ALGEBRAIC AND COMBINATORIAL STRUCTURES ON PAIRS OF TWIN BINARY TREES

SAMUELE GIRAUDO

Abstract. We give a new construction of a Hopf algebra defined first by Reading [Rea05] whose bases are indexed by objects belonging to the Baxter combinatorial family (*i.e.*, Baxter permutations, pairs of twin binary trees, *etc.*). Our construction relies on the definition of the Baxter monoid, analog of the plactic monoid and the sylvester monoid, and on a Robinson-Schensted-like correspondence and insertion algorithm. Indeed, the Baxter monoid leads to the definition of a lattice structure over pairs of twin binary trees and the definition of a Hopf algebra. The algebraic properties of this Hopf algebra are studied and among other, multiplicative bases are provided, and freeness and self-duality proved.

Contents

Date: April 20, 2012.
Key words and phrases. Hopf algebra; Robinson-Schensted algorithm; Quotient monoid; Lattice; Baxter permutation.

1. Introduction

In recent years, many combinatorial Hopf algebras, whose bases are indexed by combinatorial objects, have been intensively studied. For example, the Malvenuto-Reutenauer Hopf algebra **FQSym** of Free quasi-symmetric functions [MR95, DHT02] has bases indexed by permutations. This Hopf algebra admits several Hopf subalgebras: The Hopf algebra of Free symmetric functions **FSym** [PR95, DHT02], whose bases are indexed by standard Young tableaux, the Hopf algebra **Bell** [Rey07] whose bases are indexed by set partitions, the Loday-Ronco Hopf algebra **PBT** [LR98, HNT05] whose bases are indexed by planar binary trees, and the Hopf algebra **Sym** of non-commutative symmetric functions [GKL+94] whose bases are indexed by integer compositions. A unifying approach to construct all these structures relies on a definition of a congruence on words leading to the definition of monoids on combinatorial objects. Indeed, **FSym** is directly obtained from the plactic monoid [LS81, DHT02, Lot02], **Bell** from the Bell monoid [Rey07], **PBT** from the sylvester monoid [HNT02, HNT05], and **Sym** from the hypoplactic monoid [KT97, Nov98]. The richness of these constructions relies on the fact that, in addition to constructing Hopf algebras, the definition of such monoids often brings partial orders, combinatorial algorithms and Robinson-Schensted-like algorithms, of independent interest.

The Baxter combinatorial family admits various representations. The most famous of these are Baxter permutations [Bax64], which are permutations that avoid certain patterns, and pairs of twin binary trees [DG94]. This family also contains more exotic objects like quadrangulations [ABP04] and plane bipolar orientations [BBMF08]. In this paper, we propose to enrich the above collection of Hopf algebras by providing a plactic-like monoid, namely the Baxter monoid, leading to the construction of a Hopf algebra whose bases are indexed by objects belonging to this combinatorial family.

In order to show examples of relations between lattice congruences [CS98] and Hopf algebras, Reading presented in [Rea05] a lattice congruence of the permutohedron whose equivalence classes are indexed by twisted Baxter permutations. These permutations were defined by a pattern avoidance property. This congruence is very natural: The meet of two lattice congruences of the permutohedron related to the construction of **PBT** is one starting point to build **Sym**; A natural question is to understand what happens when the join, instead of the meet, of these two lattice congruences is considered. Reading proved that his lattice congruence is precisely this last one, and that the minimal elements of its equivalence classes are twisted Baxter permutations. Besides, thanks to his theory, he gets for free a Hopf algebra whose bases are indexed by twisted Baxter permutations. Actually, twisted Baxter permutations are equinumerous with Baxter permutations. Indeed, Law and Reading pointed out in [LR12] that the first proof occurred in unpublished notes of West. Hence, the Hopf algebra of Reading defined in [Rea05] can already be seen as a Hopf algebra on Baxter permutations, and our construction, considered as a different construction of the same Hopf algebra. Moreover, very recently, Law and Reading [LR12] detailed their construction of this Hopf algebra and studied some of its algebraic properties.

We started independently the study of Baxter objects in a different way: We looked for a quotient of the free monoid analogous to the plactic and the sylvester monoid. Surprisingly, the equivalence classes of permutations under our monoid congruence are the same as the equivalence classes of the lattice congruence of Law and Reading, and hence have the same by-products, as *e.g.*, the Hopf algebra structure and the fact that each class contains both one twisted and one non-twisted Baxter permutation. However, even if both points of view lead to the same general theory, their paths are different and provide different ways of understanding

the construction, one centered on lattice theory, the other centered on combinatorics on words. Moreover, a large part of the results of each paper do not appear in the other as, in our case, the Robinson-Schensted-like correspondence and its insertion algorithm, the polynomial realization, the bidendriform bialgebra structure, the freeness, cofreeness, self-duality, primitive elements, and multiplicative bases of the Hopf algebra, and a few other combinatorial properties.

We begin by recalling in Section 2 the preliminary notions about words, permutations, and pairs of twin binary trees used thereafter. In Section 3, we define the Baxter congruence. This congruence allows to define a quotient of the free monoid, the Baxter monoid, which has a number of properties required for the Hopf algebraic construction which follows. We show that the Baxter monoid is intimately linked to the sylvester monoid and that the equivalence classes of the permutations under the Baxter congruence form intervals of the permutohedron. Next, in Section 4, we develop a Robinson-Schensted-like insertion algorithm that allows to decide if two words are equivalent according to the Baxter congruence. Given a word, this algorithm computes iteratively a pair of twin binary trees inserting one by one the letters of u. We give as well some algorithms to read the minimal, the maximal and the Baxter permutation of a Baxter equivalence class encoded by a pair of twin binary trees. We also show that each equivalence class of permutations under the Baxter congruence contains exactly one Baxter permutation. Section 5 is devoted to the study of some properties of the equivalence classes of permutations under the Baxter congruence. This leads to the definition of a lattice structure on pairs of twin binary trees, very similar to the Tamari lattice [Tam62, Knu06] since covering relations can be expressed by binary tree rotations. We introduce in this section *twin Tamari diagrams* that are objects in bijection with pairs of twin binary trees and offer a simple way to test comparisons in this lattice. Finally, in Section 6, we start by recalling some basic facts about the Hopf algebra of Free quasi-symmetric functions **FQSym**, and then give our construction of the Hopf algebra **Baxter** and study it. Using the polynomial realization of **FQSym**, we deduce a polynomial realization of **Baxter**. Using the order structure on pairs of twin binary trees defined in the above section, we describe its product as an interval of this order. Moreover, we prove that this Hopf algebra is free as an algebra by constructing two multiplicative bases, and introduce two operators on pairs of twin binary trees, analogous to the operators *over* and *under* of Loday-Ronco on binary trees [LR02]. Using the results of Foissy on bidendriform bialgebras [Foi07], we show that this Hopf algebra is also self-dual and that the Lie algebra of its primitive elements is free. We conclude by explaining some morphism with other known Hopf subalgebras of **FQSym**.

This paper is an extended version of [Gir11]. It contains all proofs and Sections 4 and 6 have new results.

Acknowledgments. The author would like to thank Florent Hivert and Jean-Christophe Novelli for their advice and help during all stages of the preparation of this paper. The computations of this work have been done with the open-source mathematical software Sage [S+11].

2. Preliminaries

2.1. **Words, definitions and notations.** In the sequel, $A := \{a_1 < a_2 < \cdots\}$ is a totally ordered infinite alphabet and A^* is the free monoid generated by A. Let $u \in A^*$. We shall denote by $|u|$ the length of u and by ϵ the word of length 0. The largest (resp. smallest) letter of u is denoted by $\max(u)$ (resp. $\min(u)$). The *evaluation* $\mathrm{ev}(u)$ of the word u is the non-negative integer vector such that its i-th entry is the number of occurrences of the letter a_i in u. It is convenient to denote by $\mathrm{Alph}(u) := \{u_i : 1 \leqslant i \leqslant |u|\}$ the smallest alphabet on which u is

defined. We say that (i, j) is an *inversion* of u if $i < j$ and $u_i > u_j$. Additionally, i is *descent* of u if $(i, i+1)$ is an inversion of u.

Let us now recall some classical operations on words. We shall denote by $u^\sim := u_{|u|} \dots u_1$ the *mirror image* of u and by $u_{|S}$ the *restriction* of u on the alphabet $S \subseteq A$, that is the longest subword of u such that $\mathrm{Alph}(u) \subseteq S$. Let $v \in A^*$. The *shuffle product* $\sqcup\!\sqcup$ is recursively defined on the linear span of words $\mathbb{Z}\langle A \rangle$ by

$$u \sqcup\!\sqcup v := \begin{cases} u & \text{if } v = \epsilon, \\ v & \text{if } u = \epsilon, \\ \mathtt{a}(u' \sqcup\!\sqcup \mathtt{b}v') + \mathtt{b}(\mathtt{a}u' \sqcup\!\sqcup v') & \text{otherwise, where } u = \mathtt{a}u',\ v = \mathtt{b}v', \text{ and } \mathtt{a}, \mathtt{b} \in A. \end{cases} \tag{2.1}$$

For example,

$$\begin{aligned} \mathbf{a_1 a_2} \sqcup\!\sqcup a_2 a_1 &= \mathbf{a_1 a_2} a_2 a_1 + \mathbf{a_1} a_2 \mathbf{a_2} a_1 + \mathbf{a_1} a_2 a_1 \mathbf{a_2} + a_2 \mathbf{a_1 a_2} a_1 + a_2 \mathbf{a_1} a_1 \mathbf{a_2} + a_2 a_1 \mathbf{a_1 a_2}, \\ &= a_1 a_2 a_1 a_2 + 2\, a_1 a_2 a_2 a_1 + 2\, a_2 a_1 a_1 a_2 + a_2 a_1 a_2 a_1. \end{aligned} \tag{2.2}$$

Let $A^\# := \{a_1^\# > a_2^\# > \cdots\}$ be the alphabet A on which the order relation has been reversed. The *Schützenberger transformation* $\#$ is defined on words by

$$u^\# = \left(u_1 u_2 \dots u_{|u|}\right)^\# := u_{|u|}^\# \dots u_2^\# u_1^\#. \tag{2.3}$$

For example, $(a_5 a_3 a_1 a_1 a_5 a_2)^\# = a_2^\# a_5^\# a_1^\# a_1^\# a_3^\# a_5^\#$. Note that by setting $A^{\#\#} := A$, the transformation $\#$ becomes an involution on words.

