Balanced binary trees in the Tamari lattice

Samuele Giraudo

Balanced binary trees in the Tamari lattice

Samuele Giraudo[1]

[1]*Institut Gaspard Monge, Université Paris-Est Marne-la-Vallée, 5 Boulevard Descartes, Champs-sur-Marne, 77454 Marne-la-Vallée cedex 2, France*

Abstract. We show that the set of balanced binary trees is closed by interval in the Tamari lattice. We establish that the intervals $[T_0, T_1]$ where T_0 and T_1 are balanced trees are isomorphic as posets to a hypercube. We introduce tree patterns and synchronous grammars to get a functional equation of the generating series enumerating balanced tree intervals.

Résumé. Nous montrons que l'ensemble des arbres équilibrés est clos par intervalle dans le treillis de Tamari. Nous caractérisons la forme des intervalles du type $[T_0, T_1]$ où T_0 et T_1 sont équilibrés en montrant qu'en tant qu'ensembles partiellement ordonnés, ils sont isomorphes à un hypercube. Nous introduisons la notion de motif d'arbre et de grammaire synchrone dans le but d'établir une équation fonctionnelle de la série génératrice qui dénombre les intervalles d'arbres équilibrés.

Keywords: balanced trees, Tamari lattice, posets, grammars, generating series, combinatorics

1   Introduction

Binary search trees are used as data structures to represent dynamic totally ordered sets [7, 6, 3]. The algorithms solving classical related problems such as the insertion, the deletion or the search of a given element can be performed in a time logarithmic in the cardinality of the represented set, provided that the encoding binary tree is balanced. Recall that a binary tree is balanced if for each node x, the height of the left subtree of x and the height of the right subtree of x differ by at most one.

The algorithmic of balanced trees relies fundamentally on the so-called rotation operation. An insertion or a deletion of an element in a dynamic ordered set modifies the tree encoding it and can imbalance it. The efficiency of these algorithms comes from the fact that binary search trees can be rebalanced very quickly after the insertion or the deletion, using no more than two rotations [2].

Surprisingly, this operation appears in a different context since it defines a partial order on the set of binary trees of a given size. A tree T_0 is smaller than a tree T_1 if it is possible to transform the tree T_0 into the tree T_1 by performing a succession of right rotations. This partial order, known as the Tamari order [8, 10], defines a lattice structure on the set of binary trees of a given size.

Since binary trees are naturally equipped with this order structure induced by rotations, and the balance of balanced trees is maintained doing rotations, we would like to investigate if balanced trees play a particular role in the Tamari lattice. Our goal, in this is paper, is to combine the two points of view of the rotation operation. A first simple computer observation is that the intervals $[T_0,\ T_1]$ where T_0 and T_1 are balanced trees are only made up of balanced trees. The main goal of this paper is to prove this property. As a consequence, we give a characterization on the shape of these intervals and, using grammars allowing to generate trees, enumerate them.

This article is organized as follows. In Section 2, we set the essential notions about binary trees and balanced trees, and we give the definition of the Tamari lattice in our setting. Section 3 is devoted to establish the main result: the set of balanced trees is closed by interval in the Tamari lattice. In Section 4, we define tree patterns and synchronous grammars. These grammars allow us to generate trees avoiding a given set of tree patterns. We define a subset of balanced trees where elements hold a peculiar position in the Tamari lattice and we give, using the synchronous grammar generating these, a functional equation of the generating series enumerating these. Finally, in Section 5, we look at balanced tree intervals and show that they are, as posets, isomorphic to hypercubes. Encoding balanced tree intervals by particular trees, and establishing the synchronous grammar generating these trees, we give a functional equation satisfied by the generating series enumerating balanced tree intervals.

Acknowledgments

2   Preliminaries

2.1   Complete rooted planar binary trees

In this article, we consider complete rooted planar binary trees. Nodes are denoted by circles like ○ and leaves by squares like ▫. The empty tree is also denoted by ▫. Assuming L and R are complete rooted planar binary trees, let $L \wedge R$ be the (unique) complete rooted planar binary tree which has L as left subtree and R as right subtree. Let also $\mathcal{T}_n$ be the set of complete rooted planar binary trees with n nodes and $\mathcal{T}$ be the set of all complete rooted planar binary trees. We use in the sequel the standard terminology (ie. *child*, *ancestor*, *edge*, *path*, ...) about complete rooted planar binary trees [3].

Recall that the nodes of a complete rooted planar binary tree T can be visited in the infix order: it consists in visiting recursively the left subtree of T, then the root, and finally the right subtree. We say that a node y is *on the right compared to* a node x in T if the node x appears strictly before the node y in the infix order and we denote that by $x \rightsquigarrow_T y$. We extend this notation to subtrees saying that a subtree S of root y of T is on the right compared to a node x in T if for all nodes y' of S we have $x \rightsquigarrow_T y'$. We say that a node x of T is the *leftmost node* of T if x is the first visited node in the infix order.

