inférieures à $nd(G)$. La composante $v[i]$ du vecteur de poids

$$v = V(G).nd(D) + 1.V(D) \quad (2.42)$$

tel que $v[i] = \text{nd}(T) - i + 1$ avec i minimal est donc $v[\text{nd}(G) + 1] = \text{nd}(D) + 1$. On a donc

$$V^{-1}(V(G). \text{nd}(D) + 1. V(D)) = V^{-1}(V(G)) \wedge V^{-1}(V(D)) \quad (2.43)$$

$$= G \wedge D \quad (2.44)$$

$$= T \quad (2.45)$$

Montrons maintenant $V(V^{-1}(v)) = v$ pour tout vecteur de poids v par récurrence sur la longueur de v .

Si $v = \epsilon$, $V(V^{-1}(\epsilon)) = V(\perp) = \epsilon$.

Supposons la proposition vraie pour tout vecteur de poids de longueur inférieure à n et considérons le vecteur de poids v de longueur $n + 1$. Posons $v = gad$ avec $a = |d| + 1$ et g de longueur minimale. Nous avons

$$V(V^{-1}(v)) = V(V^{-1}(gad)) \quad (2.46)$$

$$= V(V^{-1}(g) \wedge V^{-1}(d)) \quad (2.47)$$

$$= V(V^{-1}(g)). \text{nd}(V(V^{-1}(d))) + 1. V(V^{-1}(d)) \quad (2.48)$$

$$= g. |d| + 1. d \quad (2.49)$$

$$= gad \quad (2.50)$$

$$= v \quad (2.51)$$

Les applications V et V^{-1} sont donc des bijections, l'une inverse de l'autre. \square

La Figure 2.6 illustre la bijection entre les arbres binaires à 4 nœuds et les vecteurs de poids.

2.3.3 Comportements avec les opérations des arbres binaires

Analysons les impacts des opérations \wedge , $/$ et \setminus sur les vecteurs de poids. À notre connaissance, les opérations \wedge , $/$ et \setminus n'ont pas encore été étudiées à travers le codage par vecteur de poids. Nous établissons donc ici des résultats faciles mais nouveaux.

Proposition 2.3.2. (Opération de greffe et vecteurs de poids)

Si T_0 et T_1 sont deux arbres binaires, alors

$$V(T_0 \wedge T_1) = V(T_0). \text{nd}(T_1) + 1. V(T_1). \quad (2.52)$$

Preuve. Clair d'après la Proposition 2.3.1. \square

La Figure 2.3 illustre l'opération \wedge sur les vecteurs de poids de longueur 3.

\wedge	111	121	211	311	321
111	1114111	1144121	1144211	1144311	1144321
121	1214111	1214121	1214211	1214311	1214321
211	2114111	2114121	2114211	2114311	2114321
311	3114111	3114121	3114211	3114311	3114321
321	3214111	3214121	3214211	3214311	3214321

FIG. 2.3 – L'opération \wedge sur les vecteurs de poids de \mathcal{V}_3 .

Proposition 2.3.3. (Opération / et vecteurs de poids)

Si T_0 et T_1 sont deux arbres binaires, alors

$$V(T_0/T_1) = V(T_0) \cdot V(T_1). \quad (2.53)$$

Preuve. Par induction structurelle sur l'arbre T_1 .

Si $T_1 = \perp$, la proposition est clairement vraie.

Supposons la proposition vraie pour $T_1 = G$ et $T_1 = D$ avec G et D deux arbres binaires et montrons qu'elle reste vraie pour $T_1 = G \wedge D$. Calculons

$$V(T_0/T_1) = V(T_0/(G \wedge D)) \quad (2.54)$$

$$= V((T_0/G) \wedge D) \text{ (par l'Équation 1.24)} \quad (2.55)$$

$$= V(T_0/G) \cdot \text{nd}(D) + 1 \cdot V(D) \text{ (par la Proposition 2.3.2)} \quad (2.56)$$

$$= V(T_0) \cdot V(G) \cdot \text{nd}(D) + 1 \cdot V(D) \text{ (par hypothèse d'induction)} \quad (2.57)$$

$$= V(T_0) \cdot V(G \wedge D) \quad (2.58)$$

$$= V(T_0) \cdot V(T_1) \quad (2.59)$$

ce qui prouve la proposition. \square

La Figure 2.4 illustre l'opération / sur les vecteurs de poids de longueur 3.

/	111	121	211	311	321
111	111111	111121	111211	111311	111321
121	121111	121121	121211	121311	121321
211	211111	211121	211211	211311	211321
311	311111	311121	311211	311311	311321
321	321111	321121	321211	321311	321321

FIG. 2.4 – L'opération / sur les vecteur de poids de \mathcal{V}_3 .

Proposition 2.3.4. (Opération \ et vecteurs de poids)

Si T_0 et T_1 sont deux arbres binaires, alors $V(T_0 \setminus T_1) = v \cdot V(T_1)$ où v est le vecteur de poids défini pour $1 \leq i \leq \text{nd}(T_0)$ par :

$$v[i] = \begin{cases} V(T_0)[i] + \text{nd}(T_1) & \text{si } V(T_0)[i] = \text{nd}(T_0) - i + 1, \\ V(T_0)[i] & \text{sinon.} \end{cases} \quad (2.60)$$

Preuve. Par induction structurelle sur l'arbre T_0 .

Si $T_0 = \perp$, la proposition est clairement vraie.

Supposons la proposition vraie pour $T_0 = G$ et $T_0 = D$ avec G et D deux arbres binaires et montrons qu'elle reste vraie pour $T_0 = G \wedge D$. Calculons

$$V(T_0 \setminus T_1) = V((G \wedge D) \setminus T_1) \quad (2.61)$$

$$= V(G \wedge (D \setminus T_1)) \text{ (par l'Équation 1.25)} \quad (2.62)$$

$$= V(G) \cdot \text{nd}(D \setminus T_1) + 1 \cdot V(D \setminus T_1) \text{ (par la Proposition 2.3.2)} \quad (2.63)$$

$$= V(G) \cdot \text{nd}(D) + 1 + \text{nd}(T_1) \cdot v \cdot V(T_1) \text{ (par hypothèse d'induction)} \quad (2.64)$$

$$\text{avec } v \text{ le vecteur de poids qui répond aux conditions} \quad (2.65)$$

$$\text{de la proposition appliquée pour } D \setminus T_1. \quad (2.66)$$

$$= v' \cdot V(T_1) \quad (2.67)$$

$$\text{avec } v' = V(G) \cdot \text{nd}(D) + 1 + \text{nd}(T_1) \cdot v. \quad (2.68)$$

Le vecteur v' respecte les conditions de la proposition. Elle est donc prouvée. \square

La Figure 2.5 illustre l'opération \setminus sur les vecteurs de poids de longueur 3.

\setminus	111	121	211	311	321
111	114111	114121	114211	114311	114321
121	154111	154121	154211	154311	154321
211	214111	214121	214211	214311	214321
311	614111	614121	614211	614311	614321
321	654111	654121	654211	654311	654321

FIG. 2.5 – L'opération \setminus sur les vecteur de poids de \mathcal{V}_3 .

L'opération $/$ avec le codage par vecteur de poids et l'opération \setminus avec le codage par mot de Dyck se réalisent, d'après la Proposition 2.3.3 et la Proposition 2.1.4 par des simples concaténations de mots. En revanche, comme nous l'avons vu par la Proposition 2.3.4 et la Proposition 2.1.3, l'opération \setminus en terme de vecteurs de poids et l'opération $/$ en terme de mots de Dyck sont plus délicates et moins naturelles.

Intéressons-nous maintenant à l'opération de rotation et donnons une condition nécessaire et suffisante pour décider si $T_1 \in \text{Rot}(T_0)$ étant donné deux arbres binaires T_1 et T_0 .

Lemme 2.3.1. (Facteurs d'un vecteur de poids qui correspondent à des sous-arbres) Soit T un arbre binaire et v le vecteur de poids qui lui est associé. Le vecteur de poids correspondant au sous-arbre de T de racine i pour l'étiquetage canonique est $u = p.v[i].s$ où :

- (i) $u \in \text{Fact}(v)$,
- (ii) $|s| = v[i] - 1$ et
- (iii) p est le plus long mot qui vérifie $p[j] < |u| - j + 1$ pour $1 \leq j \leq |p|$.

Preuve. Clair d'après la bijection établie dans la Proposition 2.3.1. \square

Proposition 2.3.5. (Opération de rotation et vecteurs de poids)

Soient T_0 et T_1 des arbres binaires à n nœuds.

$T_1 \in \text{Rot}(T_0)$ si et seulement si $\text{ham}(V(T_0), V(T_1)) = 1$ et il existe un entier i tel que $1 \leq i \leq n$ et $V(T_1)[i] > V(T_0)[i]$ avec $V(T_1)[i]$ minimal.

Preuve. (\Rightarrow) : supposons $T_1 \in \text{Rot}(T_0)$. D'après la définition 1.3.3, il existe un sous-arbre $S_0 = (A \wedge B) \wedge C$ dans T_0 qui se retrouve sous la forme $S_1 = A \wedge (B \wedge C)$ dans T_1 par ailleurs identique. En appliquant le codage, nous obtenons

$$V(S_0) = V(A).nd(B) + 1.V(B).nd(C) + 1.V(C) \quad (2.69)$$

et

$$V(S_1) = V(A).nd(B) + nd(C) + 2.V(B).nd(C) + 1.V(C). \quad (2.70)$$

Nous avons donc $\text{ham}(V(S_0), V(S_1)) = 1$ avec exactement une composante de $V(S_1)$ strictement supérieure à la composante de $V(S_0)$ qui occupe la même position. Maintenant, comme $nd(S_0) = nd(S_1)$ et que les arbres T_0 et T_1 ne sont différents que localement pour les sous-arbres S_0 et S_1 , si $V(T_0) = u.V(S_0).v$, alors $V(T_1) = u.V(S_1).v$ avec u et v des mots sur \mathbb{N} .

(\Leftarrow) : posons $v_0 = V(T_0)$ et $v_1 = V(T_1)$. Considérons le sous-arbre de T_0 de racine i

2.3. CODAGE PAR VECTEURS DE POIDS

pour l'étiquetage canonique et posons-le égal à $A \wedge B$ pour deux arbres binaires A et B . Les arbres A et B sont également des sous-arbres de T_0 . D'après le Lemme 2.3.1, ils correspondent à des facteurs de v_0 . On note v_A le facteur de v_0 qui correspond au sous-arbre A et v_B le facteur de v_0 qui correspond au sous-arbre B . Nous avons donc $v_0 = u.v_A.v_0[i].v_B.w$ avec u et w des mots sur \mathbb{N} . Comme $v_1[i] > v_0[i]$, on a $v_0[i] < |v_0| - i + 1$. Le mot w n'est donc pas vide et il existe un facteur v_C , pouvant être vide, qui correspond à un sous-arbre de T_0 tel que $v_0 = u.v_A.v_0[i].v_B.w[1].v_C.w'$ avec w' un mot sur \mathbb{N} . La bijection indique que $(A \wedge B) \wedge C$ est sous-arbre de T_0 . D'autre part, nous avons $v_1 = u.v_A.v_0[i] + \Delta.v_B.w[1].v_C.w'$ où $\Delta = v_1[i] - v_0[i]$. Comme v_1 est un vecteur de poids valide et que Δ est minimal, $v_1[i] = |v_B| + |w[1]| + |v_C| + 1$. La bijection indique donc que $A \wedge (B \wedge C)$ est sous-arbre de T_1 . L'arbre T_1 n'est modifié que localement au sous-arbre d'indice canonique i . Nous avons donc bien $T_1 \in \text{Rot}(T_0)$. \square

Arbre binaire	Mot de Dyck	Langage des feuilles	Vecteur de poids
	<i>aaaabbbb</i>	{0000, 0001, 001, 01, 1}	1111
	<i>aaabbbab</i>	{000, 001, 01, 10, 11}	1121
	<i>aaabbabb</i>	{000, 001, 010, 011, 1}	1211
	<i>aabbaabb</i>	{00, 01, 100, 101, 11}	1311
	<i>aabbabab</i>	{00, 01, 10, 110, 111}	1321
	<i>aaababbb</i>	{000, 0010, 0011, 01, 1}	2111
	<i>aababbab</i>	{00, 010, 011, 10, 11}	2121
	<i>aabaabbb</i>	{00, 0100, 0101, 011, 1}	3111
	<i>aabababb</i>	{00, 010, 0110, 0111, 1}	3211
	<i>abaaabbb</i>	{0, 1000, 1001, 101, 11}	4111
	<i>abaabbab</i>	{0, 100, 101, 110, 111}	4121
	<i>abaababb</i>	{0, 100, 1010, 1011, 11}	4211
	<i>ababaabb</i>	{0, 10, 1100, 1101, 111}	4311
	<i>abababab</i>	{0, 10, 110, 1110, 1111}	4321

FIG. 2.6 – Les arbres binaires de \mathcal{A}_4 et leurs codages.

Chapitre 3

Le treillis de Tamari

Dans cette partie, nous allons rappeler ce qu'est l'ordre de Tamari. Nous allons commencer par définir une relation sur l'ensemble des arbres binaires, puis, après avoir montré que cette relation est une relation d'ordre, nous allons montrer qu'elle définit une structure de treillis. Nous donnons une procédure de calcul pour déterminer la borne inférieure de deux arbres binaires dans le treillis de Tamari puis nous réaliserons une courte étude bibliographique sur ce treillis très connu et étudié.

3.1 Définitions et propriétés de base

3.1.1 La relation d'ordre de Tamari

Nous commençons par définir une relation sur l'ensemble des arbres binaires à n nœuds :

Définition 3.1.1. (Relation partielle de Tamari)

Soient T_0 et T_1 deux arbres binaires à n nœuds. La relation \prec , dite *relation partielle de Tamari* est définie par l'équivalence :

$$T_0 \prec T_1 \Leftrightarrow T_1 \in \text{Rot}(T_0). \quad (3.1)$$

Cette relation est étendue de la manière suivante :

Définition 3.1.2. (Relation de Tamari)

La relation \preceq , dite *relation de Tamari*, est la clôture réflexive et transitive de la relation partielle de Tamari \prec .

Nous allons bientôt montrer que la relation de Tamari est une relation d'ordre et qu'elle définit une structure de treillis sur l'ensemble des arbres binaires à n nœuds. En attendant, nous pouvons d'ores et déjà donner quelques diagrammes de Hasse de ces relations. Les premiers treillis sont très simples (voir Figure 3.1). Le treillis d'ordre 3 est appelé le *pentagone* (voir Figure 3.2).

Le fait que la relation \preceq est un ordre se montre facilement en considérant le codage des arbres binaires par les vecteurs de poids munis de leur relation d'ordre définie naturellement de la façon suivante : si v_0 et v_1 sont deux vecteurs de poids à n composantes, on a $v_0 \leq v_1$ si et seulement si $\forall 1 \leq i \leq n, v_0[i] \leq v_1[i]$. Cette relation est clairement une relation d'ordre. Montrons que la relation d'ordre sur les vecteurs de poids de et la relation de Tamari coïncident :

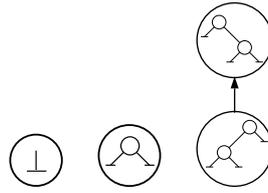


FIG. 3.1 – Les treillis de Tamari d'ordre 0, 1 et 2.