2.2. **Permutations, definitions and notations.** Denote by $\mathfrak{S}_n$ the set of permutations of size n and by $\mathfrak{S}$ the set of all permutations. One can see a permutation of size n as a word without repetition of length n on the first letters of A. We shall call i a *recoil* of $\sigma \in \mathfrak{S}_n$ if $(i, i+1)$ is an inversion of σ^{-1}. By convention, n also is a recoil of σ.

The *(right) permutohedron order* is the partial order $\leqslant_{\mathrm{P}}$ defined on $\mathfrak{S}_n$ where σ is covered by ν if $\sigma = u\,\mathtt{ab}\,v$ and $\nu = u\,\mathtt{ba}\,v$ where $\mathtt{a} < \mathtt{b} \in A$, and u and v are words. Recall that one has $\sigma \leqslant_{\mathrm{P}} \nu$ if and only if any inversion of σ^{-1} also is an inversion of ν^{-1}.

Let $\sigma, \nu \in \mathfrak{S}$. The permutation $\sigma \diagup \nu$ is obtained by concatenating σ and the letters of ν incremented by $|\sigma|$; In the same way, the permutation $\sigma \diagdown \nu$ is obtained by concatenating the letters of ν incremented by $|\sigma|$ and σ. For example,

$$\mathbf{312} \diagup 2314 = \mathbf{312}5647 \quad \text{and} \quad \mathbf{312} \diagdown 2314 = 5647\mathbf{312}. \tag{2.4}$$

A permutation σ is *connected* if $\sigma = \nu \diagup \pi$ implies $\nu = \sigma$ or $\pi = \sigma$. Similarly, σ is *anti-connected* if $\sigma^\sim$ is connected. The *shifted shuffle product* $\overline{\sqcup\!\sqcup}$ of two permutations is defined by

$$\sigma \mathbin{\overline{\sqcup\!\sqcup}} \nu := \sigma \sqcup\!\sqcup \left(\nu_1 + |\sigma| \dots \nu_{|\nu|} + |\sigma|\right). \tag{2.5}$$

For example,

$$\mathbf{12} \mathbin{\overline{\sqcup\!\sqcup}} 21 = \mathbf{12} \sqcup\!\sqcup 43 = \mathbf{1243} + \mathbf{1423} + \mathbf{1432} + 4\mathbf{123} + 4\mathbf{132} + 43\mathbf{12}. \tag{2.6}$$

The *standardized word* $\mathrm{std}(u)$ of $u \in A^*$ is the unique permutation of size $|u|$ having the same inversions as u. For example, $\mathrm{std}(a_3 a_1 a_4 a_2 a_5 a_7 a_4 a_2 a_3) = 416289735$.

2.3. **Binary trees, definitions and notations.** We call *binary tree* any complete rooted planar binary tree. Recall that a binary tree T is either a *leaf* (also called *empty tree*) denoted by $\perp$, or a node that is attached through two edges to two binary trees, called respectively the *left subtree* and the *right subtree* of T. Let $\mathcal{BT}_n$ be the set of binary trees with n nodes and $\mathcal{BT}$ be the set of all binary trees. We use in the sequel the standard terminology (*i.e.*, *child*, *ancestor*, *path*, *etc.*) about binary trees [AU94]. In our graphical representations, nodes are represented by circles ○, leaves by squares ■, edges by segments ／ or ＼, and arbitrary subtrees by big squares like ▢.

2.3.1. *The Tamari order.* The *Tamari order* [Tam62, Knu06] is the partial order $\leqslant_{\mathrm{T}}$ defined on $\mathcal{BT}_n$ where $T_0 \in \mathcal{BT}_n$ is covered by $T_1 \in \mathcal{BT}_n$ if it is possible to obtain T_1 by performing a *right rotation* into T_0 (see Figure 1). One has $T_0 \leqslant_{\mathrm{T}} T_1$ if and only if starting from T_0, it is

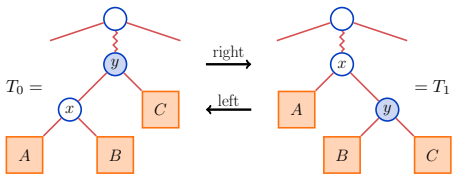


FIGURE 1. The right rotation of root y.

possible to obtain T_1 by performing some right rotations.

2.3.2. *Operations on binary trees.* If L and R are binary trees, denote by $L \wedge R$ the binary tree which has L as left subtree and R as right subtree. Similarly, if L and R are A-labeled binary trees, denote by $L \wedge_{\mathtt{a}} R$ the A-labeled binary tree which has L as left subtree, R as right subtree and a root labeled by $\mathtt{a} \in A$.

Let $T_0, T_1 \in \mathcal{BT}$. The binary tree $T_0 \,／\, T_1$ is obtained by grafting T_0 from its root on the leftmost leaf of T_1; In the same way, the binary tree $T_0 \,＼\, T_1$ is obtained by grafting T_1 from its root on the rightmost leaf of T_0.

For example, for

(2.7) $T_0 :=$  and $T_1 :=$  ,

we have

(2.8) $T_0 \wedge T_1 =$  ,

(2.9) $T_0 \,／\, T_1 =$  and $T_0 \,＼\, T_1 =$  .

2.3.3. *Binary search trees, increasing, and decreasing binary trees.* An A-labeled binary tree T is a *right* (resp. *left*) *binary search tree* if for any node x labeled by $\mathtt{b}$, each label $\mathtt{a}$ of a node in the left subtree of x and each label $\mathtt{c}$ of a node in the right subtree of x, the inequality $\mathtt{a} \leqslant \mathtt{b} < \mathtt{c}$ (resp. $\mathtt{a} < \mathtt{b} \leqslant \mathtt{c}$) holds.

A binary tree $T \in \mathcal{BT}_n$ is an *increasing* (resp. *decreasing*) *binary tree* if it is bijectively labeled on $\{1, \ldots, n\}$ and, for any node x of T, if y is a child of x, then the label of y is greater (resp. smaller) than the label of x.

The *shape* $\mathrm{sh}(T)$ of an A-labeled binary tree T is the unlabeled binary tree obtained by forgetting its labels.

2.3.4. *Inorder traversal.* The *inorder traversal* of a binary tree T consists in recursively visiting its left subtree, then its root, and finally its right subtree (see Figure 2). We shall say that a

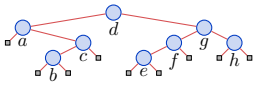


FIGURE 2. The sequence (a, b, c, d, e, f, g, h) is the sequence of all nodes of this binary tree visited by the inorder traversal.

node x is *the i-th node of T* if x is the i-th visited node by the inorder traversal of T. In the same way, a leaf y is *the j-th leaf of T* if y is the j-th visited leaf by the inorder traversal of T. We also say that i is the *index* of x and j is the *index* of y. If T is labeled, its *inorder reading* is the word $u_1 \dots u_{|u|}$ such that for any $1 \leqslant i \leqslant |u|$, u_i is the label of the i-th node of T. Note that when T is a right (or left) binary search tree, its inorder reading is a nondecreasing word.

2.3.5. *The canopy of binary trees.* The *canopy* (see [LR98] and [Vie04]) $\mathrm{cnp}(T)$ of a binary tree T is the word on the alphabet $\{0, 1\}$ obtained by browsing the leaves of T from left to right except the first and the last one, writing 0 if the considered leaf is oriented to the right, 1 otherwise (see Figure 3). Note that the orientation of the leaves in a binary tree is determined only by its nodes so that we can omit to draw the leaves in our graphical representations.

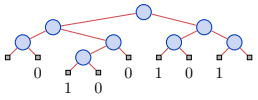


FIGURE 3. The canopy of this binary tree is 0100101.

2.4. **Baxter permutations and pairs of twin binary trees.**

2.4.1. *Baxter permutations.* A permutation σ is a *Baxter permutation* if for any subword $u := u_1u_2u_3u_4$ of σ such that the letters u_2 and u_3 are adjacent in σ, $\mathrm{std}(u) \notin \{2413, 3142\}$. In other words, σ is a Baxter permutation if it avoids the *generalized permutation patterns* $2-41-3$ and $3-14-2$ (see [BS00] for an introduction on generalized permutation patterns). For example, **42173856** is not a Baxter permutation; On the other hand 436975128, is a Baxter permutation. Let us denote by $\mathfrak{S}_n^{\mathrm{B}}$ the set of Baxter permutations of size n and by $\mathfrak{S}^{\mathrm{B}}$ the set of all Baxter permutations.