If T is a complete rooted planar binary tree, we shall denote by $\mathrm{ht}(T)$ the *height* of T, that is the length of the longest path connecting the root of T to one of its leaves. For example, we have $\mathrm{ht}\,(\text{▫}) = 0$, $\mathrm{ht}\,(\overset{\wedge}{}) = 1$, and $\mathrm{ht}\left(\overset{\wedge}{}\right) = 2$.

In the sequel, we shall mainly talk about complete rooted planar binary trees so we shall call them simply *trees*.

2.2   Balanced trees

Let us define, for each tree T, the mapping γ_T called the *imbalance mapping* which associates an element of $\mathbb{Z}$ with a node x of T, namely the *imbalance value* of x. It is defined for a node x by:

$$\gamma_T(x) = \mathrm{ht}(R) - \mathrm{ht}(L) \tag{2.1}$$

where L (resp. R) is the left (resp. right) subtree of x.

Balanced trees form a subset of $\mathcal{T}$ composed of trees which have the property of being balanced:

Definition 2.1 *A tree T is* balanced *if for all node x of T, we have*

$$\gamma_T(x) \in \{-1, 0, 1\}. \tag{2.2}$$

Let us denote by $\mathcal{B}_n$ the set of balanced trees with n nodes (see Figure 1 for the first sets) and $\mathcal{B}$ the set of all balanced trees.
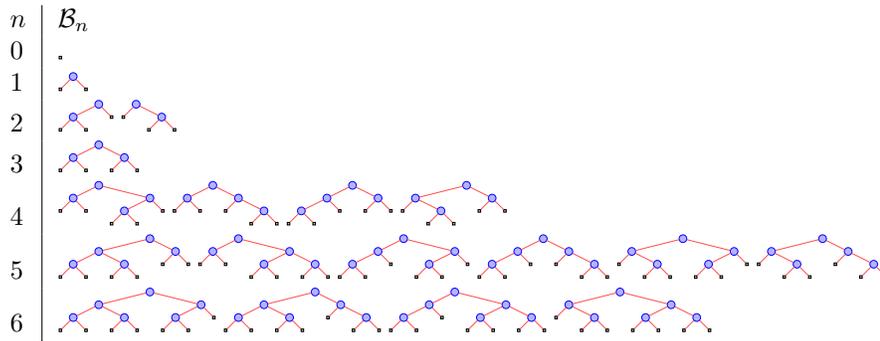

Fig. 1: The first balanced trees.

2.3 The Tamari lattice

The Tamari lattice can be defined in several ways [10, 5] depending on which kind of catalan object (ie. in bijection with trees) the order relation is defined. We give here the most convenient definition for our use. First, let us recall the right rotation operation:

Definition 2.2 *Let T_0 be a tree and $S_0 = (A \wedge B) \wedge C$ be the subtree of root y of T_0. If T_1 is the tree obtained by replacing the tree S_0 by the tree $A \wedge (B \wedge C)$ in T_0 (see Figure 2), we say that T_1 is obtained from T_0 by a* right rotation *of* root y.
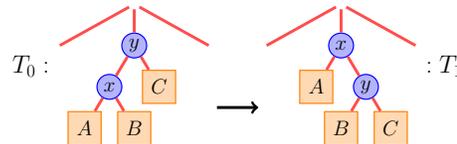

Fig. 2: The right rotation of root y.

We write $T_0 \lessdot T_1$ if T_1 can be obtained by a right rotation from T_0. We call the relation $\lessdot$ the *partial Tamari relation*.

Remark 2.3 *Applying a right rotation to a tree does not change the infix order of its nodes.*

In the sequel, we only talk about right rotations, so we call these simply *rotations*. We are now in a position to give our definition of the Tamari relation:

Definition 2.4 *The* Tamari relation*, written $\preccurlyeq$, is the reflexive and transitive closure of the partial Tamari relation $\lessdot$.*

The Tamari relation is an order relation. For $n \geq 0$, the set $\mathcal{T}_n$ with the $\preccurlyeq$ order relation defines a lattice: the Tamari lattice. We denote by $\mathbb{T}_n = (\mathcal{T}_n, \preccurlyeq)$ the Tamari lattice of order n.
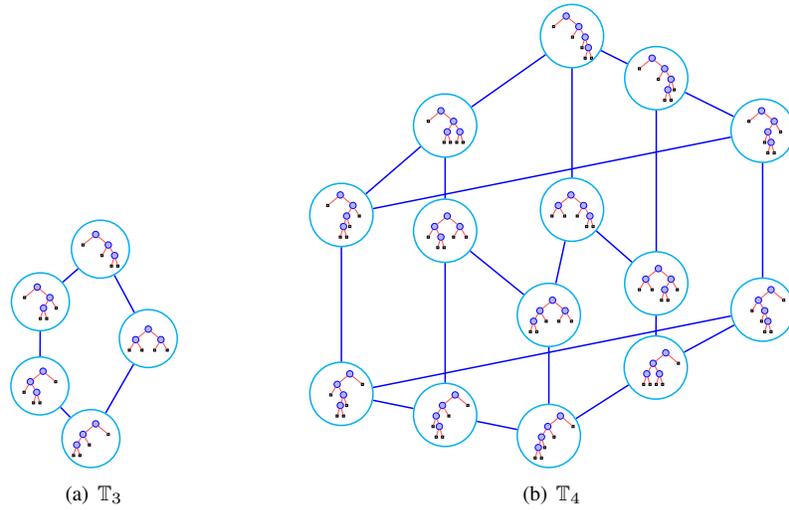
(a) $\mathbb{T}_3$ (b) $\mathbb{T}_4$

Fig. 3: The Tamari lattices $\mathbb{T}_3$ and $\mathbb{T}_4$.