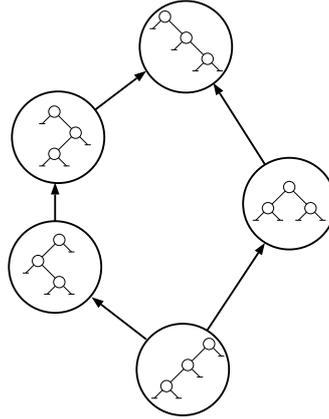


FIG. 3.2 – Le treillis de Tamari d'ordre 3.

Proposition 3.1.1. (Relation de Tamari et vecteurs de poids)

Soient T_0 et T_1 deux arbres binaires à n nœuds. Nous avons $T_0 \preceq T_1$ si et seulement si $V(T_0) \leq V(T_1)$.

Preuve. (\Rightarrow) : supposons $T_0 \preceq T_1$. Par définition, il existe une suite finie d'arbres $(S_i)_{1 \leq i \leq k}$ avec $k \geq 0$ telle que $T_0 \prec S_1 \prec \dots \prec S_k \prec T_1$. D'après la Proposition 2.3.5, nous avons $V(T_0) \leq V(S_1)$, $V(S_i) \leq V(S_{i+1})$ pour $1 \leq i \leq k-1$ et $V(S_k) \leq V(T_1)$. Ceci implique $V(T_0) \leq V(T_1)$ par la transitivité de la relation d'ordre sur les vecteurs.

(\Leftarrow) : Supposons $V(T_0) \leq V(T_1)$. Il existe alors naturellement une suite finie de vecteurs de poids $(v_i)_{1 \leq i \leq k}$ avec $k \geq 0$ telle que $V(T_0) \leq v_1 \leq \dots \leq v_k \leq V(T_1)$ avec $V(T_0)$ couvert par v_1 , v_i couvert par v_{i+1} pour $1 \leq i \leq k-1$ et v_k couvert par $V(T_1)$. En réalisant la bijection établie par la Proposition 2.3.1, on en déduit qu'il existe une suite d'arbres $(S_i)_{1 \leq i \leq k}$ tels que $S_i = V^{-1}(v_i)$ pour $1 \leq i \leq k$. D'après la Proposition 2.3.5 et la définition 3.1.2, nous avons bien $T_0 \preceq T_1$. \square

La relation de Tamari définit donc une relation d'ordre sur l'ensemble des arbres binaires. On note $\mathcal{T}_n = (\mathcal{A}_n, \preceq)$ l'ensemble des arbres binaires à n nœuds munis de la relation d'ordre de Tamari. La relation d'ordre \preceq est une relation d'ordre partiel (voir Figure 3.2 pour un exemple minimal d'un couple d'arbres incomparables). L'objet \mathcal{T}_n est donc un ensemble partiellement ordonné.

3.1.2 La structure de treillis

Nous allons montrer que \mathcal{T}_n est plus qu'un ensemble partiellement ordonné : \mathcal{T}_n est un treillis. Nous supposons connues les bases de la théorie des ensembles partiellement ordonnés et des treillis[22].

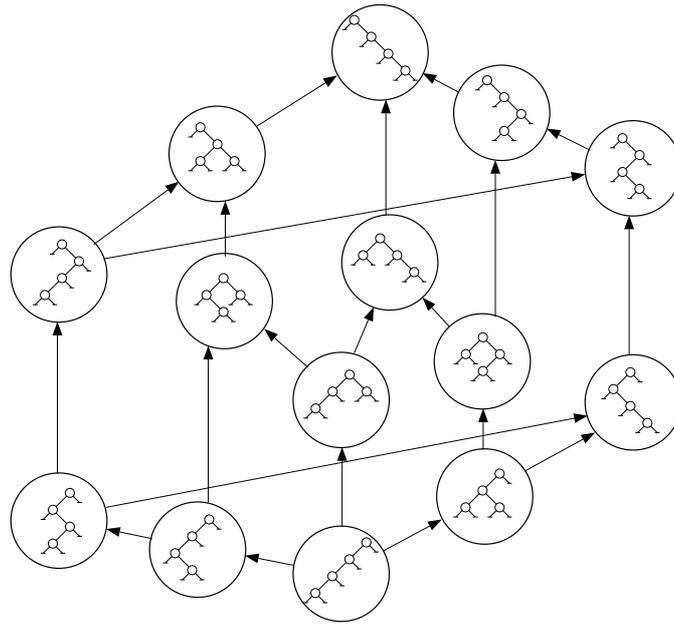


FIG. 3.3 – Le treillis de Tamari d'ordre 4.

Proposition 3.1.2. (Borne inférieure dans l'ordre de Tamari)

Pour tout couple $(T_0, T_1) \in \mathcal{A}_n \times \mathcal{A}_n$, la borne inférieure de l'ensemble $\{T_0, T_1\}$ est l'arbre binaire T_2 défini par

$$T_2 = V^{-1}(v) \text{ où } v[i] = \min \{V(T_0)[i], V(T_1)[i]\} \text{ pour } 1 \leq i \leq n. \quad (3.2)$$

Preuve. Montrons que v est bien un vecteur de poids. Posons $t_0 = V(T_0)$ et $t_1 = V(T_1)$. La condition (C_1) de la définition 2.3.1 est clairement respectée par v . Supposons par l'absurde que v ne respecte pas la condition (C_2) . Il existe donc des indices $1 \leq i \leq n$ et $1 \leq r \leq v[i] - 1$ tels que $v[i+r] > v[i] - r$. Nous avons maintenant plusieurs cas en fonction des valeurs $t_0[i]$, $t_0[i+r]$, $t_1[i]$ et $t_1[i+r]$:

- cas $t_0[i] = v[i]$ et $t_0[i+r] = v[i+r]$. Cela signifie que le vecteur t_0 ne respecte pas la condition (C_2) , ce qui est contraire aux hypothèses. Le cas $t_1[i] = v[i]$ et $t_1[i+r] = v[i+r]$ est analogue.
- Cas $t_0[i+r] = v[i+r]$ et $t_0[i] \neq v[i]$. Comme $t_1[i] = v[i]$, nous avons $1 \leq r \leq t_1[i]$ et comme le vecteur t_1 respecte la condition (C_2) , on a $t_1[i+r] \leq t_1[i] - r$. Comme $v[i+r] \leq t_1[i+r]$ et $t_1[i] = v_1[i]$, nous avons $v[i+r] \leq v[i] - r$, ce qui est contradictoire avec nos hypothèses sur v . Le cas $t_1[i+r] = v[i+r]$ et $t_1[i] \neq v[i]$ est analogue.

Ceci montre que v est bien un vecteur de poids.

Montrons maintenant que v est le plus grand minorant de $\{t_0, t_1\}$. Supposons qu'il existe un vecteur de poids v' tel que $v' \leq t_0$, $v' \leq t_1$ et $v < v'$. Il existe donc un indice $1 \leq i \leq n$ tel que $v[i] < v'[i]$. Ceci implique $t_0[i] < v'[i]$ ou $t_1[i] < v'[i]$, ce qui est contradictoire avec l'hypothèse que v' est un minorant de $\{t_0, t_1\}$. Le vecteur v est donc le plus grand minorant de $\{t_0, t_1\}$.

Par la Proposition 3.1.1, le treillis \mathcal{T}_n et le treillis constitué des vecteurs de poids muni de leur relation d'ordre \leq sont isomorphes, $T_2 = V^{-1}(v)$ est donc bien la borne inférieure de l'ensemble $\{T_0, T_1\}$. \square

Cette proposition nous donne un moyen effectif de calculer la borne inférieure

d'un couple d'arbres binaires.

```

BORNEINFÉRIEURE( $T_0, T_1$ )
  > Entrée :  $T_0, T_1 \in \mathcal{A}_n$ 
  > Sortie :  $T_2 \in \mathcal{A}_n$ , égal à  $\inf\{T_0, T_1\}$ 
  1 Début
  2    $t_0 \leftarrow V(T_0)$ 
  3    $t_1 \leftarrow V(T_1)$ 
  4    $v \leftarrow \epsilon$ 
  5   pour  $i \leftarrow 1$  à  $n$  faire
  6      $v \leftarrow v.\min\{t_0[i], t_1[i]\}$ 
  7   retourner  $V^{-1}(v)$ 
  8 Fin

```

FIG. 3.4 – Algorithme de calcul de \inf dans le treillis de Tamari.

Le calcul de la borne supérieure d'un couple d'arbres binaires se réalise à partir du calcul de la borne inférieure et du symétrique des arbres du couple. Nous énonçons la proposition suivante sans preuve supplémentaire[13] :

Proposition 3.1.3. (Borne supérieure dans l'ordre de Tamari)

Pour tout couple $(T_0, T_1) \in \mathcal{A}_n \times \mathcal{A}_n$, la borne supérieure de l'ensemble $\{T_0, T_1\}$ est l'arbre binaire T_2 défini par

$$T_2 = \leftrightarrow (\inf (\{\leftrightarrow (T_0), \leftrightarrow (T_1)\})). \quad (3.3)$$

Nous pouvons finalement établir le théorème suivant :

Théorème 3.1.1. (Treillis de Tamari)

Pour tout $n \geq 0$, \mathcal{T}_n est un treillis.

Preuve. D'après la Proposition 3.1.2, tout couple d'éléments de \mathcal{T}_n admet une borne inférieure, et d'après la Proposition 3.1.2, tout couple d'éléments de \mathcal{T}_n admet une borne supérieure. Ceci montre que \mathcal{T}_n est un treillis. \square

Caractérisons le plus petit et le plus grand élément du treillis de Tamari :

Proposition 3.1.4. (Plus grand et plus petit élément dans l'ordre de Tamari)

Pour tout entier $n \geq 0$, il existe dans le treillis de Tamari \mathcal{T}_n les arbres

$$T_n^{\min} = V^{-1}(1^n) \quad (3.4)$$

et

$$T_n^{\max} = V^{-1}(n.(n-1)...1) \quad (3.5)$$

où T_n^{\min} est le plus petit élément de \mathcal{T}_n et T_n^{\max} le plus grand élément.

Preuve. Raisonnons sur l'ensemble partiellement ordonné des vecteurs de poids de longueur n munis de leur relation d'ordre \leq . Le vecteur $u = 1^n$ est bien un vecteur de poids et il est clairement toujours comparable inférieurement à tout autre vecteur de poids de longueur n . De même, le vecteur $v = n.(n-1)...1$ est un vecteur de poids et il est toujours supérieur à tout autre vecteur de poids de même longueur. D'après la Proposition 3.1.1, ce dernier ensemble partiellement ordonné est isomorphe à \mathcal{T}_n , ce qui montre la proposition. \square

La figure 3.5 illustre le plus petit et le plus grand élément de \mathcal{T}_n conformément à la Proposition 3.1.4.

Il est clair que cette chaîne contient bien des vecteurs de poids et qu'elle est de longueur $\frac{(n-1)n+2}{2}$. \square

Si nous regardons ce qu'il se passe sur les arbres binaires, pour construire une chaîne de longueur minimale de la forme $C = \{T_n^{\min}, \dots, T_n^{\max}\}$, il suffit, en partant de l'arbre T_n^{\min} de réaliser successivement les rotations d'adresse u où u est l'adresse du nœud qui possède le plus grand indice canonique de T_i . Graphiquement, le successeur T_{i+1} d'un élément T_i de la chaîne est obtenu en effectuant une rotation le plus à droite possible dans T_i .

Similairement, pour construire une chaîne de longueur maximale de la forme $C = \{T_n^{\min}, \dots, T_n^{\max}\}$, en partant de l'arbre T_n^{\min} , on construit le successeur T_{i+1} d'un arbre T_i en réalisant une rotation d'adresse u où u est l'adresse du nœud qui possède le plus petit indice canonique de T_i . Graphiquement, la rotation est effectuée le plus à gauche possible.

Passons à présent en revue quelques propriétés remarquables.

Le nombre d'intervalles de \mathcal{T}_n , c'est à dire le nombre de couples (T_1, T_2) qui vérifient $T_1 \preceq T_2$ est $\frac{2(4n+1)!}{(n+1)!(3n+2)!}$. Cette propriété a été montrée par F. Chapoton en 2006[8].

En considérant le graphe non orienté dont les sommets sont les éléments de \mathcal{A}_n et les arrêtes les paires $\{T_1, T_2\}$ lorsque T_1 couvre T_2 ou T_2 couvre T_1 pour la relation d'ordre de Tamari, nous obtenons un objet combinatoire nommé *associaèdre*[16]. Il est naturel de vouloir calculer le diamètre de ce graphe. Le diamètre est la distance maximale entre deux sommets et la distance entre deux sommets T_1 et T_2 est la longueur du plus court chemin qui passe par les sommets T_1 et T_2 [10, 20]. En 1988, D. Sleator, R. Tarjan et W. Thurston montrent de manière combinatoire que ce diamètre est inférieur à $2n-6$ pour $n \geq 11$, et, en utilisant des arguments de géométrie hyperbolique, ils montrent que ce diamètre est supérieur à $2n-6$ pour $n \geq n_0$ ou n_0 est un très grand entier[20]. En 2009, P. Dehornoy développe de nouvelles méthodes combinatoires pour affiner les résultats précédents, et en particulier pour préciser la borne inférieure du diamètre de l'associaèdre. Il montre entre autres que cette borne inférieure est $2n - O(\sqrt{n})$ [10].

En 1993, J. M. Lucas, D. R. van Baronaigien et F. Ruskey montrent que l'associaèdre admet un parcours hamiltonien[18] et donnent un algorithme qui permet de générer l'ensemble des arbres binaires par des rotations gauches et droites en exploitant cette propriété[18, 19, 13]. Ils obtiennent ainsi un code de Gray[19] qui permet de générer l'ensemble des objets catalans de manière très efficace.

3.2.2 Manifestations de l'ordre de Tamari et remarques

L'ordre de Tamari n'a pas été initialement conçu sur le modèle des arbres binaires. L'opération de rotation dans un arbre binaire à n nœuds peut être interprétée comme une modification de l'associativité dans une expression à $n+1$ opérands où un opérateur non associatif est appliqué n fois. Par exemple, en fixant le symbole $*$ comme opérateur non associatif, les expressions $((x_1 * x_2) * (x_3 * x_4))$ et $(x_1 * (x_2 * (x_3 * x_4)))$ sont en relation de Tamari. La relation de couverture revient à modifier localement une expression de la forme $((a * b) * c)$ en l'expression $(a * (b * c))$. Il n'est pas très difficile de montrer que l'ensemble de telles expressions à $n+1$ opérands munies de cette relation de couverture est un treillis isomorphe à \mathcal{T}_n .