2.4.2. *Pairs of twin binary trees.* A *pair of twin binary trees* (T_L, T_R) is made of two binary trees $T_L, T_R \in \mathcal{BT}_n$ such that the canopies of T_L and T_R are complementary, that is

$$\mathrm{cnp}(T_L)_i \neq \mathrm{cnp}(T_R)_i \text{ for all } 1 \leqslant i \leqslant n-1 \tag{2.10}$$

(see Figure 4).

Denote by $\mathcal{TBT}_n$ the set of pairs of twin binary trees where each binary tree has n nodes and by $\mathcal{TBT}$ the set of all pairs of twin binary trees.

An A-labeled pair of twin binary trees (T_L, T_R) is a *pair of twin binary search trees* if T_L (resp. T_R) is an A-labeled left (resp. right) binary search tree and T_L and T_R have the same

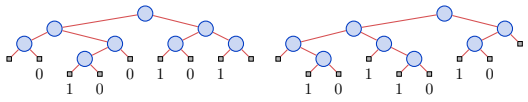


FIGURE 4. A pair of twin binary trees.

inorder reading. The *shape* $\mathrm{sh}(J)$ of an A-labeled pair of twin binary trees $J := (T_L, T_R)$ is the unlabeled pair of twin binary trees $(\mathrm{sh}(T_L), \mathrm{sh}(T_R))$.

In [DG94], Dulucq and Guibert have highlighted a bijection between Baxter permutations and unlabeled pairs of twin binary trees. In the sequel, we shall make use of a very similar bijection.

3. The Baxter monoid

3.1. **Definition and first properties.** Recall that an equivalence relation $\equiv$ defined on A^* is a *congruence* if for all $u, u', v, v' \in A^*$, $u \equiv u'$ and $v \equiv v'$ imply $uv \equiv u'v'$. Note that the quotient $A^*/_{\equiv}$ of A^* by a congruence $\equiv$ is naturally a monoid. Indeed, by denoting by $\tau : A^* \to A^*/_{\equiv}$ the canonical projection, the set $A^*/_{\equiv}$ is endowed with a product $\cdot$ defined by $\widehat{u} \cdot \widehat{v} := \tau(uv)$ for all $\widehat{u}, \widehat{v} \in A^*/_{\equiv}$ where u and v are any words such that $\tau(u) = \widehat{u}$ and $\tau(v) = \widehat{v}$.

Definition 3.1. *The* Baxter monoid *is the quotient of the free monoid A^* by the congruence $\equiv_{\mathrm{B}}$ that is the reflexive and transitive closure of the* Baxter adjacency relations $\leftharpoonup_{\mathrm{B}}$ *and* $\rightharpoonup_{\mathrm{B}}$ *defined for $u, v \in A^*$ and $\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d} \in A$ by*

$$\mathtt{c}\, u\, \mathtt{ad}\, v\, \mathtt{b} \leftharpoonup_{\mathrm{B}} \mathtt{c}\, u\, \mathtt{da}\, v\, \mathtt{b} \qquad \textit{where} \quad \mathtt{a} \leqslant \mathtt{b} < \mathtt{c} \leqslant \mathtt{d}, \tag{3.1}$$

$$\mathtt{b}\, u\, \mathtt{da}\, v\, \mathtt{c} \rightharpoonup_{\mathrm{B}} \mathtt{b}\, u\, \mathtt{ad}\, v\, \mathtt{c} \qquad \textit{where} \quad \mathtt{a} < \mathtt{b} \leqslant \mathtt{c} < \mathtt{d}. \tag{3.2}$$

For example, the $\equiv_{\mathrm{B}}$-equivalence class of 2415253 (see Figure 5) is

$$\{2142553, 2145253, 2145523, 2412553, 2415253, 2415523, 2451253, 2451523, 2455123\}. \tag{3.3}$$

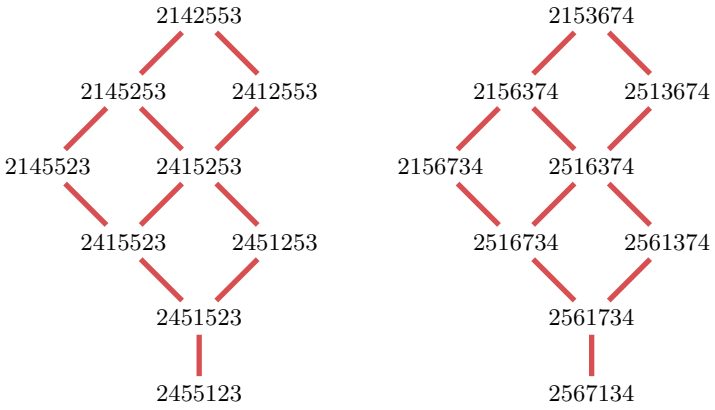


FIGURE 5. The Baxter equivalence class of the word $u := 2415253$ and of the permutation $2516374 = \mathrm{std}(u)$. Edges represent Baxter adjacency relations.

Note that if the Baxter congruence is applied on words without repetition, the two Baxter adjacency relations $\leftharpoonup_{\mathrm{B}}$ and $\rightharpoonup_{\mathrm{B}}$ can be replaced by the only adjacency relation $\rightleftharpoons_{\mathrm{B}}$ defined

for $u, v \in A^*$ and $\mathtt{a}, \mathtt{b}, \mathtt{b}', \mathtt{d} \in A$ by

$$\mathtt{b}\, u\, \mathtt{ad}\, v\, \mathtt{b}' \rightleftarrows_{\mathrm{B}} \mathtt{b}\, u\, \mathtt{da}\, v\, \mathtt{b}' \qquad \text{where} \qquad \mathtt{a} < \mathtt{b}, \mathtt{b}' < \mathtt{d}. \tag{3.4}$$

3.1.1. *Compatibility with the destandardization process.* A monoid $A^*/_{\equiv}$ is *compatible with the destandardization process* if for all $u, v \in A^*$, $u \equiv v$ if and only if $\mathrm{std}(u) \equiv \mathrm{std}(v)$ and $\mathrm{ev}(u) = \mathrm{ev}(v)$.

Proposition 3.2. *The Baxter monoid is compatible with the destandardization process.*

Proof. It is enough to check the property on adjacency relations. Let $u, v \in A^*$. Assume $u \leftharpoonup_{\mathrm{B}} v$. We have

$$u = x\, \mathtt{c}\, y\, \mathtt{ad}\, z\, \mathtt{b}\, t \quad \text{and} \quad v = x\, \mathtt{c}\, y\, \mathtt{da}\, z\, \mathtt{b}\, t \tag{3.5}$$

for some letters $\mathtt{a} \leqslant \mathtt{b} < \mathtt{c} \leqslant \mathtt{d}$ and words x, y, z, and t. Since $\leftharpoonup_{\mathrm{B}}$ acts by permuting letters, we have $\mathrm{ev}(u) = \mathrm{ev}(v)$. Moreover, the letters $\mathtt{a}'$, $\mathtt{b}'$, $\mathtt{c}'$ and $\mathtt{d}'$ of $\mathrm{std}(u)$ respectively at the same positions as the letters $\mathtt{a}$, $\mathtt{b}$, $\mathtt{c}$ and $\mathtt{d}$ of u satisfy $\mathtt{a}' < \mathtt{b}' < \mathtt{c}' < \mathtt{d}'$ due to their relative positions into $\mathrm{std}(u)$ and the order relations between $\mathtt{a}$, $\mathtt{b}$, $\mathtt{c}$ and $\mathtt{d}$. The same relations hold for the letters of $\mathrm{std}(v)$, showing that $\mathrm{std}(u) \leftharpoonup_{\mathrm{B}} \mathrm{std}(v)$. The proof is analogous for the case $u \rightharpoonup_{\mathrm{B}} v$.

Conversely, assume that v is a permutation of u and $\mathrm{std}(u) \leftharpoonup_{\mathrm{B}} \mathrm{std}(v)$. We have

$$\mathrm{std}(u) = x\, \mathtt{c}\, y\, \mathtt{ad}\, z\, \mathtt{b}\, t \quad \text{and} \quad \mathrm{std}(v) = x\, \mathtt{c}\, y\, \mathtt{da}\, z\, \mathtt{b}\, t \tag{3.6}$$

for some letters $\mathtt{a} < \mathtt{b} < \mathtt{c} < \mathtt{d}$ and words x, y, z, and t. The word u is a non-standardized version of $\mathrm{std}(u)$ so that the letters $\mathtt{a}'$, $\mathtt{b}'$, $\mathtt{c}'$ and $\mathtt{d}'$ of u respectively at the same positions as the letters $\mathtt{a}$, $\mathtt{b}$, $\mathtt{c}$ and $\mathtt{d}$ of $\mathrm{std}(u)$ satisfy $\mathtt{a}' \leqslant \mathtt{b}' < \mathtt{c}' \leqslant \mathtt{d}'$ due to their relative positions into u and the order relations between $\mathtt{a}$, $\mathtt{b}$, $\mathtt{c}$ and $\mathtt{d}$. The same relations hold for the letters of v, showing that $u \leftharpoonup_{\mathrm{B}} v$. The proof is analogous for the case $\mathrm{std}(u) \rightharpoonup_{\mathrm{B}} \mathrm{std}(v)$. □

3.1.2. *Compatibility with the restriction of alphabet intervals.* A monoid $A^*/_{\equiv}$ is *compatible with the restriction of alphabet intervals* if for any interval I of A and for all $u, v \in A^*$, $u \equiv v$ implies $u_{|I} \equiv v_{|I}$.