3 Closure by interval of the set of balanced trees

3.1 Rotations and balance

Let us first consider the modifications of the imbalance values of the nodes of a tree $T_0 = (A \wedge B) \wedge C$ when a rotation at its root is applied. Let T_1 be the tree obtained by this rotation, y the root of T_0 and x the left child of y in T_0. Note first that the imbalance values of the nodes of the trees A, B and C are not modified by the rotation. Indeed, only the imbalance values of the nodes x and y are changed. Since T_0 is balanced, we have $\gamma_{T_0}(x) \in \{-1, 0, 1\}$ and $\gamma_{T_0}(y) \in \{-1, 0, 1\}$. Thus, the pair $(\gamma_{T_0}(x), \gamma_{T_0}(y))$ can take nine different values. Here follows the list of the imbalance values of the nodes x and y in the trees T_0 and T_1:

	(B1)	(U1)	(U2)	(B2)	(U3)	(U4)	(U5)	(U6)	(U7)
$(\gamma_{T_0}(x), \gamma_{T_0}(y))$	**(-1, -1)**	(-1, 0)	(-1, 1)	**(0, -1)**	(0, 0)	(0, 1)	(1, -1)	(1, 0)	(1, 1)
$(\gamma_{T_1}(x), \gamma_{T_1}(y))$	**(1, 1)**	(2, 2)	(3, 3)	**(1, 0)**	(2, 1)	(3, 2)	(2, 0)	(3, 1)	(4, 2)

Tab. 1: Imbalance values of the nodes x and y in T_0 and T_1.

Notice that only in (B1) and (B2) the tree T_1 is balanced. We have the following lemma:

Lemma 3.1 *Let T_0 and T_1 be two balanced trees such that $T_0 \lessdot T_1$. Then, the trees T_0 and T_1 have the same height.*

Proof: Since T_0 and T_1 are both balanced, the rotation modifies a subtree S_0 of T_0 such that the imbalance values of the root of S_0, namely y, and the left child of y, namely x, satisfy (B1) or (B2). Let S_1 be the tree obtained by the rotation of root y from S_0. Computing the height of the trees S_0 and S_1, we have $\mathrm{ht}(S_0) = \mathrm{ht}(S_1)$. Thus, as a rotation modifies a tree locally, we have $\mathrm{ht}(T_0) = \mathrm{ht}(T_1)$. $\square$

A rotation transforming a tree T_0 into a tree T_1 is a *conservative balancing rotation* if both T_0 and T_1 are balanced. Considering y the root of this rotation and x the left child of y, we see, by the previous

computations and Lemma 3.1, that T_0 and T_1 are both balanced if and only if T_0 is balanced and

$$(\gamma_{T_0}(x), \gamma_{T_0}(y)) \in \{(-1, -1), (0, -1)\}. \tag{3.1}$$

Similarly, a rotation is an *unbalancing rotation* if T_0 is balanced but T_1 not.

Lemma 3.2 *Let T_0 be a balanced tree and T_1 be an unbalanced tree such that $T_0 \curlywedge T_1$. Then, there exists a node z in T_1 such that $\gamma_{T_1}(z) \geq 2$ and the left subtree and the right subtree of z are both balanced.*

Proof: Immediate, looking at (U1), (U2), (U3), (U4), (U5), (U6) and (U7). □

3.2 Admissible words

Definition 3.3 *A word $z \in \mathbb{N}^*$ is* admissible *if either $|z| \leq 1$ or we have $z_1 - 1 \leq z_2$, and the word obtained by applying the substitution*

$$z_1.z_2 \longrightarrow \begin{cases} \max\{z_1, z_2\} + 1 & \text{if } z_1 - 1 \leq z_2 \leq z_1 + 1, \\ z_2 & \text{otherwise} \end{cases} \tag{3.2}$$

to z is admissible. Let us denote by $\mathcal{A}$ the set of admissible words.

For example, we can check that the word $z = 00122$ is admissible. Indeed, applying the substitution (3.2), we have $00122 \to 1122 \to 222 \to 32 \to 4$ and at each step, the condition $z_1 - 1 \leq z_2$ holds. The word $z' = 1234488$ is also admissible: $1234488 \to 334488 \to 44488 \to 5488 \to 688 \to 88 \to 9$. The word $z'' = 3444$ is not admissible because we have $3444 \to 544 \to 64$ and since that $6 - 1 \not\leq 4$, we have $z'' \notin \mathcal{A}$.