Certains auteurs[23, 11] définissent cet ordre sur l'ensemble des vecteurs v à n composantes à valeurs dans \mathbb{N} tels que $i \leq v[i] \leq n$ et $i \leq j \leq v[i] \Rightarrow v[j] \leq v[i]$ et dont la relation d'ordre est la même que celle définie sur les vecteurs de poids. C'est d'ailleurs la définition originale du treillis de Tamari introduite par D. Tamari.

Il est bien entendu possible d'utiliser n'importe quel objet catalan muni de la relation de couverture adéquate[23, 24]. Cependant, pour certains de ces objets, la relation de couverture peut être assez difficile à exprimer, ce qui peut rendre, étant donné deux objets catalans C_0 et C_1 le problème consistant à décider si $C_0 \leq C_1$ beaucoup plus difficile que ce qu'il en est pour les vecteur de poids.

L'ordre de Tamari intervient également dans l'algèbre des arbres binaires[11, 15, 17] ou la multiplication de deux arbres binaires est exactement un intervalle du treillis de Tamari.

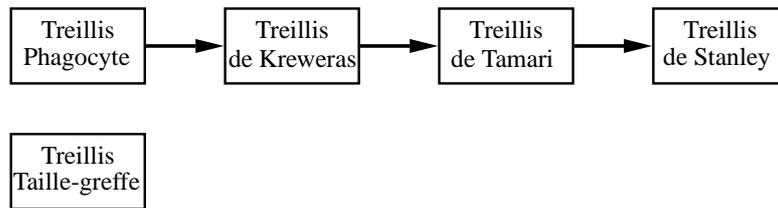


FIG. 3.6 – Différents treillis catalans et leur relation de raffinement.

Nous terminons cette étude du treillis de Tamari en évoquant d'autres treillis sur les ensembles catalans. En effet, la relation de Tamari ne définit pas la seule structure de treillis intéressante sur ces objets. Il existe les treillis *Taille-greffe*[5], *Phagocyte*[4], de *Kreweras*[14, 7, 13] et de *Stanley*[7, 13]. Une propriété remarquable est que le treillis de Stanley est un raffinement du treillis de Tamari qui lui même est un raffinement du treillis de Kreweras, qui est enfin un raffinement du treillis Phagocyte[4, 7, 13]. Le treillis Taille-greffe ne semble pas avoir de relation avec ces treillis (voir Figure 3.6).

Chapitre 4

Les arbres équilibrés et l'ordre de Tamari

Cette partie contient les résultats principaux de notre travail. Nous commençons par définir un ensemble d'outils qui vont nous servir à formaliser la suite de notre propos. Nous introduisons les concepts d'application de déséquilibre, de découpages, de motifs de déséquilibres et d'ensembles d'arbres caractérisés par les motifs qu'ils évitent.

Nous analysons ensuite les modifications qu'une opération de rotation entraîne sur l'application de déséquilibre de l'arbre sur lequel elle est appliquée et isolons un type particulier de rotations qui ne déséquilibrent pas un arbre initialement équilibré.

Nous divisons l'ensemble des arbres équilibrés en diverses classes selon la position qu'ils occupent dans le treillis de Tamari et donnons pour certaines d'entre elles une équation fonctionnelle qui admet comme point fixe la série génératrice qui dénombre ses éléments.

Nous terminons en montrant que les intervalles d'arbres équilibrés sont isomorphes à un hypercube d'une certaine dimension. L'outillage qui a été défini est utilisé pour montrer ce résultat, pour compter le nombre d'intervalles d'arbres équilibrés ainsi que le nombre d'hypercubes maximaux dans chaque treillis de Tamari.

4.1 Application et motif de déséquilibre

Nous définissons dans cette partie l'ensemble d'outils qui vont nous servir pour établir certains des résultats que contient ce mémoire.

4.1.1 Application de déséquilibre

Définition 4.1.1. (Application de déséquilibre)

Soit T un arbre binaire. On associe à T l'application γ_T qui fait correspondre à tout nœud x de T l'entier relatif défini par

$$\gamma_T(x) = \text{ht}(D) - \text{ht}(G) \quad (4.1)$$

où G est le sous-arbre gauche de x et D son sous-arbre droit. L'application γ_T est dite *application de déséquilibre* de l'arbre T et l'entier $\gamma_T(x)$ est appelé *valeur de déséquilibre* du nœud x .

Étant donné que γ_T est définie sur l'ensemble des nœuds de T , il est possible de voir γ_T comme une application définie de l'ensemble $\mathbb{A}_N(T)$ vers \mathbb{Z} par :

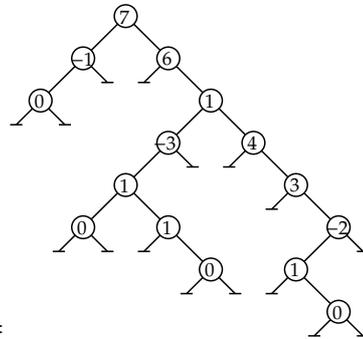
$$\gamma_T(u) = \text{ht}(\text{SArbre}(T, u1)) - \text{ht}(\text{SArbre}(T, u0)). \quad (4.2)$$

Par abus de notation, nous garderons le même symbole γ_T que son ensemble de définition soit l'ensemble des nœuds de T ou l'ensemble $\mathbb{A}_N(T)$.

L'application γ_T associe donc à chaque nœud un entier. Cette application n'est ni injective, ni surjective. Il est possible de considérer le triplet $(T, \gamma_T, \mathbb{Z})$ comme un arbre binaire étiqueté (voir définition 1.2.1), où chaque nœud est étiqueté par son déséquilibre.

On représente l'application γ_T par un mot v sur l'alphabet \mathbb{Z} tel que $v[i] = \gamma_T(i)$ pour $1 \leq i \leq \text{nd}(T)$ où les entiers i sont les indices canoniques des nœuds de T . Ce mot est appelé *mot infixe des déséquilibres*.

Exemple.



Soit l'arbre $T =$  étiqueté par les valeurs de déséquilibre associées à chacun de ses nœuds. Le mot infixe des déséquilibres qui représente γ_T est $0 - 170110 - 314310 - 2$.

Lemme 4.1.1. (Encadrement des valeurs de déséquilibre)

Si T est un arbre binaire à n nœuds et $u \in \mathbb{A}_N(T)$ une adresse d'un nœud de T , alors

$$n + 1 \leq \gamma_T(u) \leq n - 1. \quad (4.3)$$

Preuve. La borne maximale est atteinte pour $\gamma_T(\epsilon)$ avec $T = T_n^{\max}$, et la borne minimale pour $\gamma_T(\epsilon)$ avec $T = T_n^{\min}$. \square

4.1.2 Motifs de déséquilibre

De la même façon qu'il est possible de définir des langages de mots en spécifiant des facteurs obligatoires ou des facteurs à éviter, il est possible de définir des ensembles d'arbres binaires qui possèdent ou évitent certains motifs. Commençons par donner la définition d'un motif de déséquilibre :

Définition 4.1.2. (Motif de déséquilibre)

Un *motif de déséquilibre* est un arbre binaire non vide étiqueté sur \mathbb{Z} .

Nous sommes amenés à considérer des régions d'un arbre binaire que nous appelons *découpages*. Pour définir cette notion, procédons en deux temps en définissant le concept d'inclusion d'arbres :

Définition 4.1.3. (Inclusion d'arbres)

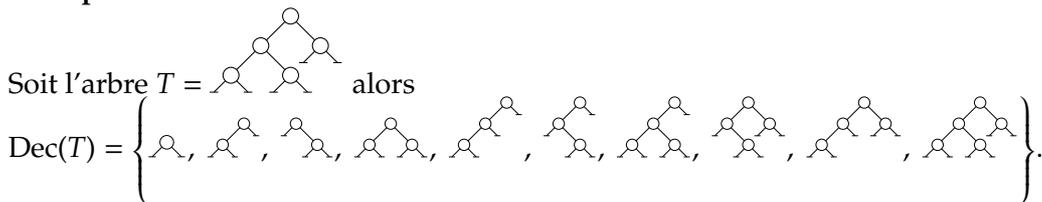
L'arbre T_0 est *inclus* dans l'arbre T_1 si le langage $\mathbb{A}_N(T_0)$ est inclus dans le langage $\mathbb{A}_N(T_1)$.

Définition 4.1.4. (Découpages)

L'arbre T_0 est un découpage de l'arbre T_1 si T_0 est non vide et est inclus dans un sous-arbre de T_1 .

Nous notons $\text{Dec}(T)$ l'ensemble des découpages de l'arbre T . Intuitivement, l'ensemble des découpages d'un arbre binaire T est l'ensemble des sous-graphes ordonnés connexes de T vu comme un graphe ordonné. Si la notion de facteur d'un mot est l'analogie à la notion de sous-arbre d'un arbre, la notion de sous-mot est aux mots ce que la notion de découpage est aux arbres binaires.

Exemple.



Nous définissons maintenant la notion d'occurrence d'un motif de déséquilibre dans un arbre binaire :

Définition 4.1.5. (Occurrence d'un motif de déséquilibre)

Soit m un motif de déséquilibre. On dit que l'arbre T admet une *occurrence* de m si il existe un découpage $D \in \text{Dec}(T)$ tel que $(D, \gamma_T, \mathbb{Z}) = m$.

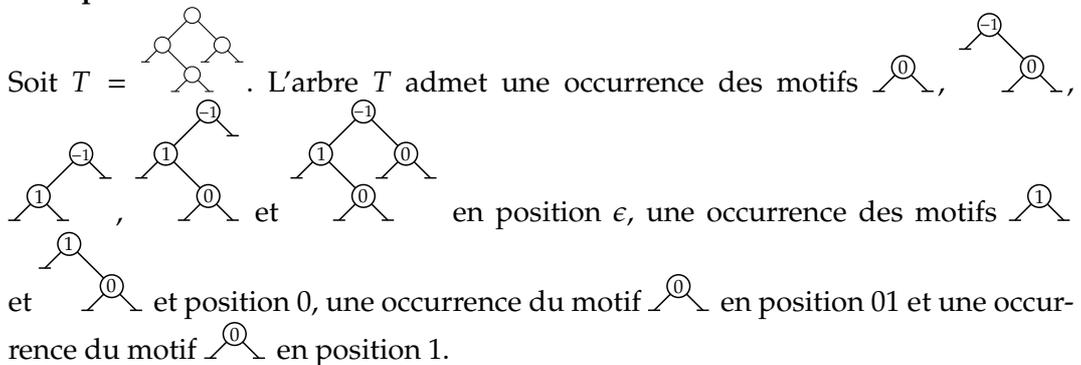
Autrement dit, l'arbre T admet une occurrence d'un motif de déséquilibre m s'il existe un découpage de T , qui, s'il est étiqueté par l'application de déséquilibre associée à T , est égal à m .

Bien entendu, la définition d'occurrence d'un motif de déséquilibre entraîne la notion de *position d'occurrence* d'un motif de déséquilibre. L'ensemble des positions d'occurrence du motif de déséquilibre m dans l'arbre T , noté $\text{Pos}_T(m)$ est un ensemble d'adresses de nœuds :

$$\text{Pos}_T(m) = \{v \in \{0, 1\}^* \mid \forall u \in \mathbb{A}_N(m) \mid v.u \in \mathbb{A}_N(T) \text{ et } (T, \gamma_T, \mathbb{Z}) = m\}. \quad (4.4)$$

C'est l'ensemble des adresses des racines des découpages de T qui étiquetés par l'application de déséquilibre associée à T sont égaux au motif de déséquilibre m .

Exemple.



S'il n'y a pas d'ambiguïté, nous emploierons le terme *motif* à la place du terme *motif de déséquilibre*.

Définition 4.1.6. (Arbres qui évitent un ensemble de motifs)

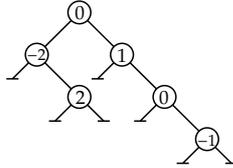
Soit M un ensemble de motifs. L'ensemble des arbres qui évitent M est l'ensemble

$$\mathcal{N}(M) = \{T \in \mathcal{A}_\infty \mid \forall m \in M, \text{Pos}_T(m) = \emptyset\}. \quad (4.5)$$

L'ensemble $\mathcal{N}(M)$ contient donc les arbres qui n'admettent aucune occurrence des motifs appartenant à M .

Pour éviter d'avoir à dessiner des arbres avec excès, nous allons adopter la notation suivante : étant donné un motif m , nous notons m par le couple (u, v) où u est le vecteur de poids correspondant à la forme de m et v la suite des étiquettes de m lues selon le parcours infixe.

Exemple.



Le motif $(214321, -22010 - 1)$.

- En posant $M = \{(1, d) \mid d \notin \{-1, 0, 1\}\}$, nous avons $\mathcal{N}(M) = \mathcal{E}_\infty$. L'ensemble M permet donc de décrire l'ensemble des arbres équilibrés.
- En posant $M = \{(1, d) \mid d \in \mathbb{N} \setminus \{0\}\}$, l'ensemble $\mathcal{N}(M)$ contient l'ensemble des arbres parfaits, c'est à dire l'ensemble des arbres de la suite $(P_i)_{i \geq 0}$ définie par $P_0 = \perp$ et $P_{i+1} = P_i \wedge P_i$.
- En posant $M = \{(11, d_1 d_2) \mid d_1, d_2 \in \mathbb{Z}\}$, l'ensemble $\mathcal{N}(M)$ contient l'ensemble des arbres T_n^{\max} pour $n \geq 0$.

4.2 Modification des déséquilibres pour la relation de couverture

Le but de cette section est de caractériser les modifications de l'application de déséquilibre d'un arbre T_0 lorsque une rotation lui est appliquée. Nous verrons que si l'arbre T_0 est équilibré et l'arbre T_1 résultant de la rotation l'est aussi, alors l'application de déséquilibre de T_1 est facilement calculable et est relativement peu modifiée par rapport à celle de T_0 .

En revanche, si l'arbre T_0 n'est pas équilibré, ou si l'arbre T_0 est équilibré mais T_1 ne l'est pas, nous verrons que l'application de déséquilibre de T_1 est sujette à plus de modifications et est donc moins facile à calculer.