Proposition 3.3. *The Baxter monoid is compatible with the restriction of alphabet intervals.*

Proof. It is enough to check the property on adjacency relations. Moreover, by Proposition 3.2, it is enough to check the property for permutations. Let $\sigma, \nu \in \mathfrak{S}_n$ such that $\sigma \rightleftarrows_{\mathrm{B}} \nu$. We have $\sigma = x\, \mathtt{b}\, y\, \mathtt{ad}\, z\, \mathtt{b}'\, t$ and $\nu = x\, \mathtt{b}\, y\, \mathtt{da}\, z\, \mathtt{b}'\, t$ for some letters $\mathtt{a} < \mathtt{b}, \mathtt{b}' < \mathtt{d}$ and words x, y, z, and t. Let I be an interval of $\{1, \dots, n\}$ and $R := I \cap \{\mathtt{a}, \mathtt{b}, \mathtt{b}', \mathtt{d}\}$. If $R = \{\mathtt{a}, \mathtt{b}, \mathtt{b}', \mathtt{d}\}$,

$$\sigma_{|I} = x_{|I}\, \mathtt{b}\, y_{|I}\, \mathtt{ad}\, z_{|I}\, \mathtt{b}'\, t_{|I} \quad \text{and} \quad \nu_{|I} = x_{|I}\, \mathtt{b}\, y_{|I}\, \mathtt{da}\, z_{|I}\, \mathtt{b}'\, t_{|I} \tag{3.7}$$

so that $\sigma_{|I} \rightleftarrows_{\mathrm{B}} \nu_{|I}$. Otherwise, we have $\sigma_{|I} = \nu_{|I}$ and thus $\sigma_{|I} \equiv_{\mathrm{B}} \nu_{|I}$. □

3.1.3. *Compatibility with the Schützenberger involution.* A monoid $A^*/_{\equiv}$ is *compatible with the Schützenberger involution* if for all $u, v \in A^*$, $u \equiv v$ implies $u^{\#} \equiv v^{\#}$.

Proposition 3.4. *The Baxter monoid is compatible with the Schützenberger involution.*

Proof. It is enough to check the property on adjacency relations. Moreover, by Proposition 3.2, it is enough to check the property for permutations. Let $\sigma, \nu \in \mathfrak{S}_n$ and assume that $\sigma \rightleftarrows_{\mathrm{B}} \nu$. We have $\sigma = x\, \mathtt{b}\, y\, \mathtt{ad}\, z\, \mathtt{b}'\, t$ and $\nu = x\, \mathtt{b}\, y\, \mathtt{da}\, z\, \mathtt{b}'\, t$ for some letters $\mathtt{a} < \mathtt{b}, \mathtt{b}' < \mathtt{d}$ and words x, y, z, and t. We have

$$\sigma^{\#} = t^{\#}\, \mathtt{b}'^{\#}\, z^{\#}\, \mathtt{d}^{\#} \mathtt{a}^{\#}\, y^{\#}\, \mathtt{b}^{\#}\, x^{\#} \quad \text{and} \quad \nu^{\#} = t^{\#}\, \mathtt{b}'^{\#}\, z^{\#}\, \mathtt{a}^{\#} \mathtt{d}^{\#}\, y^{\#}\, \mathtt{b}^{\#}\, x^{\#}. \tag{3.8}$$

Since $\mathtt{d}^{\#} < \mathtt{b}'^{\#}, \mathtt{b}^{\#} < \mathtt{a}^{\#}$, we have $\sigma^{\#} \rightleftarrows_{\mathrm{B}} \nu^{\#}$. □

3.2. **Connection with the sylvester monoid.** The *sylvester monoid* [HNT02, HNT05] is the quotient of the free monoid A^* by the congruence $\equiv_{\mathrm{S}}$ that is the reflexive and transitive closure of the *sylvester adjacency relation* $\leftharpoonup_{\mathrm{S}}$ defined for $u \in A^*$ and $\mathtt{a}, \mathtt{b}, \mathtt{c} \in A$ by

$$\mathtt{ac}\, u\, \mathtt{b} \leftharpoonup_{\mathrm{S}} \mathtt{ca}\, u\, \mathtt{b} \qquad \text{where} \qquad \mathtt{a} \leqslant \mathtt{b} < \mathtt{c}. \tag{3.9}$$

In the same way, let us define the *#-sylvester monoid*, the quotient of A^* by the congruence $\equiv_{\mathrm{S}\#}$ that is the reflexive and transitive closure of the *#-sylvester adjacency relation* $\leftharpoonup_{\mathrm{S}\#}$ defined for $u \in A^*$ and $\mathtt{a}, \mathtt{b}, \mathtt{c} \in A$ by

$$\mathtt{b}\, u\, \mathtt{ac} \leftharpoonup_{\mathrm{S}\#} \mathtt{b}\, u\, \mathtt{ca} \qquad \text{where} \qquad \mathtt{a} < \mathtt{b} \leqslant \mathtt{c}. \tag{3.10}$$

Note that this adjacency relation is defined by taking the images by the Schützenberger involution of the sylvester adjacency relation. Indeed, for all $u, v \in A^*$, $u \equiv_{\mathrm{S}\#} v$ if and only if $u^{\#} \equiv_{\mathrm{S}} v^{\#}$.

In [HNT05], Hivert, Novelli and Thibon have shown that two words are sylvester equivalent if and only if each gives the same right binary search tree by inserting their letters from right to left using the binary search tree insertion algorithm [AU94]. In our setting, we call this process the *leaf insertion* and it comes in two versions, depending on if the considered binary tree is a right or a left binary search tree:

Algorithm: LeafInsertion.
Input: An A-labeled right (resp. left) binary search tree T, a letter $\mathtt{a} \in A$.
Output: T after the leaf insertion of $\mathtt{a}$.

(1) If $T = \perp$, return the one-node binary search tree labeled by $\mathtt{a}$.
(2) Let $\mathtt{b}$ be the label of the root of T.
(3) If $\mathtt{a} \leqslant \mathtt{b}$ (resp. $\mathtt{a} < \mathtt{b}$):
  (a) Then, recursively leaf insert $\mathtt{a}$ into the left subtree of T.
  (b) Otherwise, recursively leaf insert $\mathtt{a}$ into the right subtree of T.

End.

For further reference, let us recall the following theorem due to Hivert, Novelli and Thibon [HNT05], restated in our setting and supplemented with a respective part:

Theorem 3.5. *Two words are $\equiv_{\mathrm{S}}$-equivalent (resp. $\equiv_{\mathrm{S}\#}$-equivalent) if and only if they give the same right (resp. left) binary search tree by inserting their letters from right to left (resp. left to right).*

In other words, any A-labeled right (resp. left) binary search tree encodes a sylvester (resp. #-sylvester) equivalence class of words of A^*, and conversely.

Let us explain the respective part of Theorem 3.5. It follows from (3.10) that encoding the $\equiv_{\mathrm{S}\#}$-equivalence class of a word u is equivalent to encoding the $\equiv_{\mathrm{S}}$-equivalence class of $u^{\#}$. For this, simply insert u from left to right by considering that the reversed order relation holds between its letters. In this way, we obtain a binary tree such that for any node x labeled by a letter $\mathtt{b}$, all labels $\mathtt{a}$ of the nodes of the left subtree of x, and all labels $\mathtt{c}$ of the nodes of the right subtree of x, the inequality $\mathtt{a} \geqslant \mathtt{b} > \mathtt{c}$ holds. This binary tree is obviously not a left binary search tree. Nevertheless, a left binary search tree can be obtained from it after swapping, for each node, its left and right subtree recursively. One can prove by induction on $|u|$ that this left binary search tree is the one that LeafInsertion constructs by inserting the letters of u from left to right and hence, this remark explains the difference of treatment between right and left binary search trees for the instruction (3) of LeafInsertion.

Lemma 3.6. *Let $u := x\,\mathtt{ac}\,y$ and $v := x\,\mathtt{ca}\,y$ be two words such that x and y are two words, $\mathtt{a} < \mathtt{c}$ are two letters, and $u \equiv_{\mathrm{S}} v$. Then, $u \leftrightharpoons_{\mathrm{S}} v$.*

Proof. Follows from Theorem 3.5: Since u and v give the same right binary search tree T by inserting these from right to left, the node labeled by $\mathtt{a}$ and the node labeled by $\mathtt{c}$ in T cannot be ancestor one of the other. That implies that there exists a node labeled by a letter $\mathtt{b}$, common ancestor of both nodes labeled by $\mathtt{a}$ and $\mathtt{c}$ such that $\mathtt{a} \leqslant \mathtt{b} < \mathtt{c}$. Thus, $u \leftrightharpoons_{\mathrm{S}} v$. □

Lemma 3.6 also proves that the $\leftrightharpoons_{\mathrm{S}}$-adjacency relations of any equivalence class C of $\mathfrak{S}_n/_{\equiv_{\mathrm{S}}}$ are exactly the covering relations of the permutohedron restricted to the elements of C. Note that it is also the case for the $\leftrightharpoons_{\mathrm{S}\#}$-adjacency relations.

The Baxter monoid, the sylvester monoid and the #-sylvester monoid are related in the following way.