Remark 3.4 *If z is an admissible word, then, for all $1 \leq i \leq |z| - 1$ the inequality $z_i - 1 \leq z_{i+1}$ holds.*

Remark 3.5 *The prefixes and suffixes of an admissible word are still admissible.*

Remark 3.6 *If $z = u.v$ where $z, u, v \in \mathbb{N}^*$ are admissible words, after applying the substitution (3.2) to v to obtain the word v', the word $z' = u.v'$ is still admissible.*

Let the *potential* $\mathrm{P}(z)$ of an admissible word z be the outcome of the application of the substitution (3.2). In the previous examples, we have $\mathrm{P}(z) = 4$ and $\mathrm{P}(z') = 9$.

Let T be a tree, x be a node of T, $(x = x_1, x_2, \ldots, x_\ell)$ be the sequence of all ancestors of x whose right sons are not themselves ancestors of x, ordered from bottom to top and $(S_{x_i})_{1 \leq i \leq \ell}$ be the sequence of the right subtrees of the nodes x_i (see Figure 4). The word z on the alphabet $\mathbb{N}$ defined by $z_i = \mathrm{ht}(S_{x_i})$ is called the *characteristic word* of the node x in the tree T and denoted by $\mathrm{c}_T(x)$.

Lemma 3.7 *Let T be a balanced tree, x a node of T, and z the characteristic word of x. Then, z is admissible and $\mathrm{P}(z) \leq \mathrm{ht}(T)$.*

Proof: By structural induction on balanced trees. The lemma is obviously true for the trees of the set $\mathcal{B}_0 \cup \mathcal{B}_1$. Let L and R be two balanced trees such that $T = L \wedge R$ is balanced too and assume that the lemma is true for both L and R. Let x be a node of T. Distinguishing the cases where x is a node of L, a node of R, or the root of T, we have, by induction, the statement of the lemma. □

Lemma 3.8 *Let T be a tree and y a node of T such that $\mathrm{c}_T(y)$ is admissible and all subtrees of the sequence $(S_{y_i})_{1 \leq i \leq \ell}$ are balanced. Then, for all node x of T such that $y \rightsquigarrow_T x$, the word $\mathrm{c}_T(x)$ is admissible.*
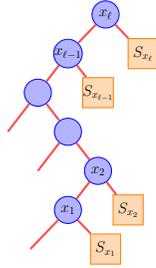
Fig. 4: The sequence $(S_{x_i})_{1 \leq i \leq \ell}$ associated to the node $x = x_1$.

Proof: If x is an ancestor of y, the word $c_T(x)$ is a suffix of $c_T(y)$, thus we have, by Remark 3.5, $c_T(x) \in \mathcal{A}$. Otherwise, let S be the subtree of T such that x is a node of S and the parent of S in T is an ancestor of y. We have $c_T(y) = u.\,\mathrm{ht}(S).v$ where $u, v \in \mathcal{A}$. As $y \rightsquigarrow_T S$, we have $S \in \mathcal{B}$ and by Lemma 3.7, we have $c_S(x) \in \mathcal{A}$ and $\mathrm{P}(c_S(x)) \leq \mathrm{ht}(S)$. Thus, thanks to Remark 3.5, $\mathrm{ht}(S).v \in \mathcal{A}$, so that $c_T(x) = c_S(x).v \in \mathcal{A}$. □

3.3 The main result

Theorem 3.9 *Let T_0 and T_1 be two balanced trees such that $T_0 \preccurlyeq T_1$. Then, the interval $[T_0, T_1]$ only contains balanced trees. In other words, all successors of a tree obtained doing an unbalancing rotation into a balanced tree are unbalanced.*

Proof: To prove the theorem, we shall show that for all balanced tree T_0 and an unbalanced tree T_1 such that $T_0 \lessdot T_1$, all trees T_2 such that $T_1 \preccurlyeq T_2$ are unbalanced. Indeed, T_1 has a property guaranteeing it is unbalanced that can be kept for all its successors.

Let $\mathrm{Imb}_T(x)$ be the property: the node x of T and the node y which is the leftmost node of the left subtree of x satisfy: (see Figure 5):

(1) $\gamma_T(x) \geq 2$;

(2) the left subtree of x is balanced;

(3) all the subtrees S such that $y \rightsquigarrow_T S$ are balanced;

(4) $c_T(y) \in \mathcal{A}$.

Point (2) guarantees that each tree having the previous property is unbalanced.

First, let us show that there exists a node x such that $\mathrm{Imb}_{T_1}(x)$ is true. The tree T_1 is obtained by an unbalancing rotation from T_0. By Lemma 3.2, there exists a node x in T_1 satisfying points (1) and (2). As the left and right subtrees of x are balanced and as all the trees on the right compared to x are balanced in T_0, they remain balanced in T_1, so that point (3) checks out. To establish (4), denoting by y the leftmost node of the left subtree of x in T_1, we have, by Remark 3.6 and Lemmas 3.7 and 3.8, $c_{T_1}(y) \in \mathcal{A}$.