4.2.1 Modification des déséquilibres locaux

Commençons par réaliser une étude de cas en analysant les modifications des déséquilibres de T_0 et T_1 lorsque $T_1 = \text{Rot}(T_0, \epsilon)$. Posons donc $T_0 = (A \wedge B) \wedge C$ et $T_1 = A \wedge (B \wedge C)$ avec A, B, C des arbres binaires. On pose y le nœud d'adresse ϵ dans T_0 et x le nœud d'adresse 0 dans T_1 (voir Figure 4.1). Pour alléger la notation, écrivons respectivement a, b et c pour les valeurs $\text{ht}(A), \text{ht}(B)$ et $\text{ht}(C)$. Remarquons tout de suite que les déséquilibres des nœuds qui appartiennent aux arbres A, B et C ne sont pas modifiés par la rotation. Seuls les déséquilibres des nœuds x et y sont potentiellement modifiés. D'après la définition 4.1.1, nous avons

4.2. MODIFICATION DES DÉSÉQUILIBRES POUR LA RELATION DE COUVERTURE

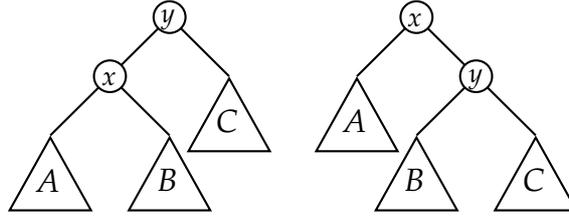


FIG. 4.1 – Les arbres T_0 et T_1 .

$$\begin{cases} \gamma_{T_0}(x) = b - a \\ \gamma_{T_0}(y) = c - \max\{a, b\} - 1, \end{cases} \quad (4.6) \quad \begin{cases} \gamma_{T_1}(x) = \max\{b, c\} + 1 - a \\ \gamma_{T_1}(y) = c - b. \end{cases} \quad (4.7)$$

Les deux occurrences de la fonction max dans ce système nous obligent à regarder les quatre cas qui surviennent lors de la comparaison de a avec b et de b avec c :

(1) si $b \geq a$ et $b \geq c$ alors

$$\begin{cases} \gamma_{T_0}(x) = b - a \\ \gamma_{T_0}(y) = c - b - 1, \end{cases} \quad \begin{cases} \gamma_{T_1}(x) = b + 1 - a \\ \gamma_{T_1}(y) = c - b; \end{cases}$$

(2) si $b \geq a$ et $b < c$ alors

$$\begin{cases} \gamma_{T_0}(x) = b - a \\ \gamma_{T_0}(y) = c - b - 1, \end{cases} \quad \begin{cases} \gamma_{T_1}(x) = c + 1 - a \\ \gamma_{T_1}(y) = c - b; \end{cases}$$

(3) si $b < a$ et $b \geq c$ alors

$$\begin{cases} \gamma_{T_0}(x) = b - a \\ \gamma_{T_0}(y) = c - a - 1, \end{cases} \quad \begin{cases} \gamma_{T_1}(x) = b + 1 - a \\ \gamma_{T_1}(y) = c - b; \end{cases}$$

(4) si $b < a$ et $b < c$ alors

$$\begin{cases} \gamma_{T_0}(x) = b - a \\ \gamma_{T_0}(y) = c - a - 1, \end{cases} \quad \begin{cases} \gamma_{T_1}(x) = c + 1 - a \\ \gamma_{T_1}(y) = c - b. \end{cases}$$

Calculons à présent les variations de $\gamma_{T_1}(x)$ par rapport à $\gamma_{T_0}(x)$ et de $\gamma_{T_1}(y)$ par rapport à $\gamma_{T_0}(y)$ pour chacun des cas précédents :

- (1) $\gamma_{T_1}(x) - \gamma_{T_0}(x) = b + 1 - a - (b - a) = 1,$
 $\gamma_{T_1}(y) - \gamma_{T_0}(y) = c - b - (c - b - 1) = 1;$
- (2) $\gamma_{T_1}(x) - \gamma_{T_0}(x) = c + 1 - a - (b - a) = c - b + 1 \geq 2,$
 $\gamma_{T_1}(y) - \gamma_{T_0}(y) = c - b - (c - b - 1) = 1;$
- (3) $\gamma_{T_1}(x) - \gamma_{T_0}(x) = b + 1 - a - (b - a) = 1,$
 $\gamma_{T_1}(y) - \gamma_{T_0}(y) = c - b - (c - a - 1) = a - b + 1 \geq 2;$
- (4) $\gamma_{T_1}(x) - \gamma_{T_0}(x) = c + 1 - a - (b - a) = c - b + 1 \geq 2,$
 $\gamma_{T_1}(y) - \gamma_{T_0}(y) = c - b - (c - a - 1) = a - b + 1 \geq 2.$

Dans chaque cas, nous avons $\gamma_{T_1}(x) > \gamma_{T_0}(x)$ et $\gamma_{T_1}(y) > \gamma_{T_0}(y)$. Nous venons donc de montrer le lemme suivant :

Lemme 4.2.1. (Incrémentation des déséquilibres lors d'une rotation)

Si T_0 et T_1 sont des arbres binaires tels que $T_0 \prec T_1$ avec $T_0 = (A \wedge B) \wedge C$, $T_1 = A \wedge (B \wedge C)$ et A, B, C des arbres binaires, alors $\gamma_{T_1}(\epsilon) > \gamma_{T_0}(\epsilon)$ et $\gamma_{T_1}(1) > \gamma_{T_0}(1)$.

Intéressons-nous maintenant aux variations de hauteur de l'arbre T_1 par rapport à la hauteur de l'arbre T_0 . Résolvons donc l'équation :

$$\text{ht}(T_1) - \text{ht}(T_0) = (\max\{a, \max\{b, c\} + 1\} + 1) - (\max\{\max\{a, b\} + 1, c\} + 1) \quad (4.8)$$

pour chacun de nos cas :

- (1) $\text{ht}(T_1) - \text{ht}(T_0) = (b + 1) - (b + 1) = 0$;
- (2) $\text{ht}(T_1) - \text{ht}(T_0) = (c + 1) - (c) = 1$;
- (3) $\text{ht}(T_1) - \text{ht}(T_0) = (a) - (a + 1) = -1$;
- (4) $\text{ht}(T_1) - \text{ht}(T_0) = (\max\{a, c + 1\}) - (\max\{a + 1, c\})$, ce qui donne
 - (4.1) 0 si $a = c$,
 - (4.2) 1 si $a < c$,
 - (4.3) -1 si $a > c$.

Nous trouvons que la différence absolue de hauteur entre les arbres T_0 et T_1 est d'au plus 1. Formalisons cette propriété par le lemme suivant :

Lemme 4.2.2. (Modification de la hauteur lors d'une rotation)

Si T_0 et T_1 sont des arbres binaires tels que $T_0 \prec T_1$ avec $T_0 = (A \wedge B) \wedge C$, $T_1 = A \wedge (B \wedge C)$ et A, B, C des arbres binaires, alors $|\text{ht}(T_1) - \text{ht}(T_0)| \leq 1$.

Dans la Figure 4.2, les arbres T_0 et T_1 sont représentés avec les valeurs de déséquilibres de leurs nœuds x et y . Elle récapitule les différents cas que nous avons examinés jusqu'à présent.

Nous avons aussi le lemme suivant :

Lemme 4.2.3. (Hauteur de la couverture d'un arbre équilibré)

Si T_0 et T_1 sont des arbres binaires équilibrés tels que $T_0 \prec T_1$ avec $T_0 = (A \wedge B) \wedge C$, $T_1 = A \wedge (B \wedge C)$ et A, B, C des arbres binaires, alors $\text{ht}(T_0) = \text{ht}(T_1)$.

Preuve. Un examen des cas que nous venons de lister montre que T_0 et T_1 ne peuvent être équilibrés que dans les cas (1) et (4.1), tout en vérifiant bien $\text{ht}(T_0) = \text{ht}(T_1)$. \square

Par induction, il est simple de voir que le Lemme 4.2.3 est également vrai pour deux arbres équilibrés T_0 et T_1 tel que $T_1 = \text{Rot}(T_0, u)$ avec u une adresse de rotation non nécessairement égale à ϵ .

4.2.2 Propagation des modifications des déséquilibres

Une rotation d'adresse $u \neq \epsilon$ appliquée à l'arbre T_0 est susceptible d'engendrer d'autres modifications dans l'application de déséquilibre de T_0 que celles que nous avons étudiées. Ceci provient du fait qu'une rotation peut modifier la hauteur du sous-arbre impliqué dans la rotation et donc potentiellement modifier d'autres valeurs. Commençons par isoler les adresses des nœuds dont nous sommes certains que la valeur de déséquilibre n'est pas modifiée par la rotation :

4.2. MODIFICATION DES DÉSÉQUILIBRES POUR LA RELATION DE COUVERTURE

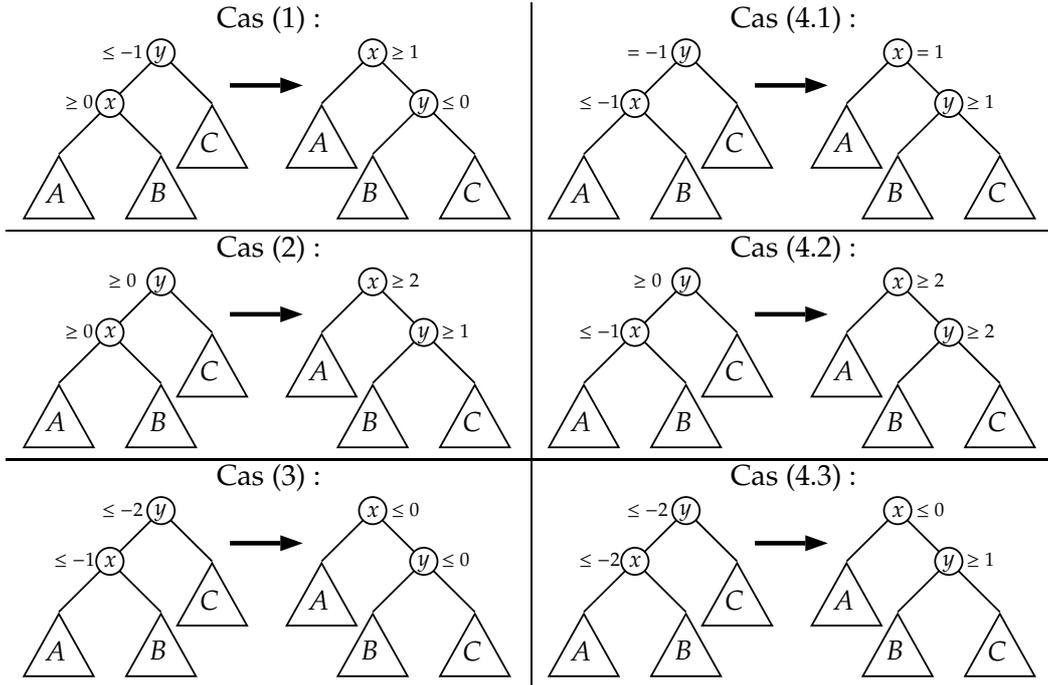


FIG. 4.2 – Résumé des modifications sur les déséquilibres de T_0 et T_1 .

Lemme 4.2.4. (Nœuds non impliqués dans la modification des déséquilibres)
 Soient T_0 et T_1 des arbres binaires tels que $T_1 = \text{Rot}(T_0, u)$ pour une adresse de rotation u .
 Alors pour tout $v \in \{0, 1\}^*$ tel que $v \notin \text{Pref}(u)$ et $u \notin \text{Pref}(v)$, nous avons $\gamma_{T_0}(v) = \gamma_{T_1}(v)$.

Preuve. La rotation modifie uniquement le sous-arbre d'adresse u de T_0 . Comme u et v ne sont pas en relation de préfixité, on a $\text{SArbre}(T_0, v) = \text{SArbre}(T_1, v)$, ce qui établit la proposition. \square

Les seules valeurs de déséquilibre qui sont susceptibles d'être modifiées, en plus des deux valeurs de déséquilibres locales au sous-arbre impliqué dans la rotation, sont donc les valeurs des nœuds de T_1 d'adresse $v \in \text{Pref}(u) \setminus \{u\}$.

Lemme 4.2.5. (Propagation des modifications)
 Soient T_0 et T_1 des arbres binaires tels que $T_1 = \text{Rot}(T_0, u)$ pour une adresse de rotation u ,
 $v \in \{0, 1\}^*$ et $\Delta \in \mathbb{Z}$. Alors :

- (i) si $v0 \in \text{Pref}(u)$ et $\text{ht}(\text{SArbre}(T_0, v0)) + \Delta = \text{ht}(\text{SArbre}(T_1, v0))$ alors
 $\gamma_{T_0}(v) = \gamma_{T_1}(v) + \Delta$;
- (ii) si $v1 \in \text{Pref}(u)$ et $\text{ht}(\text{SArbre}(T_0, v1)) + \Delta = \text{ht}(\text{SArbre}(T_1, v1))$ alors
 $\gamma_{T_0}(v) + \Delta = \gamma_{T_1}(v)$.

Preuve. (i) : nous avons

$$\gamma_{T_1}(v) = \text{ht}(\text{SArbre}(T_1, v1)) - \text{ht}(\text{SArbre}(T_1, v0)) \quad (4.9)$$

$$= \text{ht}(\text{SArbre}(T_0, v1)) - (\text{ht}(\text{SArbre}(T_0, v0)) + \Delta) \quad (4.10)$$

$$= \gamma_{T_0}(v) - \Delta \quad (4.11)$$

$$\Rightarrow \gamma_{T_0}(v) = \gamma_{T_1}(v) + \Delta. \quad (4.12)$$

(ii) : nous avons similairement

$$\gamma_{T_1}(v) = \text{ht}(\text{SArbre}(T_1, v1)) - \text{ht}(\text{SArbre}(T_1, v0)) \quad (4.13)$$

$$= \text{ht}(\text{SArbre}(T_0, v1)) + \Delta - \text{ht}(\text{SArbre}(T_0, v0)) \quad (4.14)$$

$$= \gamma_{T_0}(v) + \Delta \quad (4.15)$$

$$\Rightarrow \gamma_{T_0}(v) + \Delta = \gamma_{T_1}(v). \quad (4.16)$$

□

En fait, d'après le Lemme 4.2.2, nous pouvons facilement voir par induction que les hauteurs des sous-arbres modifiés par la rotation ne varient pas plus de 1. Nous avons donc $\Delta \in \{-1, 0, 1\}$.

Lemme 4.2.6. (Déséquilibre inchangé, même hauteur)

Soient T_0 et T_1 des arbres binaires tels que $T_1 = \text{Rot}(T_0, u)$ pour une adresse de rotation u et $v \in \text{Pref}(u) \setminus \{u\}$.

Si $\gamma_{T_0}(v) = \gamma_{T_1}(v)$ alors $\text{ht}(\text{SArbre}(T_0, v)) = \text{ht}(\text{SArbre}(T_1, v))$.

Preuve. Comme $\gamma_{T_0}(v) = \gamma_{T_1}(v)$, nous avons l'égalité :

$$\text{ht}(\text{SArbre}(T_0, v1)) - \text{ht}(\text{SArbre}(T_0, v0)) = \text{ht}(\text{SArbre}(T_1, v1)) - \text{ht}(\text{SArbre}(T_1, v0)). \quad (4.17)$$

Si $v0 \in \text{Pref}(u)$, alors $v1 \notin \text{Pref}(u)$ et $u \notin \text{Pref}(v1)$, ce qui, d'après le Lemme 4.2.4, implique

$$\text{ht}(\text{SArbre}(T_0, v1)) = \text{ht}(\text{SArbre}(T_1, v1)). \quad (4.18)$$

Des deux égalités précédentes nous déduisons l'égalité :

$$\text{ht}(\text{SArbre}(T_0, v0)) = \text{ht}(\text{SArbre}(T_1, v0)), \quad (4.19)$$

ce qui force $\text{ht}(\text{SArbre}(T_0, v)) = \text{ht}(\text{SArbre}(T_1, v))$. Le cas $v1 \in \text{Pref}(u)$ est analogue. □

Lemme 4.2.7. (Rupture de propagation des modifications des déséquilibres)

Soient T_0 et T_1 des arbres binaires tels que $T_1 = \text{Rot}(T_0, u)$ pour une adresse de rotation u et $v \in \text{Pref}(u) \setminus \{u\}$ tel que $\gamma_{T_0}(v) = \gamma_{T_1}(v)$.