Proposition 3.7. *Let $u, v \in A^*$. Then, $u \equiv_{\mathrm{B}} v$ if and only if $u \equiv_{\mathrm{S}} v$ and $u \equiv_{\mathrm{S}\#} v$.*

Proof. ($\Rightarrow$): Once more, it is enough to check the property on adjacency relations. Moreover, by Proposition 3.2, it is enough to check the property for permutations. Let $\sigma, \nu \in \mathfrak{S}_n$ and assume that $\sigma \rightleftarrows_{\mathrm{B}} \nu$. We have $\sigma = x\,\mathtt{b}\,y\,\mathtt{ad}\,z\,\mathtt{b}'\,t$ and $\nu = x\,\mathtt{b}\,y\,\mathtt{da}\,z\,\mathtt{b}'\,t$ for some letters $\mathtt{a} < \mathtt{b}, \mathtt{b}' < \mathtt{d}$ and words x, y, z, and t. The presence of the letters $\mathtt{a}$, $\mathtt{d}$ and $\mathtt{b}'$ with $\mathtt{a} < \mathtt{b}' < \mathtt{d}$ ensures that $\sigma \leftrightharpoons_{\mathrm{S}} \nu$. Besides, the presence of the letters $\mathtt{b}$, $\mathtt{a}$ and $\mathtt{d}$ with $\mathtt{a} < \mathtt{b} < \mathtt{d}$ ensures that $\sigma \leftrightharpoons_{\mathrm{S}\#} \nu$.

($\Leftarrow$): Since the sylvester and the #-sylvester monoids are compatible with the destandardization process [HNT05], it is enough to check the property for permutations. Let $\sigma, \nu \in \mathfrak{S}_n$ such that $\sigma \equiv_{\mathrm{S}} \nu$ and $\sigma \equiv_{\mathrm{S}\#} \nu$. Set $\tau := \inf_{\leqslant_{\mathrm{P}}} \{\sigma, \nu\}$. Since the permutohedron is a lattice, τ is well-defined, and since the equivalence classes of permutations under the $\equiv_{\mathrm{S}}$ and $\equiv_{\mathrm{S}\#}$ congruences are intervals of the permutohedron [HNT05], we have $\sigma \equiv_{\mathrm{S}} \tau \equiv_{\mathrm{S}} \nu$ and $\sigma \equiv_{\mathrm{S}\#} \tau \equiv_{\mathrm{S}\#} \nu$. Moreover, by Lemma 3.6, and again since that the equivalence classes of permutations under the $\equiv_{\mathrm{S}}$ and the $\equiv_{\mathrm{S}\#}$ congruences are intervals of the permutohedron, for each saturated chains $\tau \leqslant_{\mathrm{P}} \sigma' \leqslant_{\mathrm{P}} \cdots \leqslant_{\mathrm{P}} \sigma$ and $\tau \leqslant_{\mathrm{P}} \nu' \leqslant_{\mathrm{P}} \cdots \leqslant_{\mathrm{P}} \nu$, there are sequences of adjacency relations $\tau \leftrightharpoons_{\mathrm{S}} \sigma' \leftrightharpoons_{\mathrm{S}} \cdots \leftrightharpoons_{\mathrm{S}} \sigma$, $\tau \leftrightharpoons_{\mathrm{S}\#} \sigma' \leftrightharpoons_{\mathrm{S}\#} \cdots \leftrightharpoons_{\mathrm{S}\#} \sigma$, $\tau \leftrightharpoons_{\mathrm{S}} \nu' \leftrightharpoons_{\mathrm{S}} \cdots \leftrightharpoons_{\mathrm{S}} \nu$ and $\tau \leftrightharpoons_{\mathrm{S}\#} \nu' \leftrightharpoons_{\mathrm{S}\#} \cdots \leftrightharpoons_{\mathrm{S}\#} \nu$. Hence, $\tau \equiv_{\mathrm{B}} \sigma$ and $\tau \equiv_{\mathrm{B}} \nu$, implying $\sigma \equiv_{\mathrm{B}} \nu$. □

Proposition 3.7 shows that the $\equiv_{\mathrm{B}}$-equivalence classes are the intersection of $\equiv_{\mathrm{S}}$-equivalence classes and $\equiv_{\mathrm{S}\#}$-equivalence classes.

By the characterization of the $\equiv_{\mathrm{B}}$-equivalence classes provided by Proposition 3.7, restricting the Baxter congruence on permutations, we have the following property:

Proposition 3.8. *For any $n \geqslant 0$, each equivalence class of $\mathfrak{S}_n/_{\equiv_{\mathrm{B}}}$ is an interval of the permutohedron.*

Proof. By Proposition 3.7, the $\equiv_{\mathrm{B}}$-equivalence classes are the intersection of the $\equiv_{\mathrm{S}}$ and the $\equiv_{\mathrm{S}\#}$-equivalence classes. Moreover, the permutations under the $\equiv_{\mathrm{S}}$ and the $\equiv_{\mathrm{S}\#}$ equivalence relations are intervals of the permutohedron [HNT05]. The proposition comes from the fact that the intersection of two lattice intervals is also an interval and that the permutohedron is a lattice. □

Lemma 3.9. *Let $u := x\,\mathtt{ad}\,y$ and $v := x\,\mathtt{da}\,y$ such that x and y are two words, $\mathtt{a} < \mathtt{d}$ are two letters, and $u \equiv_{\mathrm{B}} v$. Then, $u \leftrightharpoons_{\mathrm{B}} v$ or $u \rightleftharpoons_{\mathrm{B}} v$.*

Proof. By Proposition 3.7, since $u \equiv_{\mathrm{B}} v$, we have $u \equiv_{\mathrm{S}} v$ and thus by Lemma 3.6 we have $u \leftrightharpoons_{\mathrm{S}} v$, implying the existence of a letter $\mathtt{b}'$ in the factor y satisfying $\mathtt{a} \leqslant \mathtt{b}' < \mathtt{d}$. In the same way, we also have $u \equiv_{\mathrm{S}\#} v$ and thus $u \leftrightharpoons_{\mathrm{S}\#} v$, hence the existence of a letter $\mathtt{b}$ in the factor x satisfying $\mathtt{a} < \mathtt{b} \leqslant \mathtt{d}$. That proves that u and v are $\leftrightharpoons_{\mathrm{B}}$ or $\rightleftharpoons_{\mathrm{B}}$-adjacent. □

Lemma 3.9 is the analog, in the case of the Baxter congruence, of Lemma 3.6 and also proves that the $\leftharpoonup_{\mathrm{B}}$ and $\rightharpoonup_{\mathrm{B}}$-adjacency relations of any equivalence class C of $\mathfrak{S}_n/_{\equiv_{\mathrm{B}}}$ are exactly the covering relations of the permutohedron restricted to the elements of C.

3.3. **Connection with the 3-recoil monoid.** If $\mathtt{a}$ and $\mathtt{c}$ are two letters of A, denote by $\mathtt{c}-\mathtt{a}$ the cardinality of the set $\{\mathtt{b} \in A : \mathtt{a} < \mathtt{b} \leqslant \mathtt{c}\}$. In [NRT11], Novelli, Reutenauer and Thibon defined for any $k \geqslant 0$ the congruence $\equiv_{\mathrm{R}^{(k)}}$. This congruence is the reflexive and transitive closure of the *k-recoil adjacency relation*, defined for $\mathtt{a}, \mathtt{b} \in A$ by

$$\mathtt{ab} \leftharpoonup_{\mathrm{R}^{(k)}} \mathtt{ba} \qquad \text{where} \qquad \mathtt{b}-\mathtt{a} \geqslant k. \tag{3.11}$$

The *k-recoil monoid* is the quotient of the free monoid A^* by the congruence $\equiv_{\mathrm{R}^{(k)}}$. Note that the congruence $\equiv_{\mathrm{R}^{(2)}}$ restricted to permutations is nothing but the *hypoplactic congruence* [Nov98].

The Baxter monoid and the 3-recoil monoid are related in the following way.

Proposition 3.10. *Each $\equiv_{\mathrm{R}^{(3)}}$-equivalence class of permutations can be expressed as a union of some $\equiv_{\mathrm{B}}$-equivalence classes.*

Proof. This amounts to prove that for all permutations σ and ν, if $\sigma \equiv_{\mathrm{B}} \nu$ then $\sigma \equiv_{\mathrm{R}^{(3)}} \nu$. It is enough to check this property on adjacency relations. Hence, assume that $\sigma \rightleftharpoons_{\mathrm{B}} \nu$. We have $\sigma = x\,\mathtt{b}\,y\,\mathtt{ad}\,z\mathtt{b}'\,t$ and $\nu = x\,\mathtt{b}\,y\,\mathtt{da}\,z\,\mathtt{b}'\,t$ for some letters $\mathtt{a} < \mathtt{b}, \mathtt{b}' < \mathtt{d}$ and words x, y, z, and t. Since σ and ν are permutations, $\mathtt{b} \neq \mathtt{b}'$ and thus, we have $\mathtt{a} < \mathtt{b} < \mathtt{b}' < \mathtt{d}$ or $\mathtt{a} < \mathtt{b}' < \mathtt{b} < \mathtt{d}$, implying that $\mathtt{d}-\mathtt{a} \geqslant 3$. Hence, $\sigma \equiv_{\mathrm{R}^{(3)}} \nu$. □

Note that Proposition 3.10 is false for the congruence $\equiv_{\mathrm{R}^{(4)}}$ since there are twenty-two equivalence classes of permutations of size 4 under the congruence $\equiv_{\mathrm{B}}$ but twenty-four under $\equiv_{\mathrm{R}^{(4)}}$. Conversely, note that $\equiv_{\mathrm{R}^{(4)}}$ is not a refinement of $\equiv_{\mathrm{B}}$ since for any $n \geqslant 5$, the permutation $1.n.n-1\ldots 2$ is the only member of its $\equiv_{\mathrm{B}}$-equivalence class but not of its $\equiv_{\mathrm{R}^{(4)}}$-equivalence class.