Now, let us show that given a tree T_1 such that $\mathrm{Imb}_{T_1}(x)$ is satisfied for a node x of T_1, for all tree T_2 such that $T_1 \lessdot T_2$, there exists a node x' of T_2 such that $\mathrm{Imb}_{T_2}(x')$ is satisfied. Let y be the leftmost node of the left subtree of x in T_1 and r be the root of the rotation that transforms T_1 into T_2. We will treat all cases depending on the position of r compared to y.

If the node r belongs to a subtree of T_1 which is on the left compared to y, the rotation does not modify any of the subtrees on the right compared to y. Thus we have $\mathrm{Imb}_{T_2}(x)$.
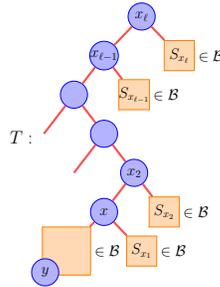
Fig. 5: The imbalance property $\mathrm{Imb}_T(x)$. The node y is the leftmost node of the left subtree of the node x.

If the subtree S_1 of root r satisfies $y \rightsquigarrow_{T_1} S_1$, let S_2 be the subtree of T_2 obtained by the rotation of S_1 which transforms T_1 into T_2. If S_2 is balanced, by Lemma 3.1, $\mathrm{ht}(S_1) = \mathrm{ht}(S_2)$ and we have $\mathrm{Imb}_{T_2}(x)$. If S_2 is not balanced, by the study of the initial case, we have $\mathrm{Imb}_{S_2}(x')$ for a node x' of S_2. Besides, by Remark 3.6 and Lemma 3.8, denoting by y' the leftmost node of the left subtree of x' in T_2, we have $\mathrm{c}_{T_2}(y') \in \mathcal{A}$ and thus, $\mathrm{Imb}_{T_2}(x')$.

If the node r is an ancestor of y and the left child of r is still an ancestor of y, let B be the right subtree of r and A the right subtree of the left child of r in T_1. The rotation replaces the trees A and B by the tree $A \wedge B$. As $\mathrm{c}_{T_1}(y) \in \mathcal{A}$, we have, by Remark 3.4, $\mathrm{ht}(A) - 1 \leq \mathrm{ht}(B)$. Thus, if $A \wedge B$ is balanced, we have $\mathrm{Imb}_{T_2}(x)$. Indeed, points (1), (2) and (3) are clearly satisfied and, by Remark 3.6, we have (4). If $A \wedge B$ is unbalanced, calling x' the root of this tree in T_2, we have $\gamma_{T_2}(x') \geq 2$, and, calling y' the leftmost node of A, we have, by Lemma 3.8, $\mathrm{c}_{T_2}(y') \in \mathcal{A}$. Thus we have $\mathrm{Imb}_{T_2}(x')$.

If the node r is an ancestor of y and the right child of r is still an ancestor of y, the rotation does not modify any of the subtrees on the right compared to y. Thus, we have $\mathrm{Imb}_{T_2}(x)$. $\qquad\square$

4 Tree patterns and synchronous grammars

Word patterns are usually used to describe languages by considering the set of words avoiding them. We use the same idea to describe sets of trees. We show first that we can describe two interesting subsets of the set of balanced trees only by two-nodes patterns.

Next, we follow the methods of [7, 4] to characterize, in our setting, a way to obtain a functional equation admitting as fixed point the generating series enumerating balanced trees. In this purpose, we introduce *synchronous grammars*, allowing to generate trees iteratively. This method gives us a way to enumerate trees avoiding a set of tree patterns because, as we shall see, functional equations of generating series can be extracted from synchronous grammars.

4.1 Tree patterns

Definition 4.1 *A* tree pattern *is a nonempty non complete rooted planar binary tree with labels in $\mathbb{Z}$.*

Let T be a tree and T_γ be the labeled tree of shape T where each node of T_γ is labeled by its imbalance value. The tree T admits an *occurrence* of a tree pattern p if a connected component of T_γ has the same shape and same labels as p.

Now, given a set P of tree patterns, we can define the set composed of the trees that do not admit any occurrence of the elements of P. For example, the set

$$\{\,\textcircled{\small$i$} \mid i \notin \{-1, 0, 1\}\,\} \tag{4.1}$$

describes the set of balanced trees; the set

$$\{ \, \textcircled{i} \mid i \neq 0 \} \tag{4.2}$$

describes the set of perfect trees and

$$\left\{ \, \begin{array}{c} \textcircled{j} \\ \textcircled{i} \end{array} \mid i, j \in \mathbb{Z} \right\} \tag{4.3}$$

describes the set of right comb trees.