Alors, pour tout $w \in \text{Pref}(v)$, $\gamma_{T_0}(w) = \gamma_{T_1}(w)$.

Preuve. C'est une simple conséquence du Lemme 4.2.6. Comme

$$\gamma_{T_0}(v) = \gamma_{T_1}(v), \quad (4.20)$$

nous avons

$$\text{ht}(\text{SArbre}(T_0, v)) = \text{ht}(\text{SArbre}(T_1, v)). \quad (4.21)$$

Si $w0 \in \text{Pref}(v)$, alors par induction nous obtenons l'égalité

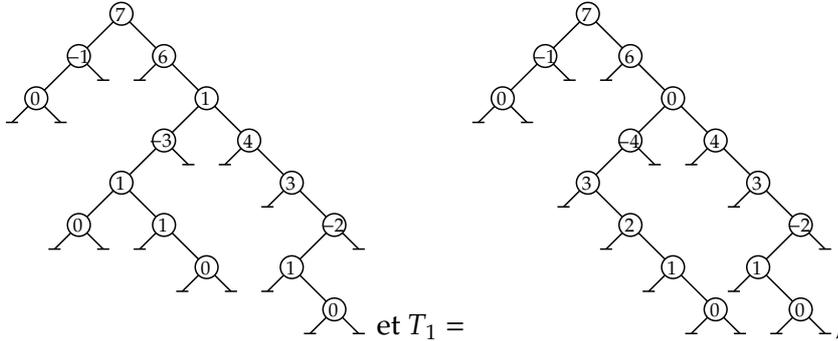
$$\text{ht}(\text{SArbre}(T_0, w0)) = \text{ht}(\text{SArbre}(T_1, w0)). \quad (4.22)$$

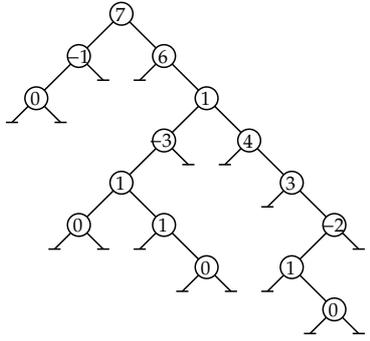
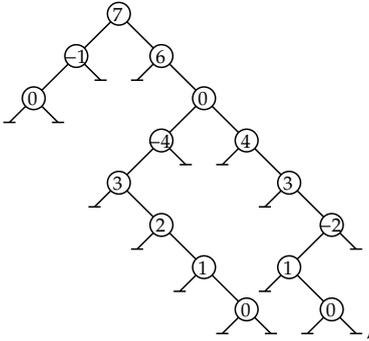
De plus, nous avons naturellement $\text{ht}(\text{SArbre}(T_0, w1)) = \text{ht}(\text{SArbre}(T_1, w1))$ ce qui montre $\gamma_{T_0}(w) = \gamma_{T_1}(w)$. Le cas $w1 \in \text{Pref}(u)$ est analogue. □

Nous avons donc montré qu'une rotation d'adresse u modifie, en plus des deux déséquilibres propres au sous-arbre impliqué dans la rotation, uniquement les déséquilibres des nœuds d'adresse w , avec $w \in \text{Pref}(u) \setminus \{u\}$ et $|v| < |w| < |u|$ pour v l'adresse la plus longue telle que $\gamma_{T_0}(v) = \gamma_{T_1}(v)$.

4.2. MODIFICATION DES DÉSÉQUILIBRES POUR LA RELATION DE COUVERTURE

Exemple. Considérons les arbres



$T_0 =$  et $T_1 =$  ,
 étiquetés par les valeurs de leur application de déséquilibre. Nous avons $T_1 = \text{Rot}(T_0, 1100)$ et $v = 1$ l'adresse de longueur maximale telle que $\gamma_{T_0}(v) = \gamma_{T_1}(v)$. L'ensemble des nœuds d'adresse à la fois préfixe de 1100 et strictement plus longue que l'adresse v possèdent une valeur de déséquilibre différente dans les arbres T_0 et T_1 . À l'inverse, l'ensemble des nœuds d'adresse préfixe de v possèdent une même valeur de déséquilibre dans T_0 et T_1 .

Dans la Figure 4.3, S_0 et S_1 sont les sous-arbres impliqués dans la rotation. Les nœuds qui appartiennent au chemin représenté en double ligne dans l'arbre T_0 possèdent des valeurs de déséquilibres différentes des nœuds du chemin représenté en double ligne dans l'arbre T_1 . Les nœuds du chemin allant des racines aux nœuds d'adresse v possèdent dans les deux arbres les mêmes valeurs de déséquilibre.

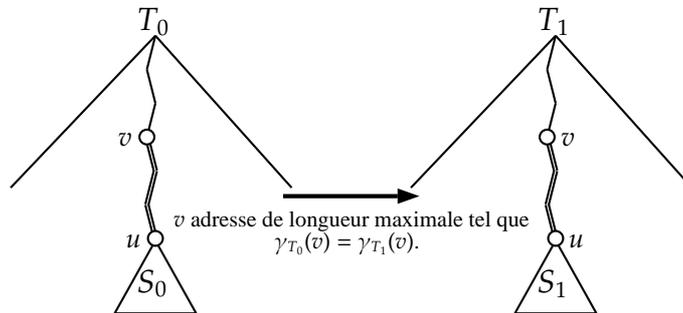


FIG. 4.3 – Région de modifications des déséquilibres.

Lemme 4.2.8. (Hauteur d'un sous-arbre non modifiée et valeurs de déséquilibre)
 Soient T_0 et T_1 des arbres binaires tels que $T_1 = \text{Rot}(T_0, u)$ pour une adresse de rotation u . Si $\text{ht}(\text{SArbre}(T_0, u)) = \text{ht}(\text{SArbre}(T_1, u))$, alors pour tout $v \in \text{Pref}(u) \setminus \{u\}$, nous avons $\gamma_{T_0}(v) = \gamma_{T_1}(v)$.

Preuve. Si u est vide, la proposition est trivialement vraie. On suppose donc u non vide et on pose w l'adresse du père de u . Si $u = w1$, alors par hypothèse

$$\text{ht}(\text{SArbre}(T_1, w1)) = \text{ht}(\text{SArbre}(T_0, w1)) \quad (4.23)$$

et par définition de l'opération de rotation,

$$\text{ht}(\text{SArbre}(T_1, w0)) = \text{ht}(\text{SArbre}(T_0, w0)). \quad (4.24)$$

Nous avons donc $\gamma_{T_0}(w) = \gamma_{T_1}(w)$. D'après le Lemme 4.2.7, pour tout $v \in \text{Pref}(w)$, $\gamma_{T_0}(v) = \gamma_{T_1}(v)$, ce qui établit la proposition. Le cas est identique si $u = w0$. \square

Proposition 4.2.1. (Rotations conservatrices d'équilibre)

Soient T_0 et T_1 des arbres équilibrés tels que $T_1 = \text{Rot}(T_0, u)$ pour une adresse de rotation u , y l'indice canonique du nœud racine de $\text{SArbre}(T_0, u)$ et x l'indice canonique du nœud fils gauche de y . Une et une seule de ces propriétés est vraie :

- (i) $(\gamma_{T_0}(x), \gamma_{T_0}(y)) = (-1, -1)$ et $(\gamma_{T_1}(x), \gamma_{T_1}(y)) = (1, 1)$;
- (ii) $(\gamma_{T_0}(x), \gamma_{T_0}(y)) = (0, -1)$ et $(\gamma_{T_1}(x), \gamma_{T_1}(y)) = (1, 0)$.

De plus, pour tout nœud d'indice canonique z différent de x et y , nous avons $\gamma_{T_0}(z) = \gamma_{T_1}(z)$.

Preuve. Nous pouvons poser $S_0 = \text{SArbre}(T_0, u) = (A \wedge B) \wedge C$ pour trois arbres binaires A, B et C . Il existe neuf configurations différentes pour le couple $(\gamma_{S_0}(x), \gamma_{S_0}(y))$ parce que chaque composante peut prendre $-1, 0$ ou 1 comme valeur puisque T_0 est équilibré. L'arbre S_0 se retrouve modifié en l'arbre $S_1 = A \wedge (B \wedge C)$ dans T_1 . D'après notre étude de cas précédente, nous voyons que seuls les cas $(\gamma_{S_0}(x), \gamma_{S_0}(y)) \in \{(-1, -1), (0, -1)\}$ forcent l'équilibre de S_1 qui vérifie $(\gamma_{S_1}(x), \gamma_{S_1}(y)) = (1, 1)$ si $(\gamma_{S_0}(x), \gamma_{S_0}(y)) = (-1, -1)$ et $(\gamma_{S_1}(x), \gamma_{S_1}(y)) = (1, 0)$ si $(\gamma_{S_0}(x), \gamma_{S_0}(y)) = (0, -1)$ (voir Figure 4.2). De plus, d'après le Lemme 4.2.3, les arbres S_1 et S_0 sont de même hauteur. D'après le Lemme 4.2.8, on en déduit qu'aucune autre modification des valeurs de déséquilibre de T_0 n'a lieu dans l'arbre T_1 . L'arbre T_1 est bien équilibré et comme l'opération de rotation conserve l'ordre canonique des nœuds, on a bien $\gamma_{T_0}(z) = \gamma_{T_1}(z)$ pour tout nœud d'indice canonique z différent de x et y . \square

Nous appelons *rotation conservatrice d'équilibre* toute rotation qui admet comme pivot initial l'un des deux pivots qui interviennent dans la Proposition 4.2.1. Ces pivots sont appelés *pivots conservateurs d'équilibre*. La Figure 4.4 résume l'opération de rotation appliquée à ces pivots de rotation.

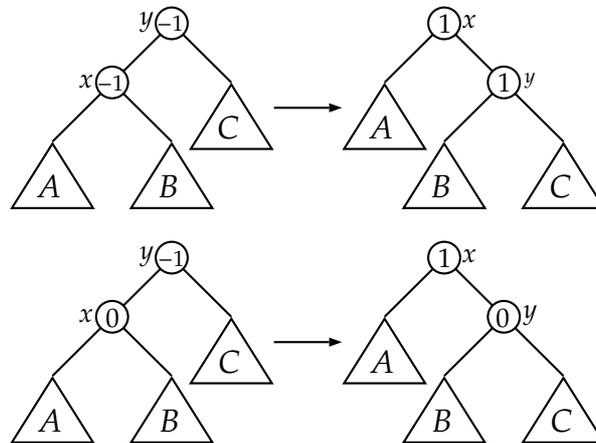


FIG. 4.4 – Rotations conservatrices d'équilibre.

On note $\text{IndiceRotEq}(T)$ l'ensemble des indices canoniques des nœuds de T qui sont racines de pivots de rotation conservateurs d'équilibre.

4.3 Une classification des arbres équilibrés dans le treillis de Tamari

Dans cette partie, nous établissons une classification des arbres équilibrés dans le treillis de Tamari selon la nature des arbres environnants, ou autrement dit des

couvertures pour l'ordre de Tamari ou des couvertures pour l'ordre dual de Tamari.

Nous montrons que pour chacune des classes que nous allons définir, nous pouvons donner un ensemble de motifs qui possèdent individuellement au plus deux nœuds permettant de décrire l'ensemble de ses éléments. Ceci donne un moyen de décider, étant donné un arbre équilibré, les classes auxquelles il appartient.

L'outillage relatif aux motifs de déséquilibre et des ensembles d'arbres évitant certains motifs que nous avons introduit dans les sections précédentes justifie maintenant de son utilité car il nous donne un moyen simple d'obtenir des équations fonctionnelles pour les séries génératrices de ces classes et donc de dénombrer leurs éléments de manière efficace.

4.3.1 Les classes d'arbres équilibrés

Définition 4.3.1. (Classe des arbres équilibrés maximaux)

T_0 est un *arbre équilibré maximal* si pour tout arbre binaire T_1 tel que $T_0 \prec T_1$ l'arbre T_1 n'est pas équilibré.

Proposition 4.3.1. (Description des arbres équilibrés maximaux)

T est un *arbre équilibré maximal* si et seulement si $T \in \mathcal{E}_\infty \cap \mathcal{N}(M)$ où

$$M = \{(11, -1 - 1), (11, 0 - 1)\}. \quad (4.25)$$

Preuve. (\Rightarrow) : par contraposée, supposons que T admette une occurrence du motif $(11, -1 - 1)$. Il existe donc dans T un pivot de rotation conservateur d'équilibre, et, d'après la Proposition 4.2.1, il existe un arbre équilibré T' tel que $T \prec T'$. L'arbre T n'est donc pas un arbre équilibré maximal. Le cas est similaire si T admet une occurrence du motif $(11, 0 - 1)$.

(\Leftarrow) : d'après la Proposition 4.2.1, comme l'arbre T évite les motifs $(11, -1 - 1)$ et $(11, 0 - 1)$, il n'admet aucun pivot de rotation conservateur d'équilibre. Tout arbre T' tel que $T \prec T'$ n'est pas équilibré. \square

Il est donc possible de décider facilement si un arbre équilibré est maximal en vérifiant qu'il est bien équilibré et qu'il ne possède pas d'occurrences des motifs $(11, -1 - 1)$ et $(11, 0 - 1)$.

Cette classe d'arbres équilibrés est la classe duale des arbres équilibrés maximaux :

Définition 4.3.2. (Classe des arbres équilibrés minimaux)

T_1 est un *arbre équilibré minimal* si pour tout arbre binaire T_0 tel que $T_0 \prec T_1$ l'arbre T_0 n'est pas équilibré.

Proposition 4.3.2. (Description des arbres équilibrés minimaux)

T est un *arbre équilibré minimal* si et seulement si $T \in \mathcal{E}_\infty \cap \mathcal{N}(M)$ où

$$M = \{(21, 11), (21, 10)\}. \quad (4.26)$$

Preuve. La preuve est analogue à celle de la Proposition 4.3.1 en considérant cette fois-ci les rotations gauches à la place des rotations droites. En effet, si T est un arbre équilibré minimal, alors toute rotation gauche le déséquilibre. Il peut donc être considéré comme maximal dans l'ordre dual de Tamari. \square

Nous pouvons ici aussi décider facilement si un arbre équilibré est minimal en vérifiant qu'il est équilibré et qu'il ne possède pas d'occurrences des motifs $(21, 11)$ et $(21, 10)$.

Définition 4.3.3. (Classe des arbres équilibrés intérieurs à droite)

T_0 est un *arbre équilibré intérieur à droite* si pour tout arbre binaire T_1 tel que $T_0 \prec T_1$ l'arbre T_1 est équilibré.