Moreover, it is clear, by definition of $\equiv_{\mathrm{R}^{(k)}}$, that the $\equiv_{\mathrm{R}^{(k)}}$-equivalence classes of permutations are union of $\equiv_{\mathrm{R}^{(k+1)}}$-equivalence classes. Hence, by Proposition 3.10, the hypoplactic equivalence classes of permutations are union of some $\equiv_{\mathrm{B}}$-equivalence classes.

4. A Robinson-Schensted-like algorithm

The goal of this section is to define an analog to the Robinson-Schensted algorithm for the Baxter monoid—see [LS81, Lot02] for the usual Robinson-Schensted insertion algorithm that associate to any word u its P-symbol, that is a Young tableau.

The interest of the Baxter monoid in our context is that the equivalence classes of the permutations of size n under the Baxter congruence are equinumerous with unlabeled pairs of twin binary trees with n nodes, and thus, by the results of Dulucq and Guibert [DG94], also equinumerous with Baxter permutations of size n. We shall provide a proof of this property in this section, using our analog of the Robinson-Schensted algorithm.

4.1. **Principle of the algorithm.** We describe here an algorithm testing if two words are equivalent according to the Baxter congruence. Given a word $u \in A^*$, it computes its *Baxter P-symbol*, that is an A-labeled pair (T_L, T_R) consisting in a left and a right binary search tree such that the nondecreasing rearrangement of u is the inorder reading of both T_L and T_R. It also computes its *Baxter Q-symbol*, that is a pair of twin binary trees (S_L, S_R) where S_L (resp. S_R) is an increasing (resp. decreasing) binary tree, such that the inorder reading of S_L and S_R are the same. Moreover, T_L and S_L have same shape, and so have T_R and S_R.

4.1.1. *The Baxter* P*-symbol.*

Definition 4.1. *The* Baxter P-symbol *(or simply* P-symbol *if the context is clear) of a word $u \in A^*$ is the pair $\mathsf{P}(u) = (T_L, T_R)$ where T_L (resp. T_R) is the left (resp. right) binary search tree obtained by leaf inserting the letters of u from left to right (resp. right to left).*

Figure 6 shows the P-symbol of $u := 2415253$. Before showing that the P-symbol of Definition 4.1 can be used to decide if two words are equivalent under the Baxter congruence, let us give an intuitive explanation of its validity.

Recall that, according to Proposition 3.7, to represent the Baxter equivalence class of a word u, one has to represent both the equivalence class of u under the $\equiv_{\mathrm{S}}$ congruence and the equivalence class of u under the $\equiv_{\mathrm{S}^\#}$ congruence. This is exactly what the Baxter P-symbol does since, for a word u, it computes a pair (T_L, T_R) where, by Theorem 3.5, T_L represents the $\equiv_{\mathrm{S}^\#}$-equivalence class of u and T_R represents the $\equiv_{\mathrm{S}}$-equivalence class of u.

4.1.2. *The Baxter* Q*-symbol.* Let us first recall two algorithms. Let u be a word. Define $\mathrm{incr}(u)$, the *increasing binary tree of* u recursively by

$$\mathrm{incr}(u) := \begin{cases} \bot & \text{if } u = \epsilon, \\ \mathrm{incr}(v) \wedge_{\mathtt{a}} \mathrm{incr}(w) & \text{where } u = v\mathtt{a}w,\ \mathtt{a} = \min(u), \text{ and } \mathtt{a} < \min(v). \end{cases} \tag{4.1}$$

In the same way, define the *decreasing binary tree of* u $\mathrm{decr}(u)$, by

$$\mathrm{decr}(u) := \begin{cases} \bot & \text{if } u = \epsilon, \\ \mathrm{decr}(v) \wedge_{\mathtt{b}} \mathrm{decr}(w) & \text{where } u = v\mathtt{b}w,\ \mathtt{b} = \max(u), \text{ and } \mathtt{b} > \max(w). \end{cases} \tag{4.2}$$

Definition 4.2. *The* Baxter Q-symbol *(or simply* Q-symbol *if the context is clear) of a word $u \in A^*$ is the pair $\mathsf{Q}(u) = (S_L, S_T)$ where*

$$S_L := \mathrm{incr}\left(\mathrm{std}(u)^{-1}\right) \qquad \text{and} \qquad S_R := \mathrm{decr}\left(\mathrm{std}(u)^{-1}\right). \tag{4.3}$$

Figure 6 shows the Q-symbol of $u := 2415253$, whose standardized word is 2516374, so that $\mathrm{std}(u)^{-1} = 3157246$.

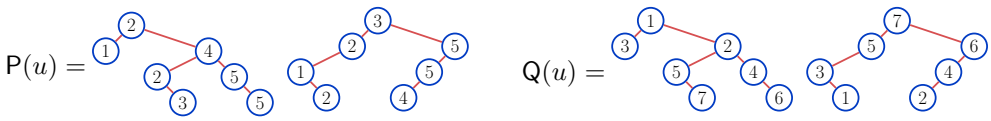


FIGURE 6. The P-symbol and the Q-symbol of $u := 2415253$.

It is plain that given a word u, the Q-symbol of u allows, in addition with its P-symbol, to retrieve the original word. Indeed, if $\mathsf{P}(u) = (T_L, T_R)$ and $\mathsf{Q}(u) = (S_L, S_R)$, the pair (T_R, S_R) is the output of the Robinson-Schensted-like algorithm in the context of the sylvester monoid, which is a bijection between words and pairs of such binary trees [HNT05]. Given (T_R, S_R), it amounts to reading the labels of T_R in the order of the corresponding labels in S_R. The same holds of the pair (T_L, S_L).

4.2. **Correctness of the insertion algorithm.**

Lemma 4.3. *Let T be a non-empty binary tree and y be the i-th leaf of T. If y is left-oriented, it is attached to the i-th node of T. If y is right-oriented, it is attached to the $i-1$-st node of T.*

Proof. We proceed by structural induction on the set of non-empty binary trees. If T is the one-node binary tree, the lemma is clearly satisfied. Otherwise, we have $T = A \wedge B$. Let y be the i-th leaf of T and x be the node where y is attached. If y is also in A and $A = \perp$, y is left-oriented and is attached to the root of T (that is the first node of T) and the lemma is satisfied. If y is in A and $A \neq \perp$, y is also the i-th leaf of A and x is a node of A, so that the lemma follows by induction hypothesis on A. Otherwise, y is in B. If $B = \perp$, y is right-oriented and is attached to the root of T (that is the last node of T) and the lemma is satisfied. Otherwise, y is the $i-(n+1)$-st leaf of B where n is the number of nodes of A. Assume that the node x is the j-st node of T, then, x becomes the $j-(n+1)$-st node of B. Hence, the lemma follows by induction hypothesis on B. □

The following proposition is the key of our construction.

Proposition 4.4. *Let σ be a permutation and T be the left binary search tree obtained by left leaf insertions of the letters of σ, from left to right. Then, the $i+1$-st leaf of T is right-oriented if and only if i is a recoil of σ.*

Proof. Set $\mathtt{a} := i$ and $\mathtt{c} := i+1$. Assume that $\mathtt{a}$ is a recoil of σ. We have $\sigma = u\,\mathtt{c}\,v\,\mathtt{a}\,w$ for some words u, v, and w. Since no letter $\mathtt{b}$ of u and v satisfies $\mathtt{a} < \mathtt{b} < \mathtt{c}$, the node of T labeled by $\mathtt{c}$ has a node labeled by $\mathtt{a}$ in its left subtree, itself having no right child and thus contributes, by Lemma 4.3, to a right-oriented leaf in position $i+1$.

Conversely, assume that $\mathtt{a}$ is not a recoil of σ. We have $\sigma = u\,\mathtt{a}\,v\,\mathtt{c}\,w$ for some words u, v, and w. For the same reason as before, the node of T labeled by $\mathtt{a}$ has a node labeled by $\mathtt{c}$ in its right subtree, itself having no left child and thus contributes, by Lemma 4.3, to a left-oriented leaf in position $i+1$. □

Figure 7 shows an example of application of Proposition 4.4.

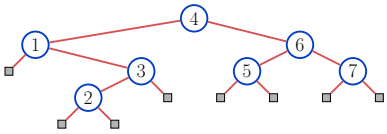


FIGURE 7. The binary search tree drawn with its leaves obtained by left leaf insertions of the letters of $\sigma := 4136275$, from left to right. The recoils of σ are 2, 3, 5, and 7 and the 3-rd, 4-th, 6-th, and 8-th leaves of this binary tree are right-oriented.

4.2.1. *The* P*-symbol.*

Proposition 4.5. *For any word $u \in A^*$, the* P*-symbol (T_L, T_R) of u is a pair of twin binary search trees—T_L (resp. T_R) is a left (resp. right) binary search tree, and the inorder reading of both T_L and T_R is the nondecreasing rearrangement of u.*

Proof. Note by definition of the LEAFINSERTION algorithm that T_L (resp. T_R) is a left (resp. right) binary search tree and the inorder reading of both T_L and T_R is the nondecreasing rearrangement of u. It is plain that the leaf insertion of u and $\mathrm{std}(u)$ from left to right (resp.

right to left) into left (resp. right) binary search trees give binary trees of same shape. That implies that we can consider that $u =: \sigma$ is a permutation. Proposition 4.4 implies that the canopies of T_L and T_R are complementary because i is a recoil of σ if and only if i is not a recoil of $\sigma^\sim$. Thus, the shapes of T_L and T_R consist in a pair of twin binary trees. □

Theorem 4.6. *Let $u, v \in A^*$. Then, $u \equiv_{\mathrm{B}} v$ if and only if $\mathsf{P}(u) = \mathsf{P}(v)$.*

Proof. Assume $u \equiv_{\mathrm{B}} v$. Then, by Proposition 3.7, u and v are $\equiv_{\mathrm{S}}$ and $\equiv_{\mathrm{S}^\#}$-equivalent. Hence, by Theorem 3.5, u and v have the same sylvester and #-sylvester P-symbol, so that $\mathsf{P}(u) = \mathsf{P}(v)$.