4.2 Two particular subsets of balanced trees

Let us describe a subset of the balanced trees and its counterpart such that its elements are, roughly speaking, at the end of the balanced trees subset in the Tamari lattice:

Definition 4.2 *A balanced tree T_0 (resp. T_1) is* maximal *(resp.* minimal*) if for all balanced tree T_1 (resp. T_0) such that $T_0 \curlywedge T_1$ we have T_1 (resp. T_0) unbalanced.*

Proposition 4.3 *A balanced tree T is maximal if and only if it avoids the set of tree patterns*

$$P_{\max} := \left\{ \, \begin{array}{c} \textcircled{-1} \\ \textcircled{-1} \end{array}, \begin{array}{c} \textcircled{-1} \\ \textcircled{0} \end{array} \right\}. \tag{4.4}$$

Similarly, a balanced tree T is minimal if and only if it avoids the set of tree patterns

$$P_{\min} := \left\{ \, \begin{array}{c} \textcircled{1} \\ \textcircled{1} \end{array}, \begin{array}{c} \textcircled{1} \\ \textcircled{0} \end{array} \right\}. \tag{4.5}$$

Proof: Assume that T is maximal. For all tree T_1 such that $T \curlywedge T_1$ we have T_1 unbalanced. Thus, it is impossible to do a conservative balancing rotation from T and it avoids the set $P_{\max}$.

Assume that T avoids the two tree patterns of $P_{\max}$, then, for every tree T_1 such that $T \curlywedge T_1$, the tree T_1 is unbalanced because we can do only unbalancing rotations in T. Thus, the tree T is maximal.

The proof of the second part of the proposition is done in an analogous way. □

4.3 Synchronous grammars and enumeration of balanced trees

Let us first describe a way to obtain the functional equation admitting as fixed point the generating series which enumerates balanced trees [7, 4].

The idea is to generate trees by allowing them to grow from the root to the leaves step by step. For that, we generate *bud trees*, that are non complete rooted planar binary trees with the particularity that the set of external nodes (the nodes without descendant) are *buds*. A bud tree grows by *simultaneously* substituting all of its buds by new bud trees. Trees are finally obtained replacing buds by leaves. The rules of substitution allowing to generate bud trees form a *synchronous grammar*. The link between tree patterns and synchronous grammars is that synchronous grammars generate trees controlling the imbalance value of the nodes. The rules generating balanced trees are

$$\textcircled{x} \quad \longrightarrow \quad \begin{array}{c} \textcircled{-1} \\ \textcircled{x} \quad \textcircled{y} \end{array} + \begin{array}{c} \textcircled{0} \\ \textcircled{x} \quad \textcircled{x} \end{array} + \begin{array}{c} \textcircled{1} \\ \textcircled{y} \quad \textcircled{x} \end{array} \tag{4.6}$$

$$\textcircled{y} \quad \longrightarrow \quad \textcircled{x} \tag{4.7}$$

The role of the bud $\textcircled{x}$ is to generate a node which has -1, 0 or 1 as imbalance value, the only values that a balanced tree can have. The role of the bud $\textcircled{y}$ is to delay the growth of the bud tree to enable the creation of the imbalance values -1 and 1. We have the following theorem:

Theorem 4.4 *Let B be a bud tree generated from the bud x by the previous synchronous grammar. If B does not contain any bud y, replacing all buds x by leaves, we obtain a tree T where each node z of T is labeled by $\gamma_T(z)$. In this way, the previous synchronous grammar generates exactly the set of balanced trees.*

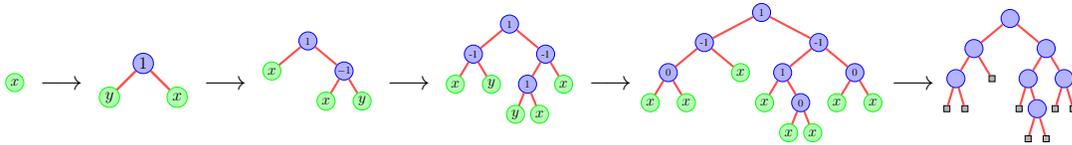Figure 6 shows an example of such a generation.

Fig. 6: Generation of a balanced tree.

The main purpose of synchronous grammars is to obtain a way to enumerate the trees generated. We can translate the set of rules to obtain a functional equation of the generating series enumerating them. For balanced trees, we have [7, 4, 9]:

Theorem 4.5 *The generating series enumerating balanced trees according to the number of leaves of trees is $G_{\mathrm{bal}}(x) := A(x, 0)$ where*

$$A(x, y) := x + A(x^2 + 2xy, x). \tag{4.8}$$

The resolution, or, in other words, the coefficient extraction for this kind of functional equation, is made by iteration. We proceed by computing the sequence of polynomials $(A_i)_{i \geq 0}$ defined by:

$$A_i(x, y) = \begin{cases} x & \text{if } i = 0, \\ x + A_{i-1}(x^2 + 2xy, x) & \text{otherwise.} \end{cases} \tag{4.9}$$

The first iterations give

$$A_0 = x, \tag{4.10}$$
$$A_1 = x + 2xy + x^2, \tag{4.11}$$
$$A_2 = x + 2xy + x^2 + 4x^2y + 2x^3 + 4x^2y^2 + 4x^3y + x^4. \tag{4.12}$$

The fixed point of the sequence $(A_i)_{i \geq 0}$, after substituting 0 to the parameter y in order to ignore bud trees with some buds y, is the generating series of balanced trees counted according to the number of leaves.