Proposition 4.3.3. (Description des arbres équilibrés intérieurs à droite)

T est un *arbre équilibré intérieur à droite* si et seulement si $T \in \mathcal{E}_\infty \cap \mathcal{N}(M)$ où

$$M = \{(11, -10), (11, -1-1), (11, 00), (11, 01), (11, 1-1), (11, 10), (11, 11)\}. \quad (4.27)$$

Preuve. (\Rightarrow) : par contraposée, supposons que T n'évite pas l'un des motifs de M . D'après la Proposition 4.2.1, la rotation autour d'un pivot égal à l'un de ces motifs non évité crée un déséquilibre dans l'arbre.

(\Leftarrow) : en supposant que T évite les motifs de M , comme il est équilibré, s'il admet des pivots de rotation, ils sont de la forme $S = (A \wedge B) \wedge C$ avec $(\gamma_S(x), \gamma_S(y)) \in \{(0, -1), (-1, -1)\}$. où y est l'indice canonique du nœud racine de S et x l'indice canonique du nœud racine de $(A \wedge B)$. D'après la Proposition 4.2.1, toute rotation autour de l'un de ces pivots conserve l'équilibre. Toute rotation appliquée à T produit donc un arbre équilibré. \square

Définition 4.3.4. (Classe des arbres équilibrés intérieurs à gauche)

T_1 est un *arbre équilibré intérieur à gauche* si pour tout arbre binaire T_0 tel que $T_0 \prec T_1$ l'arbre T_0 est équilibré.

Proposition 4.3.4. (Description des arbres équilibrés intérieurs à gauche)

T est un *arbre équilibré intérieur à gauche* si et seulement si $T \in \mathcal{E}_\infty \cap \mathcal{N}(M)$ où

$$M = \{(21, -1-1), (21, 1-1), (21, 0-1), (21, 00), (21, 01), (21, 1-1), (21, 10)\}. \quad (4.28)$$

Preuve. La preuve est analogue à celle de la Proposition 4.3.3 en considérant les rotations gauches à la place des rotations droites. \square

Nous donnons la définition des deux classes d'arbres équilibrés suivantes sans les détailler d'avantage :

Définition 4.3.5. (Classe des arbres équilibrés mixtes à droite)

T_0 est un *arbre équilibré mixte à droite* s'il existe un arbre équilibré T_1 et un arbre non équilibré T_2 tels que $T_0 \prec T_1$ et $T_0 \prec T_2$.

Définition 4.3.6. (Classe des arbres équilibrés mixtes à gauche)

T_2 est un *arbre équilibré mixte à gauche* s'il existe un arbre équilibré T_1 et un arbre non équilibré T_0 tels que $T_0 \prec T_2$ et $T_1 \prec T_2$.

4.3.2 Dénombrement des classes d'arbres équilibrés

Commençons en guise d'introduction à cette partie en expliquant la manière d'obtenir une équation fonctionnelle qui admet comme point fixe la série génératrice qui compte les arbres équilibrés[12, 6].

L'idée est d'engendrer récursivement les arbres équilibrés selon leur hauteur. Pour cela, on engendre un ensemble d'objets plus généraux que les arbres, objets que l'on nomme *arbres à bourgeons*. Ce sont des arbres binaires avec la particularité qu'ils ne possèdent pas de feuilles et l'ensemble de ses nœuds externes sont des *bourgeons*. Un arbre à bourgeon *pousse* par des substitution simultanées de ses bourgeons en de nouveaux arbres à bourgeons. Les arbres binaires équilibrés sont finalement

4.3. UNE CLASSIFICATION DES ARBRES ÉQUILIBRÉS DANS LE TREILLIS DE TAMARI

obtenus en remplaçant les bourgeons des arbres à bourgeons par des feuilles. Les règles de substitution qui permettent d'engendrer les arbres équilibrés sont

$$\begin{array}{c} \textcircled{x} \\ \textcircled{y} \end{array} \longrightarrow \begin{array}{c} \textcircled{-1} \\ \textcircled{x} \quad \textcircled{y} \end{array} + \begin{array}{c} \textcircled{0} \\ \textcircled{x} \quad \textcircled{x} \end{array} + \begin{array}{c} \textcircled{1} \\ \textcircled{y} \quad \textcircled{x} \end{array} \quad (4.29)$$

$$\textcircled{y} \longrightarrow \textcircled{x} \quad (4.30)$$

Le rôle du bourgeon \textcircled{x} est d'engendrer un nœud qui possède -1 , 0 ou 1 comme valeur de déséquilibre, les seules valeurs de déséquilibre autorisées pour les arbres équilibrés. Le rôle du bourgeon \textcircled{y} est de retarder la croissance de l'arbre pour permettre la création des déséquilibres -1 et 1 . Chaque étape de substitution incrémente la hauteur de l'arbre en croissance d'une unité. Lorsqu'au cours des substitutions successives la structure engendrée ne contient aucun \textcircled{y} , nous pouvons alors considérer les bourgeons \textcircled{x} restants comme des feuilles, et donc la structure comme un arbre binaire. Le nombre de \textcircled{x} dans une structure est donc le nombre de feuilles de l'arbre engendré. Les structures qui possèdent encore des \textcircled{y} n'ont pas fini de pousser, elles doivent être ignorées.

De ces règles de substitution nous pouvons déduire le résultat suivant [12, 6] :

Théorème 4.3.1. (Série génératrice des arbres équilibrés)

La série génératrice qui compte le nombre d'arbres équilibrés selon leur nombre de feuilles est $E(x) = A(x, 0)$ où

$$A(x, y) = x + A(x^2 + 2xy, x). \quad (4.31)$$

La résolution, ou autrement dit, l'extraction des coefficients, pour ce type d'équation fonctionnelle se fait par itération. On procède en calculant la suite de polynômes $(A_i)_{i \geq 0}$ définie de la manière suivante :

$$A_i(x, y) = \begin{cases} x & \text{si } i = 0, \\ x + A_{i-1}(x, y) & \text{sinon.} \end{cases} \quad (4.32)$$

Les premières itérations sont

$$\begin{aligned} A_0 &= x, \\ A_1 &= x + 2xy + x^2, \\ A_2 &= x + 2xy + x^2 + 4x^2y + 2x^3 + 4x^2y^2 + 4x^3y + x^4, \\ A_3 &= x + 2xy + x^2 + 4x^2y + 2x^3 + 4x^2y^2 + 4x^3y + x^4 + 16x^3y^2 + 16x^4y + 4x^5 + \\ &\quad 16x^3y^3 + 40x^4y^2 + 28x^5y + 6x^6 + 32x^4y^3 + 48x^5y^2 + 24x^6y + 4x^7 + 16x^4y^4 + \\ &\quad 32x^5y^3 + 24x^6y^2 + 8x^7y + x^8. \end{aligned}$$

Le point fixe de la suite $(A_i)_{i \geq 0}$ après substitution de 0 au paramètre y est la suite du nombre des arbres équilibrés comptés selon leur nombre de feuilles.

Comptons maintenant le nombre d'arbres équilibrés maximaux. On utilise pour cela la même idée de comptage que celle que nous venons de rappeler pour compter les arbres équilibrés. La différence cette fois-ci est que, conformément à la Proposition 4.3.1, les arbres équilibrés maximaux n'admettent aucune occurrence des motifs $(11, -1 - 1)$ et $(11, 0 - 1)$. Il faut donc contrôler la pousse du bourgeon \textcircled{x} lorsque il génère un arbre S dont la racine est de déséquilibre -1 . En effet, si

le nœud fils gauche de S pousse avec un déséquilibre -1 ou 0 , l'arbre S contient une occurrence de l'un des motifs qui doivent être évités. L'idée est donc de forcer le déséquilibre du nœud fils gauche de S à 1 . On utilise pour cela les règles de substitution suivantes :

$$\begin{array}{c} \textcircled{x} \\ \textcircled{y} \end{array} \longrightarrow \begin{array}{c} \textcircled{0} \\ \textcircled{x} \quad \textcircled{x} \end{array} + \begin{array}{c} \textcircled{1} \\ \textcircled{y} \quad \textcircled{x} \end{array} + \begin{array}{c} \textcircled{-1} \\ \textcircled{z} \quad \textcircled{y} \end{array} \quad (4.33)$$

$$\textcircled{y} \longrightarrow \textcircled{x} \quad (4.34)$$

$$\textcircled{z} \longrightarrow \begin{array}{c} \textcircled{1} \\ \textcircled{y} \quad \textcircled{x} \end{array} \quad (4.35)$$

Le bourgeon \textcircled{z} sert à engendrer des arbres équilibrés maximaux dont la racine est de déséquilibre 1 . En traduisant ces substitutions en série génératrice, nous obtenons la proposition suivante :

Proposition 4.3.5. (Série génératrice des arbres équilibrés maximaux)

La série génératrice qui compte le nombre d'arbres équilibrés maximaux selon leur nombre de feuilles est $M(x) = A(x, 0, 0)$ où

$$A(x, y, z) = x + A(x^2 + xy + yz, x, xy). \quad (4.36)$$

Preuve. La preuve découle directement de la validité des substitutions que nous avons établies. Il est clair que les arbres engendrés par ces substitutions sont équilibrés et que toute occurrence du motif $(1, -1)$ force à l'étape suivante la pousse d'un arbre de racine de déséquilibre à 1 comme fils gauche du nœud qui correspond à ce motif. Enfin, il est possible de faire pousser tous les autres motifs différents des deux interdits décrits dans la Proposition 4.3.1. \square

La résolution de cette équation fonctionnelle par itération nous donne $1, 1, 1, 1, 2, 2, 2, 4, 6, 9, 11, 13, 22, 38, 60, 89, 128, 183, 256, 353, 512, 805, 1336, 2221, \dots$ comme premières valeurs.

4.4 Les intervalles d'arbres équilibrés

Nous obtenons une caractérisation intéressante des intervalles formés par les arbres équilibrés. Ces intervalles possèdent en effet la forme d'hypercubes. Nous commençons par rappeler la définition de l'hypercube vu comme un treillis. Nous donnons ensuite la preuve de ce résultat et donnons un moyen de calculer, étant donné un arbre équilibré T , les dimensions des hypercubes où T joue le rôle du plus petit élément. De plus, étant donné que tout intervalle du treillis hypercube est gradué, nous déduisons que tout intervalle d'arbres équilibrés est gradué et nous donnons la fonction de graduation qui associe à un arbre équilibré T_1 son grade par rapport à un arbre équilibré T_0 dans l'hypercube formé par l'intervalle d'arbres équilibrés $[T_0, T_1]$. Nous terminons en donnant l'équation fonctionnelle qui admet comme point fixe la série génératrice qui compte le nombre d'intervalles d'arbres équilibrés puis celle qui compte le nombre d'hypercubes maximaux.

4.4.1 Description des intervalles d'arbres équilibrés

Comme annoncé, nous définissons l'hypercube en terme d'ensemble partiellement ordonné :

Définition 4.4.1. (Hypercube)

Nous appelons *hypercube* d'ordre k (ou de dimension k) l'ensemble partiellement ordonné défini sur l'ensemble des mots de longueur k sur l'alphabet $\{0, 1\}$ muni de la relation d'ordre \leq définie par $0 \leq 1$ et :

$$(u \leq v) \Leftrightarrow (\forall 1 \leq i \leq k, u[i] \leq v[i]). \quad (4.37)$$

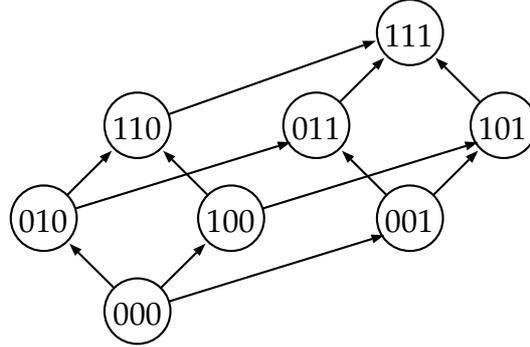


FIG. 4.5 – L'hypercube d'ordre 3.

On note \mathcal{H}_k l'hypercube défini sur des mots de longueur k . Il est clair qu'il possède 2^k éléments et que l'élément 0^k (resp. 1^k) est le plus petit élément (resp. le plus grand élément). L'hypercube possède une structure de treillis. Il vérifie la propriété suivante :

Proposition 4.4.1. (Intervalles de l'hypercube)

Pour tout $k \geq 0$ et tout intervalle $I = [u, v]$ de \mathcal{H}_k , il existe un entier $j \leq k$ tel que I est isomorphe à \mathcal{H}_j .

Preuve. Considérons l'ensemble J composé des éléments w de I auxquels on efface les lettres aux positions i lorsque $w[i] = u[i] = 1$ pour $1 \leq i \leq k$. L'ensemble J est clairement isomorphe au treillis \mathcal{H}_j où j est le nombre de 0 dans u . \square

Il est de plus facilement démontrable par récurrence que \mathcal{H}_k admet 3^k intervalles.

Rappelons que $\text{IndiceRotEq}(T)$ désigne l'ensemble des indices canoniques des nœuds de l'arbre T qui sont racines de pivots de rotation conservateurs d'équilibre. Établissons ce premier lemme, important pour la suite, exprimant le fait que toute rotation conservatrice d'équilibre ne crée pas de nouveau pivot de rotation conservateur d'équilibre.

Lemme 4.4.1. (Aucune création de pivots de rotation conservateurs d'équilibre)

Si $[T_0, T_1]$ est un intervalle d'arbres équilibrés, alors $\text{IndiceRotEq}(T_1) \subseteq \text{IndiceRotEq}(T_0)$.

Preuve. Par récurrence sur l'entier $k \geq 0$ tel que l'on a les relations

$$T_0 \prec T'_1 \prec \dots \prec T'_{k-1} \prec T_1 \quad (4.38)$$

où $(T'_i)_{1 \leq i \leq k-1}$ est une suite d'arbres équilibrés.

Si $k = 0$, nous avons $T_0 = T_1$ et le lemme est trivialement vrai.