Conversely assume that $\mathsf{P}(u) = \mathsf{P}(v) =: (T_L, T_R)$. Since the leaf insertion of both u and v from left to right gives T_L, we have, by Theorem 3.5, $u \equiv_{\mathrm{S}^\#} v$. In addition, the leaf insertion of both u and v from right to left gives T_R, so that, by the just cited theorem, $u \equiv_{\mathrm{S}} v$. By Proposition 3.7, we have $u \equiv_{\mathrm{B}} v$. □

In the case of permutations, each $\equiv_{\mathrm{B}}$-equivalence class can be encoded by an unlabeled pair of twin binary trees because there is one unique way to bijectively label a binary tree with n nodes on $\{1, \dots, n\}$ such that it is a binary search tree. Hence, in the sequel, unlabeled pairs of twin binary search trees can be considered as labeled by a permutation, and conversely.

4.2.2. *The Q-symbol.* Let us recall the following lemma of [HNT05], restated in our setting and supplemented with a respective part:

Lemma 4.7. *Let u be a word and $\sigma := \mathrm{std}(u)^{-1}$. The right (resp. left) binary search tree obtained by inserting u from right to left (resp. from left to right) and $\mathrm{decr}(\sigma)$ (resp. $\mathrm{incr}(\sigma)$) have same shape.*

Proposition 4.8. *For any word $u \in A^*$, the shape of the Q-symbol (S_L, S_R) of u is a pair of twin binary trees. Moreover, S_L is an increasing binary tree, S_R is a decreasing binary tree and their inorder reading is $\mathrm{std}(u)^{-1}$.*

Proof. By definition of the Q-symbol, S_L and S_R are respectively the increasing and the decreasing binary trees of $\sigma := \mathrm{std}(u)^{-1}$. By Lemma 4.7, a binary tree with same shape as S_L (resp. S_R) can also be obtained by leaf insertions of the letters of σ^{-1} from left to right (resp. right to left). Thus, by Proposition 4.4, the shape of (S_L, S_R) is a pair of twin binary trees. Moreover, by the definition of the algorithms incr and decr, we can prove by induction on the size of σ that the binary trees S_L and S_R have both σ as inorder reading. □

Theorem 4.9. *The map $u \longmapsto (\mathsf{P}(u), \mathsf{Q}(u))$ is a bijection between the elements of A^* and the set formed by the pairs $((T_L, T_R), (S_L, S_R))$ where*

- *(i) (T_L, T_R) is a pair of twin binary search trees—T_L (resp. T_R) is a left (resp. right) binary search tree, and T_L and T_R have both the same inorder reading;*
- *(ii) (S_L, S_R) is a pair of twin binary trees where S_L (resp. S_R) is an increasing (resp. decreasing) binary tree, and S_L and S_R have both the same inorder reading;*
- *(iii) (T_L, T_R) and (S_L, S_R) have same shape.*

Proof. Let us first show that for any $u \in A^*$, the pair $(\mathsf{P}(u), \mathsf{Q}(u))$ satisfies the assertions of the theorem. Point (i) follows from Proposition 4.5. Point (ii) follows from Proposition 4.8. Moreover, by Lemma 4.7, Point (iii) checks out. Besides, as already mentioned, it is possible to reconstruct from the pair $(\mathsf{P}(u), \mathsf{Q}(u))$ the word u and such a word is unique. That shows that the correspondence is well-defined and injective.

Conversely, assume that $((T_L, T_R), (S_L, S_R))$ satisfies the three assertions of the theorem. According to [HNT02], there is a bijection between the elements of A^* and the pairs (T_R, S_R) where T_R is a right binary search tree and S_R a decreasing binary tree of same shape. Let u

be the word in correspondence with (T_R, S_R). In the same way, there is a bijection between the elements of A^* and the pairs (T_L, S_L) where T_L is a left binary search tree and S_L an increasing binary tree of same shape. Let v be the word in correspondence with (T_L, S_L). By hypothesis, T_L and T_R have both the same inorder reading, implying $\mathrm{ev}(u) = \mathrm{ev}(v)$. In the same way, since S_L and S_R have both the same inorder reading, one has $\mathrm{std}(u)^{-1} = \mathrm{std}(v)^{-1}$. Hence, we have $\mathrm{std}(u) = \mathrm{std}(v)$ and thus $u = v$. Note also that the pair (T_L, S_L) is entirely determined by the pair (T_R, S_R) and conversely. Now, again according to [HNT02], the pair (T_R, S_R) is the sylvester P-symbol of u and the pair (T_L, S_L) is the #-sylvester P-symbol of u. Hence, the insertion of u gives the pair $((T_L, T_R), (S_L, S_R))$, showing that the correspondence is also surjective. □

4.3. **Distinguished permutations from a pair of twin binary trees.** We present in this section some algorithms to read some distinguished permutations from a pair of twin binary search trees. Let us first start with a useful characterization of $\equiv_{\mathrm{B}}$-equivalence classes.

4.3.1. *Baxter equivalence classes as linear extensions of posets.* Let T be an A-labeled binary tree. We shall denote by $\triangle(T)$ (resp. $\triangledown(T)$) the poset $(N, \leqslant)$ where $N := \{1, \dots, n\}$, n is the number of nodes of T, and $\leqslant$ is defined, for $i, j \in N$, by

$$i \leqslant j \qquad \text{if the } i\text{-th node is an ancestor (resp. descendant) of the } j\text{-th node of } T. \tag{4.4}$$

If the sequence $i_1 \dots i_n$ is a linear extension of $\triangle(T)$ (resp. $\triangledown(T)$), we shall also say that the word $u_1 \dots u_n$ is a *linear extension* of $\triangle(T)$ (resp. $\triangledown(T)$) if for any $1 \leqslant \ell \leqslant n$, the label of the i_ℓ-th node of T is u_ℓ.

The words of a sylvester equivalence class encoded by a labeled right binary search tree T coincide with the linear extensions of $\triangledown(T)$ (see Note 4 of [HNT05]). Additionally, this also says that the words of a #-sylvester equivalence class encoded by a labeled left binary search tree T are exactly the linear extensions of $\triangle(T)$. One has a similar characterization of Baxter equivalence classes:

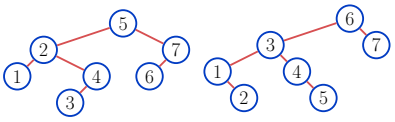
Proposition 4.10. *The words of a Baxter equivalence class encoded by a pair of twin binary search trees (T_L, T_R) coincide with the words that are both linear extensions of the posets $\triangle(T_L)$ and $\triangledown(T_R)$.*

Proof. Let u be a word belonging to the Baxter equivalence class encoded by (T_L, T_R). By Theorem 4.6, T_L (resp. T_R) can be obtained by leaf inserting u from left to right (resp. right to left). Hence, if $i \leqslant j$ in $\triangle(T_L)$ (resp. in $\triangledown(T_R)$) then i is smaller than j as integers. Thus, u is a linear extension of both $\triangle(T_L)$ and $\triangledown(T_R)$.

Assume now that u is a linear extension of $\triangle(T_L)$ and $\triangledown(T_R)$ and let v be any word of the Baxter equivalence class encoded by (T_L, T_R). By Theorem 4.6, T_L (resp. T_R) can be obtained by leaf inserting v from left to right (resp. right to left). Note 4 of [HNT05] implies that $u \equiv_{\mathrm{S}^\#} v$ and $u \equiv_{\mathrm{S}} v$. Hence, by Proposition 3.7, one has $u \equiv_{\mathrm{B}} v$, showing that u also belongs to the Baxter equivalence class represented by (T_L, T_R). □

To illustrate Proposition 4.10, consider the following labeled pair of twin binary search trees,

$$(T_L, T_R) := \tag{4.5}$$



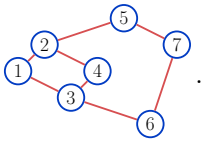
The set of words that are linear extensions of $\triangle(T_L)$ and $\triangledown(T_R)$ are (the highlighted permutation is a Baxter permutation)

$$\begin{aligned}\{&\mathbf{5214376},\ 5214736,\ 5217436,\ 5241376,\ 5241736,\\ &5247136,\ 5271436,\ 5274136,\ 5721436,\ 5724136\},\end{aligned} \tag{4.6}$$

which is exactly the Baxter equivalence class encoded by (T_L, T_R).

Note that it is possible to represent the order relations induced by the posets $\triangle(T_L)$ and $\triangledown(T_R)$ in only one poset $\triangle(T_L)\cup\triangledown(T_R)$, adding on $\triangle(T_L)$ the order relations induced by $\triangledown(T_R)$. For the previous example, we obtain the poset

$$\triangle(T_L)\cup\triangledown(T_R) = \tag{4.7}$$



4.3.2. *Extracting Baxter permutations.* The following algorithm allows, given an A-labeled pair of twin binary search trees (T_L, T_R), to compute a word belonging to the $\equiv_{\mathrm{B}}$-equivalence class encoded by (T_L, T_R). When (T_L, T_R) is labeled by a permutation, our algorithm coincides with the algorithm designed by Dulucq and Guibert to describe a bijection between pairs of twin binary trees and Baxter permutations [DG94]. Besides, since their algorithm always computes a Baxter permutation, our algorithm also returns a Baxter permutation when (T_L, T_R) is labeled by a permutation.