We can refine this idea to enumerate maximal balanced trees:

Proposition 4.6 *The generating series enumerating maximal balanced trees according to the number of leaves of the trees is $G_{\mathrm{max}}(x) := A(x, 0, 0)$ where*

$$A(x, y, z) := x + A(x^2 + xy + yz, x, xy). \tag{4.13}$$

Proof: To obtain this functional equation, let us use the following synchronous grammar which generates

maximal balanced trees:

$$(4.14)$$

$$(4.15)$$

$$(4.16)$$

This grammar must generate only maximal balanced trees. By Proposition 4.3, the generated trees must avoid the two tree patterns of $P_{\max}$. To do that, we have to control the growth of the bud x when it generates a tree S such that its root has an imbalance value of -1. Indeed, if the root of the left subtree of S grows with an imbalance value of -1 or 0, one of the two tree patterns is not avoided. The idea is to force the imbalance value of the root of left subtree of S to be 1, role played by the bud z. □

The solution of this functional equation give us the following first values for the number of maximal trees in the Tamari lattice: 1, 1, 1, 1, 2, 2, 2, 4, 6, 9, 11, 13, 22, 38, 60, 89, 128, 183, 256, 353, 512, 805, 1336, 2221, 3594, 5665, 8774, 13433, 20359.

5   The shape of the balanced tree intervals

5.1   *Isomorphism between balanced tree intervals and hypercubes*

A hypercube of dimension k can be seen as a poset whose elements are subsets of a set $\{e_1, \ldots, e_k\}$ ordered by the relation of inclusion. Let us denote by $\mathbb{H}_k$ the hypercube poset of dimension k.

We have the following characterization of the shape of balanced tree intervals:

Theorem 5.1 *Let T_0 and T_1 be two balanced trees such that $T_0 \preccurlyeq T_1$. Then there exists $k \geq 0$ such that the posets $([T_0, T_1], \preccurlyeq)$ and $\mathbb{H}_k$ are isomorphic.*

Proof: First, note by Theorem 3.9, that $I = [T_0, T_1] \subseteq \mathcal{B}$. Thus, every covering relation of the interval I is a conservative balancing rotation.

Then, note that the rotations needed to transform T_0 into T_1 are disjoint in the sense that if y is a node of $T_2 \in I$ and x its left child, if we apply a conservative balancing rotation of root y in T_2 to obtain $T_3 \in I$, all the rotations in the successors of T_3 of root y and of root x are unbalancing rotations. Indeed, by Lemma 3.1, each conservative balancing rotation modifies only the imbalance values of the root of the rotation and its left child, and, according to the values obtained, these two nodes cannot thereafter be roots of conservative balancing rotations.

Besides, by the nature of the conservative balancing rotations and by Theorem 3.9, we can see that all the ways to transform T_0 into T_1 solicit the same rotations, possibly in a different order.

Now, we can associate to a tree $T \in I$ a subset of $\mathbb{N}$ containing the positions in the infix order of the nodes y such that, to obtain T from T_0, we have done, among other, a rotation of root y. The interval I is isomorphic to the poset $\mathbb{H}_k$ where k is the number of rotations needed to transform T_0 into T_1. □

5.2   *Enumeration of balanced tree intervals*

Let us make use again of the synchronous grammars:

Proposition 5.2 *The generating series enumerating balanced tree intervals in the Tamari lattice according to the number of leaves of the trees is $G_{\text{inter}}(x) := A(x, 0, 0)$ where*

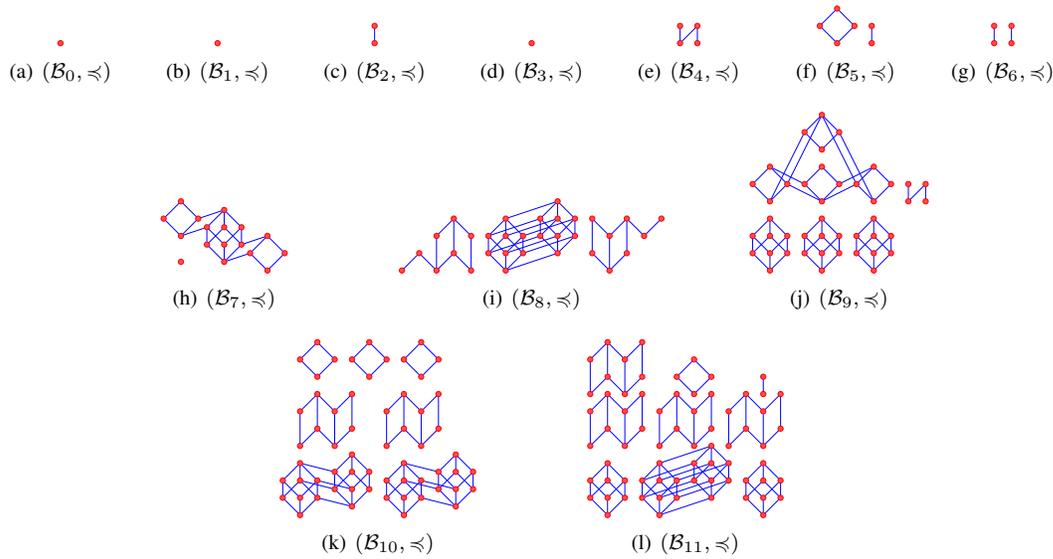$$A(x, y, z) := x + A(x^2 + 2xy + z, x, x^3 + x^2 y). \tag{5.1}$$

Fig. 7: Hasse diagrams of the first $(\mathcal{B}_n, \preccurlyeq)$ posets.