Si $k = 1$, nous avons $T_0 \prec T_1$ et donc $T_1 \in \text{Rot}(T_0)$. Comme T_0 et T_1 sont des arbres

équilibrés, la rotation qui transforme T_0 en T_1 est conservatrice d'équilibre et possède donc comme pivot de rotation le motif $(11, -1 - 1)$ ou le motif $(11, 0 - 1)$. Supposons en premier lieu que le pivot de rotation est $(11, -1 - 1)$. Il se retrouve modifié en $(21, 11)$ dans l'arbre T_1 . Clairement, aucun nouveau pivot de rotation conservateur d'équilibre n'est créé puisque aucun déséquilibre à -1 n'est créé, et les déséquilibres modifiés deviennent égaux à 1 , sans pour autant créer de nouveau pivot de rotation conservateur d'équilibre puisque aucun déséquilibre d'un tel pivot de rotation n'est égal à 1 . Dans l'autre cas où le pivot de rotation est $(11, 0 - 1)$, il se retrouve modifié en $(21, 10)$ dans T_1 . Les mêmes arguments sont valables que dans le cas précédent mais il est nécessaire d'ajouter que le motif $(1, 0)$ créé dans T_1 ne peut en aucun cas faire partie d'un nouveau pivot de rotation conservateur d'équilibre puisque le nœud correspondant à ce motif est nécessairement fils droit d'un nœud de T_1 . Un motif $(1, 0)$ peut en effet appartenir à un pivot de rotation conservateur d'équilibre mais uniquement en temps que fils gauche d'un nœud dont le déséquilibre est à -1 . On conclut dans ces deux cas en disant que l'opération de rotation respecte l'étiquetage canonique des nœuds, et donc que les pivots de rotation conservateurs d'équilibre non modifiés possèdent les mêmes indices canoniques dans T_0 et T_1 . Supposons maintenant le lemme vrai pour un certain entier k . Nous avons donc $\text{IndiceRotEq}(T_1) \subseteq \text{IndiceRotEq}(T_0)$ et considérons l'arbre équilibré T_2 tel que $T_1 \prec T_2$ et $[T_0, T_2]$ intervalle d'arbres équilibrés. Par hypothèse d'induction nous avons $\text{IndiceRotEq}(T_1) \subseteq \text{IndiceRotEq}(T_0)$ et par ce que nous venons de montrer pour le cas $k = 1$, nous avons $\text{IndiceRotEq}(T_2) \subseteq \text{IndiceRotEq}(T_1)$. Nous avons donc $\text{IndiceRotEq}(T_2) \subseteq \text{IndiceRotEq}(T_0)$ par transitivité de l'inclusion ensembliste. \square

Nous définissons une nouvelle structure arborescente qui permet de repérer les pivots de rotation conservateurs d'équilibre ainsi que ceux qui ont subi une rotation :

Définition 4.4.2. (Cristallisation d'un arbre équilibré)

On appelle *cristallisation* d'un arbre équilibré T tout objet obtenu à partir de T en remplaçant des occurrences des motifs $(11, -1 - 1)$ ou $(11, 0 - 1)$ par un *losange* (voir la Figure 4.6) et des occurrences des motifs $(21, 11)$ ou $(21, 10)$ par un *carré* (voir la Figure 4.7).

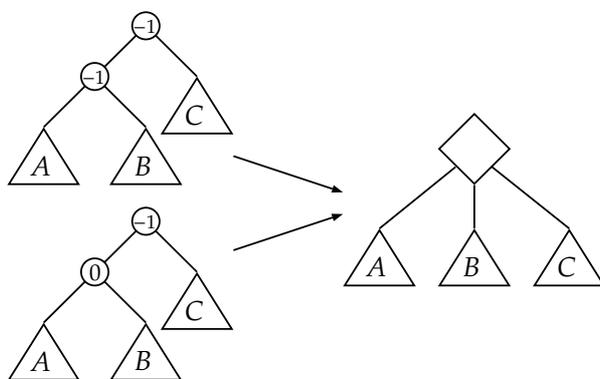


FIG. 4.6 – Codage des pivots conservateurs d'équilibre.

Étant donné un arbre équilibré T , on note $\text{Cris}(T)$ l'ensemble des cristallisations de T . On dit d'une cristallisation qu'elle est *complète* si elle ne possède plus aucune

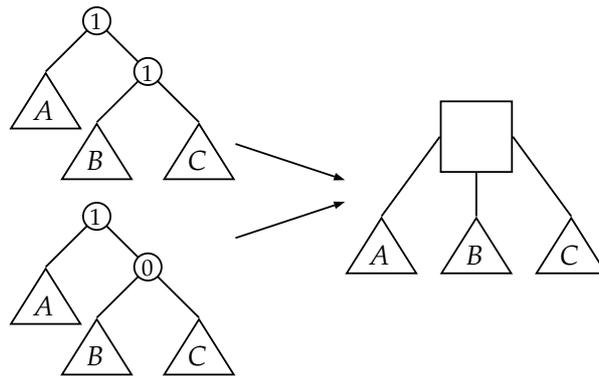
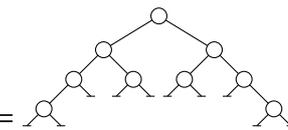
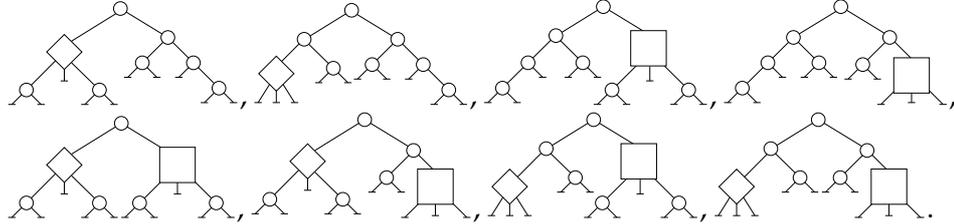


FIG. 4.7 – Codage des pivots conservateurs d'équilibre après rotation.

occurrence des motifs $(11, -1 - 1)$, $(11, 0 - 1)$, $(21, 11)$ et $(21, 10)$. Il peut exister plusieurs cristallisations complètes associées à un même arbre. Il suffit en effet que deux pivots de rotation conservateurs d'équilibre se chevauchent pour donner lieu à plusieurs cristallisations complètes différentes.

Exemple.

Soit l'arbre $T =$ . Il admet les cristallisations suivantes :



Les quatre dernières sont complètes.

D'après la Proposition 4.3.1, un arbre appartient à la classe des arbres équilibrés maximaux si toute cristallisation $C \in \text{Cris}(T)$ ne possède pas de losange. Symétriquement, d'après la Proposition 4.3.2, un arbre appartient à la classe des arbres équilibrés minimaux si toute cristallisation $C \in \text{Cris}(T)$ ne possède pas de carré.

Le but principal de cette représentation est que les arbres résultants d'une rotation conservatrice d'équilibre se calculent simplement en substituant un carré à un losange sans rien modifier dans la structure par ailleurs.

On définit une relation d'équivalence \asymp sur les cristallisations comme suit : nous avons $C_0 \asymp C_1$ si et seulement si C_0 et C_1 sont identiques lorsque l'on substitue tous les losanges de C_0 et C_1 par des carrés.

Lemme 4.4.2. (Cristallisation d'un intervalle)

Si $[T_0, T_1]$ est un intervalle d'arbres équilibrés, alors il existe $C_0 \in \text{Cris}(T_0)$ et $C_1 \in \text{Cris}(T_1)$ tels que $C_0 \asymp C_1$.

Preuve. Par récurrence sur l'entier $k \geq 0$ tel que l'on a les relations

$$T_0 \prec T'_1 \prec \dots \prec T'_{k-1} \prec T_1 \tag{4.39}$$

où $(T'_i)_{1 \leq i \leq k-1}$ est une suite d'arbres équilibrés.

Si $k = 0$, nous avons $T_0 = T_1$ et le lemme est trivialement vrai.

Si $k = 1$, nous avons $T_0 \prec T_1$ et donc $T_1 \in \text{Rot}(T_0)$. Ceci signifie qu'il est possible de transformer l'arbre T_0 en l'arbre T_1 en effectuant une rotation conservatrice d'équilibre. Il existe donc une cristallisation C_0 de T_0 qui modélise le pivot de cette rotation par un losange. La cristallisation C_1 de T_1 résultante de la rotation autour de ce pivot est obtenue en remplaçant le losange de C_0 par un carré. Nous avons bien $C_0 \asymp C_1$.

Supposons le lemme vrai pour un entier k . Nous avons donc C_0 cristallisation de T_0 et C_1 cristallisation de T_1 tels que $C_0 \asymp C_1$. Considérons un arbre T_2 tel que $T_1 \prec T_2$ et $[T_0, T_2]$ est un intervalle d'arbres équilibrés. D'après le Lemme 4.4.1, le pivot de la rotation qui transforme T_1 en T_2 existe dans T_0 . De plus, il existe nécessairement dans T_1 . En considérant la cristallisation C'_0 modifiée en remplaçant dans C_0 (si nécessaire) le pivot de la rotation qui transforme T_1 en T_2 par un losange, et, en posant C_2 la cristallisation associée à T_2 obtenue en remplaçant dans C_1 ce même pivot par un carré, nous avons bien $C'_0 \asymp C_2$. \square

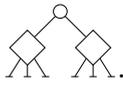
Soit $I = [T_0, T_1]$ un intervalle d'arbres équilibrés. On appelle *cristallisation de l'intervalle* I la cristallisation C_I telle que :

- $C_I \in \text{Cris}(T_0)$,
- C_I ne possède pas de carrés,
- il existe $C_1 \in \text{Cris}(T_1)$ tel que $C_I \asymp C_1$ et C_1 ne possède pas losanges.

La cristallisation C_I existe d'après le Lemme 4.4.2. Un intervalle d'arbres équilibrés possède clairement une unique cristallisation.

Exemple.

Considérons l'intervalle $I = \left[\begin{array}{c} \text{arbre 1} \\ \text{arbre 2} \end{array}, \begin{array}{c} \text{arbre 3} \\ \text{arbre 4} \end{array} \right]$, composé, ce qui est facilement

vérifiable, d'arbres équilibrés. La cristallisation de I est .

Lemme 4.4.3. (Cristallisation équivalentes pour les éléments d'un intervalle)

Si $I = [T_0, T_2]$ est un intervalle d'arbres équilibrés, C_I la cristallisation de l'intervalle I et $T_1 \in I$, alors il existe $C_1 \in \text{Cris}(T_1)$ tel que $C_1 \asymp C_I$.

Preuve. D'après le Lemme 4.4.2, la cristallisation de l'intervalle I existe bien, et d'après le Lemme 4.4.1, tous les pivots de rotation conservateurs d'équilibre dans T_1 existent dans T_0 et ceux de T_2 existent dans T_1 . Nous pouvons donc construire une cristallisation C_1 de T_1 qui satisfait $C_1 \asymp C_I$. \square

Une cristallisation d'un intervalle encode donc exactement un intervalle d'arbres équilibrés.

Voici le résultat principal de cette partie que nous avons établi :

Théoreme 4.4.1. (Intervalles d'arbres équilibrés et hypercubes)

Pour tout $n \geq 0$ et tout intervalle $I = [T_0, T_1] \subseteq \mathcal{E}_n$ du treillis de Tamari composé exclusivement d'arbres équilibrés, il existe un entier $k \geq 0$ tel que l'ensemble partiellement ordonné (I, \preceq) est isomorphe à l'hypercube \mathcal{H}_k .

Preuve. On associe un mot sur l'alphabet $\{0, 1\}$ à une cristallisation C de la manière suivante : en considérant les losanges et les carrés de C de gauche à droite dans sa structure arborescente, on écrit un 0 pour chacun de ses losanges et un 1 pour chacun de ses carrés.

4.4. LES INTERVALLES D'ARBRES ÉQUILIBRÉS

Posons C_I la cristallisation de l'intervalle I . D'après le Lemme 4.4.3, pour tout arbre $T_2 \in I$, il existe une cristallisation $C_2 \in \text{Cris}(T_2)$ qui est obtenue en remplaçant successivement des losanges par des carrés dans C_I . Le mot associé à C_2 est tout simplement le mot C_I dans lequel certains 0 ont été remplacés par des 1. La relation de couverture est bien celle de l'hypercube : un 0 remplacé par un 1 traduit le fait qu'une rotation conservatrice d'équilibre modifie un losange en un carré sans modifier l'ordre canonique des nœuds.

L'intervalle I est donc isomorphe à l'hypercube \mathcal{H}_k où k est le nombre de losanges dans C_I . □

Nous avons les corollaires suivants :

Corollaire 4.4.1. (Fonction de graduation)

Tout intervalle I d'arbres équilibrés est gradué et sa fonction de graduation g_I est définie pour tout arbre $T \in I$ par :

$$g_I(T) = \text{nombre de carrés dans } C_T \in \text{Cris}(T) \text{ tel que } C_T \asymp C_I. \quad (4.40)$$

Corollaire 4.4.2. (Dimension de l'hypercube associé à un arbre équilibré)

Si $I = [T_0, T_1]$ est un intervalle d'arbres équilibrés et la cristallisation de I possède k losanges, alors T_0 est le plus petit élément de l'hypercube \mathcal{H}_k

La Figure 4.8 illustre les diagrammes de Hasse des intervalles d'arbres équilibrés dans les treillis de Tamari \mathcal{T}_n pour $0 \leq n \leq 11$.

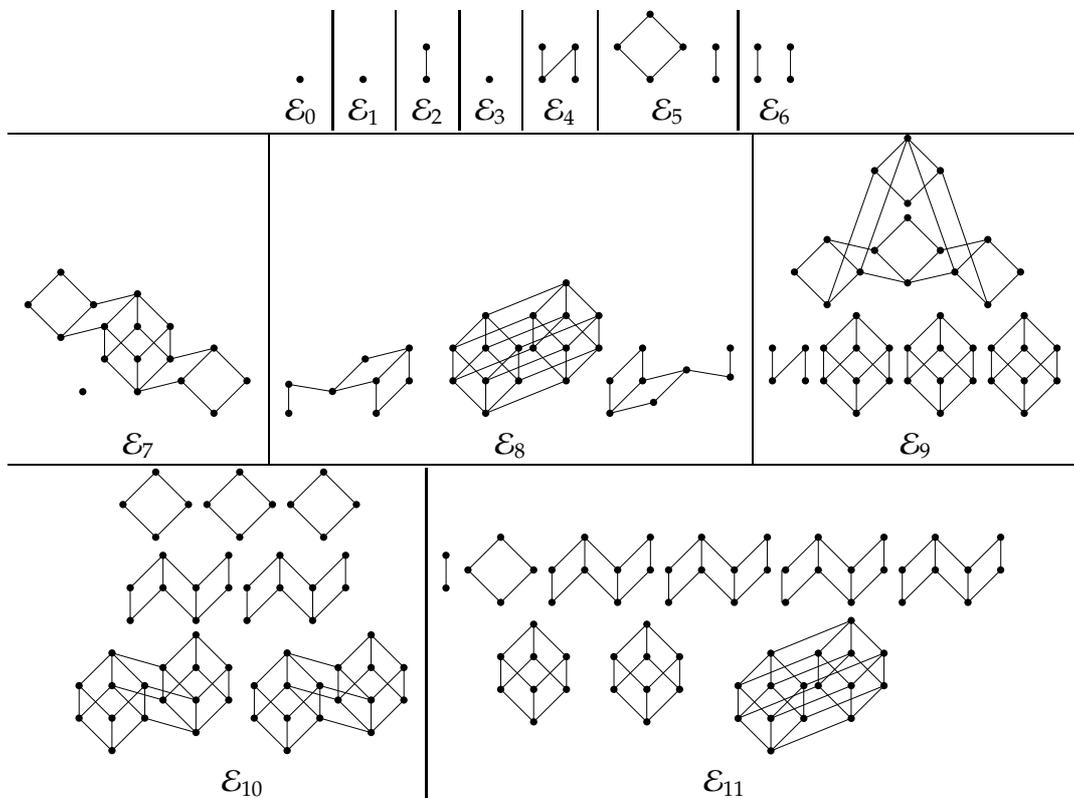


FIG. 4.8 – Diagrammes de Hasse des intervalles d'arbres équilibrés des premiers treillis de Tamari.