Proof: Let $I = [T_0, T_1]$ be a balanced tree interval. This interval can be encoded by the tree T_0 in which we mark the nodes which are roots of the conservative balancing rotations needed to transform T_0 into T_1. If a node y of T_0 is marked, then its left child cannot be marked too because the rotations of the interval I are disjoint (see the proof of Theorem 5.1). To generate these objects, we use the following synchronous grammar that generates marked trees (the marked nodes are represented by a rectangle instead of a circle):

$$(5.2)$$

$$(5.3)$$

$$(5.4)$$

$$\square$$

The solution of this functional equation gives us the following first values for the number of balanced tree intervals in the Tamari lattice: 1, 1, 3, 1, 7, 12, 6, 52, 119, 137, 195, 231, 1019, 3503, 6593, 12616, 26178, 43500, 64157, 94688, 232560, 817757, 2233757, 5179734.

The interval $[T_0, T_1]$ is a *maximal balanced tree interval* if T_0 (resp. T_1) is a minimal (resp. maximal) balanced tree.

Proposition 5.3 *The generating series enumerating maximal balanced tree intervals in the Tamari lattice according to the number of leaves of the trees is $G_{\text{intermax}}(x) := A(x, 0, 0, 0)$ where*

$$A(x, y, z, t) := x + A(x^2 + 2yz + t, x, yz + t, x^3 + x^2 y). \qquad (5.5)$$

Proof: Let $I = [T_0, T_1]$ be a maximal balanced tree interval. This interval can be encoded by the minimal tree T_0 in which we mark the nodes which are roots of the conservative balancing rotations needed to

transform T_0 into T_1. Since T_1 is a maximal balanced tree, by Proposition 4.3, it avoids the tree patterns of $P_{\max}$, thus, the object which encodes I must not have a node which is root of a conservative balancing rotation not marked if its parent or its left child is not marked. To generate these objects, we use the following synchronous grammar:

$$x \longrightarrow \quad \vcenter{\hbox{tree with 0: x, x}} + \vcenter{\hbox{tree with 1: y, z_1}} + \vcenter{\hbox{tree with -1: z_2, y}} + t \tag{5.6}$$

$$y \longrightarrow x \tag{5.7}$$

$$z_1 \longrightarrow \vcenter{\hbox{tree with -1: z_2, y}} + t \tag{5.8}$$

$$z_2 \longrightarrow \vcenter{\hbox{tree with 1: y, z_1}} + t \tag{5.9}$$

$$t \longrightarrow \vcenter{\hbox{tree with -1, 0: x, x, x}} + \vcenter{\hbox{tree with -1, -1: x, y, x}} \tag{5.10}$$

Note that the buds z_1 and z_2 play the same role so that the functional equation is simplified.   □

   The solution of this functional equation gives us the following first values for the number of maximal balanced tree intervals in the Tamari lattice: $1, 1, 1, 1, 3, 2, 2, 6, 9, 15, 15, 17, 41, 77, 125, 178, 252, 376,$ $531, 740, 1192, 2179, 4273, 7738, 13012, 20776, 32389, 49841, 75457, 113011$.

References

[1] Sage mathematics software, version 4.2, 2009. `http://www.sagemath.org/`.

[2] G.M. Adelson-Velsky and E. M. Landis. An algorithm for the organization of information. *Soviet Mathematics Doklady, 3:1259-1263*, 1962.

[3] A. Aho and J. Ullman. *Foundations of Computer Science*. W. H. Freeman, 1994.

[4] F. Bergeron, G. Labelle, and P. Leroux. *Combinatorial Species and Tree-like Structures*. Cambridge University Press, 1994.

[5] O. Bernardi and N. Bonichon. Catalan's intervals and realizers of triangulations. *FPSAC*, 2007.

[6] T.H. Cormen, C. E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to algorithms*. McGraw-Hill, 2003.

[7] D. Knuth. *The Art of Computer Programming. Volume 3. Sorting and searching*. Addison Wesley Longman, 1998.

[8] D. Knuth. *The Art of Computer Programming. Volume 4, Fascicle 4. Generating all trees - History of combinatorial generation*. Addison Wesley Longman, 2004.

[9] N. J. A. Sloane. The on-line encyclopedia of integer sequences. `http://www.research.att.com/~njas/sequences/`.

[10] R. P. Stanley. *Enumerative Combinatorics II*. Cambridge University Press, 1990.