4.4.2 Dénombrément des intervalles d'arbres équilibrés

Comptons le nombre d'intervalles d'arbres équilibrés.

Proposition 4.4.2. (Série génératrice des intervalles d'arbres équilibrés)

La série génératrice qui compte le nombre d'intervalles d'arbres équilibrés selon le nombre de feuilles des arbres qui les composent dans les treillis de Tamari est $M(x) = A(x, 0, 0)$ où

$$A(x, y, z) = x + A(x^2 + 2xy + z, x, x^3 + x^2y). \quad (4.41)$$

Preuve. Le Lemme 4.4.2 et le Lemme 4.4.2 montrent qu'un intervalle d'arbres équilibrés est encodé par exactement une cristallisation qui ne contient pas de carrés. Réciproquement, il est clair qu'une cristallisation C sans carrés encode exactement un intervalle d'arbres équilibrés : c 'est l'ensemble des arbres que l'on obtient en remplaçant des losanges de C par des carrés. Intéressons-nous à engendrer l'ensemble des cristallisations sans carrés. Suivant la même idée que celle que nous avons utilisée pour dénombrer les arbres équilibrés et la classes des arbres équilibrés maximaux, posons les règles de substitution suivantes :

$$\begin{array}{c} \textcircled{x} \\ \textcircled{y} \end{array} \longrightarrow \begin{array}{c} \textcircled{-1} \\ \textcircled{x} \quad \textcircled{y} \end{array} + \begin{array}{c} \textcircled{0} \\ \textcircled{x} \quad \textcircled{x} \end{array} + \begin{array}{c} \textcircled{1} \\ \textcircled{y} \quad \textcircled{x} \end{array} + \textcircled{z} \quad (4.42)$$

$$\textcircled{y} \longrightarrow \textcircled{x} \quad (4.43)$$

$$\textcircled{z} \longrightarrow \begin{array}{c} \diamond \\ \textcircled{x} \quad \textcircled{x} \quad \textcircled{x} \end{array} + \begin{array}{c} \diamond \\ \textcircled{x} \quad \textcircled{y} \quad \textcircled{x} \end{array} \quad (4.44)$$

Le bourgeon \textcircled{x} permet de faire pousser un arbre dont la racine est de déséquilibré $-1, 0$ ou 1 mais aussi un bourgeon \textcircled{z} . Ce dernier bourgeon contrôle la pousse des losanges. Le premier losange encode le pivot de rotation $(11, -1 - 1)$ et le deuxième le pivot de rotation $(11, 0 - 1)$. Le bourgeon \textcircled{y} est toujours utilisé pour retarder la croissance de l'arbre dont il est racine pour créer les déséquilibres à -1 et 1 .

La traduction de ces substitutions en équation fonctionnelle nous donne bien l'expression annoncée. \square

Le nombre d'intervalles d'arbres équilibrés pour les premiers treillis de Tamari est $1, 1, 3, 1, 7, 12, 6, 52, 119, 137, 195, 231, 1019, 3503, 6593, 12616, 26178, 43500, 64157, 94688, 232560, 817757, 2233757, 5179734, \dots$

Comptons maintenant le nombre d'intervalles $[T_0, T_1]$ d'arbres équilibrés où T_0 est un arbre équilibré minimal et T_1 est un arbre équilibré maximal. Ces intervalles correspondent aux hypercubes maximaux, c'est à dire les hypercubes qui ne sont des intervalles propres d'aucun autre hypercube.

Proposition 4.4.3. (Série génératrice des hypercubes maximaux)

La série génératrice qui compte le nombre d'intervalles d'arbres équilibrés $[T_0, T_1]$ avec T_0 arbre équilibré minimal et T_1 arbre équilibré maximal selon le nombre de feuilles des arbres qui les composent dans les treillis de Tamari est $M(x) = A(x, 0, 0, 0)$ où

$$A(x, y, z, t) = x + A(x^2 + 2yz + t, x, yz + t, x^3 + x^2y). \quad (4.45)$$

4.4. LES INTERVALLES D'ARBRES ÉQUILIBRÉS

Preuve. Il suffit de considérer les cristallisations associées aux arbres équilibrés minimaux qui sont complètes et qui ne contiennent pas de carré. Ce sont des cristallisations d'intervalles $[T_0, T_1]$ où T_0 est un arbre équilibré minimal et T_1 un arbre équilibré maximal. De tels objets sont engendrés par les règles de substitution suivantes :

$$\begin{array}{c} \textcircled{x} \\ \textcircled{y} \end{array} \longrightarrow \begin{array}{c} \textcircled{0} \\ \textcircled{x} \quad \textcircled{x} \\ \textcircled{1} \\ \textcircled{y} \quad \textcircled{z_1} \\ \textcircled{-1} \\ \textcircled{z_2} \quad \textcircled{y} \\ t \end{array} + \textcircled{t} \quad (4.46)$$

$$\textcircled{y} \longrightarrow \textcircled{x} \quad (4.47)$$

$$\textcircled{z_1} \longrightarrow \begin{array}{c} \textcircled{-1} \\ \textcircled{z_2} \quad \textcircled{y} \\ t \end{array} + \textcircled{t} \quad (4.48)$$

$$\textcircled{z_2} \longrightarrow \begin{array}{c} \textcircled{1} \\ \textcircled{y} \quad \textcircled{z_1} \\ t \end{array} + \textcircled{t} \quad (4.49)$$

$$\textcircled{t} \longrightarrow \begin{array}{c} \diamond \\ \textcircled{x} \quad \textcircled{x} \\ \textcircled{x} \end{array} + \begin{array}{c} \diamond \\ \textcircled{x} \quad \textcircled{y} \\ \textcircled{x} \end{array} \quad (4.50)$$

Ces substitutions assurent que l'objet généré ne contient ni les motifs (21, 11) et (11, 10) parce que la cristallisation générée est celle d'un arbre équilibré minimal ni (11, -1-1) et (11, 0-1) parce que la cristallisation est complète. De plus, comme les bourgeons $\textcircled{z_1}$ et $\textcircled{z_2}$ jouent un rôle analogue, on simplifie l'équation fonctionnelle sous-jacente à ces règles de substitution en confondant les paramètres z_1 et z_2 en un unique paramètre z . \square

Les premières valeurs sont 1, 1, 1, 1, 3, 2, 2, 6, 9, 15, 15, 17, 41, 77, 125, 178, 252, 376, 531, 740, 1192, 2179, 4273, 7738, 13012, 20776, 32389, 49841, 75457, 113011, ...

Nous pouvons facilement modifier l'équation fonctionnelle obtenue dans la Proposition 4.4.3 pour compter également en fonction de la dimension des hypercubes maximaux. Il suffit pour cela d'ajouter un paramètre formel, que nous notons w , qui compte le nombre de losanges que contient la cristallisation d'intervalle engendrée. Ce nombre est la dimension de l'hypercube correspondant à cet intervalle. Le coefficient de $x^{n+1}w^k$ de la fonction génératrice ainsi obtenue est égal au nombre d'hypercubes maximaux de dimension k formés par les intervalles d'arbres équilibrés dans \mathcal{T}_n . La série génératrice $M(x, w)$ qui compte le nombre d'hypercubes maximaux avec leur dimension vérifie l'équation fonctionnelle $M(x, w) = A(x, 0, 0, 0, w)$ où

$$A(x, y, z, t, w) = x + A(x^2 + 2yz + t, x, yz + t, x^3w + x^2yw, w). \quad (4.51)$$

Les premiers coefficients de x la série génératrice $M(x, w)$ sont

$$[x^1]M(x, w) = 1, \quad (4.52)$$

$$[x^2]M(x, w) = 1, \quad (4.53)$$

$$[x^3]M(x, w) = w, \quad (4.54)$$

$$[x^4]M(x, w) = 1, \quad (4.55)$$

$$[x^5]M(x, w) = 3w, \quad (4.56)$$

$$[x^6]M(x, w) = w + w^2, \quad (4.57)$$

$$[x^7]M(x, w) = 2w, \quad (4.58)$$

$$[x^8]M(x, w) = 1 + 4w^2 + w^3, \quad (4.59)$$

$$[x^9]M(x, w) = 4w + 4w^2 + w^4, \quad (4.60)$$

$$[x^{10}]M(x, w) = 3w + 9w^2 + 3w^3, \quad (4.61)$$

$$[x^{11}]M(x, w) = 9w^2 + 6w^3, \quad (4.62)$$

$$[x^{12}]M(x, w) = w + 13w^2 + 2w^3 + w^4, \quad (4.63)$$

$$[x^{13}]M(x, w) = 6w + 4w^2 + 16w^3 + 15w^4, \quad (4.64)$$

$$[x^{14}]M(x, w) = 2w + 18w^2 + 31w^3 + 12w^4 + 14w^5, \quad (4.65)$$

$$[x^{15}]M(x, w) = 4w + 16w^2 + 24w^3 + 74w^4 + 4w^5 + 3w^6, \quad (4.66)$$

$$[x^{16}]M(x, w) = 1 + 16w^2 + 55w^3 + 55w^4 + 50w^5 + w^6, \quad (4.67)$$

$$[x^{17}]M(x, w) = 4w + 26w^2 + 44w^3 + 133w^4 + 26w^5 + 17w^6 + 2w^7, \quad (4.68)$$

$$[x^{18}]M(x, w) = 4w + 20w^2 + 113w^3 + 131w^4 + 89w^5 + 17w^6 + w^7 + w^8, \quad (4.69)$$

$$[x^{19}]M(x, w) = 39w^2 + 114w^3 + 228w^4 + 108w^5 + 36w^6 + 6w^7, \quad (4.70)$$

$$[x^{20}]M(x, w) = 3w + 9w^2 + 173w^3 + 343w^4 + 159w^5 + 39w^6 + 14w^7, \quad (4.71)$$

$$[x^{21}]M(x, w) = 17w^2 + 226w^3 + 382w^4 + 276w^5 + 234w^6 + 52w^7 + 5w^8. \quad (4.72)$$

À titre d'exemple, une interprétation possible de ces coefficients est que dans le treillis de Tamari d'ordre 11, les intervalles composés d'arbres équilibrés forment 1 hypercube maximal de dimension 1, 13 hypercubes maximaux de dimension 2, 2 hypercubes maximaux de dimension 3 et 1 hypercube maximal de dimension 4.

La série $M(x, w)$ ne donne en revanche aucune indication sur la morphologie des hypercubes et en particulier la manière dont ces derniers se collent entre-eux.

Chapitre 5

Conclusion

La conjecture résiste toujours à nos tentatives de preuve. Notre compréhension des intervalles d'arbres équilibrés est cependant devenue plus fine et une réponse a été fournie quant au rôle des arbres équilibrés dans l'ordre de Tamari. Beaucoup de questions se posent maintenant.

Nous pouvons nous demander de quelle façon les hypercubes formés par les intervalles d'arbres équilibrés se collent entre eux. Peut-on retrouver toutes configurations possibles ? Ou autrement dit, étant donné un ensemble d'hypercubes qui se partagent certains points, arrêtes, faces, cubes, ... existe-t-il un treillis de Tamari \mathcal{T}_n qui contient cet assemblage ?

Combien de composantes connexes d'intervalles d'arbres équilibrés compte le treillis de Tamari d'ordre n ?

Nous avons vu que l'obtention de l'équation fonctionnelle qui compte le nombre d'arbres équilibrés, d'arbres équilibrés maximaux, d'intervalles d'arbres équilibrés est étroitement liée aux motifs de déséquilibre évités par les objets à énumérer. Nous obtenons de cette façon un ensemble de règles de substitution qui se traduisent naturellement en équation fonctionnelle. Une question est de savoir s'il est possible de concevoir un algorithme, qui, à partir d'un ensemble M de motifs à éviter, donne l'équation fonctionnelle qui dénombre les éléments de $\mathcal{N}(M)$.

Bibliographie

- [1] Sage : Open source mathematics software, 2009.
- [2] G.M. Adelson-Velsky and E. M. Landis. An algorithm for the organization of information. *Soviet Mathematics Doklady*, 1962.
- [3] A. Aho and J. Ullman. *Concepts fondamentaux de l'informatique*. Dunod, 1993.
- [4] J. L. Baril and J. M. Pallo. The phagocyte lattice of dyck words. *Order*, 2006.
- [5] J. L. Baril and J. M. Pallo. The pruning-grafting lattice of binary trees. *Elsevier Science*, 2008.
- [6] F. Bergeron, G. Labelle, and P. Leroux. *Combinatorial Species and Tree-like Structures*. Cambridge University Press, 1994.
- [7] O. Bernardi and N. Bonichon. Catalan's intervals and realizers of triangulations. *À paraître, Journal of Combinatorial Theory*.
- [8] F. Chapoton. Sur le nombre d'intervalles dans les treillis de tamari. *Séminaire Lotharingien de Combinatoire*, 2006.
- [9] T.H. Cormen, C. E. Leiserson, R.L. Rivest, and C. Stein. *Introduction à l'algorithmique*. Dunod, 2004.
- [10] P. Dehornoy. On the rotation distance between binary trees. *Preprint*, 2009.
- [11] F. Hivert, J.-C. Novelli, and J.-Y. Thibon. The algebra of binary search trees. *Theoretical Computer Science*, 2003.
- [12] D. Knuth. *The Art of Computer Programming. Volume 3. Sorting and searching*. Addison Wesley Longman, 1998.
- [13] D. Knuth. *The Art of Computer Programming. Volume 4, Fascicle 4. Generating all trees - History of combinatorial generation*. Addison Wesley Longman, 2004.
- [14] G. Kreweras. Sur les partitions non croisées d'un cycle. *Discrete Mathematics*, 1972.
- [15] J.-L. Loday. Arithmetree. *Journal of Algebra*, 2002.
- [16] J.-L. Loday. Les multiples facettes de l'associaèdre. *Clay Mathematics Institute Publication*, 2005.
- [17] J.-L. Loday and M. Ronco. Order structure and the algebra of permutations and of planar binary trees. *Journal of Algebraic Combinatorics* 15(3), 2002.
- [18] J. M. Lucas, D. R. van Baronaigien, and F. Ruskey. On rotations and the generation of binary trees. *Journal of Algorithms*, 1993.
- [19] F. Ruskey. *Combinatorial Generation*. 2003.
- [20] D. Sleator, R. Tarjan, and W. Thurston. Rotation distance, triangulations, and hyperbolic geometry. *Journal of the American Mathematical Society*, 1 :647–681, 1988.

- [21] N. J. A. Sloane. The on-line encyclopedia of integer sequences, 2008.
- [22] R. P. Stanley. *Enumerative Combinatorics I*. Cambridge University Press, 1986.
- [23] R. P. Stanley. *Enumerative Combinatorics II*. Cambridge University Press, 1990.
- [24] R. P. Stanley. Catalan addendum. 2009.
- [25] V. Vajnovszki. Le codage des arbres binaires. *Computer Science Journal of Moldova*, 1995.