Algorithm: ExtractBaxter.
Input: An A-labeled pair of twin binary search trees (T_L, T_R).
Output: A word belonging to the Baxter equivalence class encoded by (T_L, T_R).

(1) Let $u := \epsilon$ be the empty word.
(2) While $T_L \neq \perp$ and $T_R \neq \perp$:
  (a) Let $\mathtt{a}$ be the label of the root of T_L.
  (b) Let i be the index of root of T_L.
  (c) Set $u := u\mathtt{a}$.
  (d) Let A (resp. B) be the left (resp. right) subtree of T_L.
  (e) If the i-th node of T_R is a left child in T_R:
    (i) Then, set $T_L := A \diagup B$.
    (ii) Otherwise, set $T_L := A \diagdown B$.
  (f) Suppress the i-th node in T_R.
(3) Return u.

End.

Figure 8 shows an execution of this algorithm.

The results of Dulucq and Guibert [DG94] imply that ExtractBaxter terminates. The only thing to prove is that the computed word belongs to the $\equiv_{\mathrm{B}}$-equivalence class encoded by the pair of twin binary search trees as input. For that, let us first prove the following lemma.

Lemma 4.11. *Let (T_L, T_R) be a non-empty pair of twin binary trees. If the root of T_L is the i-th node of T_L, then, the i-th node of T_R has no child.*

Proof. Assume that $T_L = A \wedge B$. Note that if both A and B are empty, T_L and T_R are the one-node binary trees and the lemma is clearly satisfied.

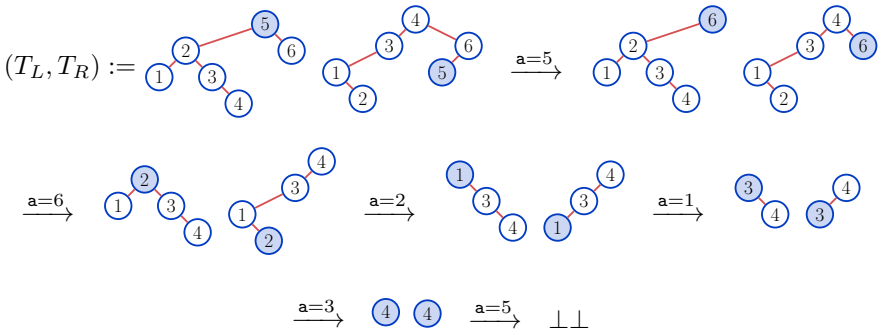


FIGURE 8. An execution of the algorithm EXTRACTBAXTER on (T_L, T_R). The computed Baxter permutation is 562134.

If $A \neq \bot$, assume that the i-th node of T_R has a non-empty left subtree. That implies that the i-th leaf of T_R is not attached to its i-th node. Thus, by Lemma 4.3, the i-th leaf of T_R is attached to its $i-1$-st node and is right-oriented. In T_L, the i-th leaf cannot be attached to its i-th node because $A \neq \bot$. Hence, by Lemma 4.3, the i-th leaf of T_L is also attached to its $i-1$-st node and is right-oriented. Since T contains at least i nodes, there is at least $i+1$ leaves in T, implying that the i-th leaf is not the rightmost leaf of T_L and T_R, and thus (T_L, T_R) is not a pair of twin binary trees, contradicting the hypothesis.

Assume now that the i-th node of T_R has a non-empty right subtree. That implies that the $i+1$-st leaf of T_R is not attached to its i-th node and thus, by Lemma 4.3, the $i+1$-st leaf of T_R is left-oriented. Moreover, since the i-th node of T_R has a non-empty right subtree and the i-th node of T_L is its root, the i-th node of T_L also has a non-empty right subtree. That implies that the $i+1$-st leaf of T_L is not attached to its i-th node and thus, by Lemma 4.3, the $i+1$-st leaf of T_R is also left-oriented. That contradicts that (T_L, T_R) is a pair of twin binary trees, and implies that the i-th node of T_R has no child. The case $B \neq \bot$ is analogous. □

Proposition 4.12. *For any A-labeled pair of twin binary search trees (T_L, T_R) as input, the algorithm* EXTRACTBAXTER *computes a word belonging to the $\equiv_{\mathrm{B}}$-equivalence class encoded by (T_L, T_R). Moreover, if (T_L, T_R) is labeled by a permutation, the computed word is a Baxter permutation.*

Proof. Let us prove by induction on n, that is the number of nodes of T_L and T_R, that if (T_L, T_R) is an A-labeled pair of twin binary search trees, then EXTRACTBAXTER returns a word that is a linear extension of $\triangle(T_L)$ and a linear extension of $\triangledown(T_R)$, *i.e.*, by Proposition 4.10, a word belonging to the $\equiv_{\mathrm{B}}$-equivalence class encoded by (T_L, T_R). This property clearly holds for $n \leqslant 1$. Now, assume that $T_L = A \wedge_{\mathtt{a}} B$ where $\mathtt{a}$ is the label of the root of T_L. By Lemma 4.11, if the root of T_L is its i-th node, the i-th node x of T_R has no child. Moreover, since T_L and T_R are binary search trees and labeled by a same word, their respective i-th nodes have the same label $\mathtt{a}$. Moreover, the canopy of T_L is of the form $v01w$ where $v := \mathrm{cnp}(A)$ and $w := \mathrm{cnp}(B)$, and the canopy of T_R is of the form $v'10w'$ where v' (resp. w') is the complementary of v (resp. w) since that (T_L, T_R) is a pair of twin binary trees. We have now two cases whether x is a left of right child in T_R.

If x is a left child in T_R, the algorithm returns the word $\mathtt{a}u$ where u is the word obtained by applying the algorithm on (T'_L, T'_R) where $T'_L = A \,/\, B$ and T'_R is obtained from T_R by suppressing the node x. First, the canopy of T'_L is of the form $v0w$ and the canopy of T'_R

is of the form $v'1w'$. Moreover, T'_L and T'_R are clearly still binary search trees. That implies that (T'_L, T'_R) is a pair of twin binary search trees. By induction hypothesis and Proposition 4.10, the word u belongs to the $\equiv_{\mathrm{B}}$-equivalence class encoded by (T'_L, T'_R), and thus, $\mathtt{a}u$ belongs to the $\equiv_{\mathrm{B}}$-equivalence class encoded by (T_L, T_R) because $\mathtt{a}u$ is a linear extension of $\triangle(T_L)$ (resp. $\triangledown(T_R)$) since u is a linear extension of both $\triangle(T'_L)$ and $\triangledown(T'_R)$. The case where x is a right child in T_R is analogous.

Finally, when (T_L, T_R) is labeled by a permutation, ExtractBaxter coincides with the algorithm of Dulucq and Guibert [DG94] and computes a Baxter permutation. □

The validity of ExtractBaxter implies the two following results.

Theorem 4.13. *For any $n \geqslant 0$, there is a bijection between the set of Baxter equivalence classes of words of length n and A-labeled pairs of twin binary search trees with n nodes.*

Proof. By Proposition 4.5 and Theorem 4.6, the P-symbol algorithm induces an injection between the set of equivalence classes of $\mathfrak{S}_n/_{\equiv_{\mathrm{B}}}$ and the set of unlabeled pairs of twin binary trees. Moreover, by Proposition 4.12, the algorithm ExtractBaxter exhibits a surjection between these two sets. Hence, these two sets are in bijection. □

Theorem 4.13 implies in particular that the Baxter equivalence classes of permutations of size n are in bijection with pairs of twin binary trees labeled by a permutation (or equivalently with unlabeled pairs of twin binary trees).

Theorem 4.14. *For any $n \geqslant 0$, each equivalence class of $\mathfrak{S}_n/_{\equiv_{\mathrm{B}}}$ contains exactly one Baxter permutation.*

Proof. Let C be an equivalence class of $\mathfrak{S}_n/_{\equiv_{\mathrm{B}}}$. By Theorem 4.13, C can be represented by an unlabeled pair of twin binary trees J. By Proposition 4.12, the algorithm ExtractBaxter computes a permutation belonging to the $\equiv_{\mathrm{B}}$-equivalence class encoded by J, showing that each $\equiv_{\mathrm{B}}$-equivalence class of permutations contains at least one Baxter permutation. The theorem follows from the fact that Baxter permutations are equinumerous with unlabeled pairs of twin binary trees. □

4.3.3. *Extracting minimal and maximal permutations.* Reading defined in [Rea05] *twisted Baxter permutations*, that are the permutations avoiding the generalized permutation patterns $2-41-3$ and $3-41-2$. These permutations are particular elements of Baxter classes of permutations:

Proposition 4.15. *Twisted Baxter permutations coincide with minimal elements of Baxter equivalence classes of permutations.*

Proof. First, note that by Proposition 3.8, every Baxter equivalence class of permutations has a minimal element. Assume that σ is minimal of its $\equiv_{\mathrm{B}}$-equivalence class of permutations. Then, it is not possible to perform any rewriting of the form

$$\mathtt{b}\,u\,\mathtt{da}\,v\,\mathtt{b}' \to \mathtt{b}\,u\,\mathtt{ad}\,v\,\mathtt{b}', \tag{4.8}$$

where $\mathtt{a} < \mathtt{b}, \mathtt{b}' < \mathtt{d}$ are letters, and u and v are words. Hence, σ avoids the patterns $2-41-3$ and $3-41-2$, and is a twisted Baxter permutation.

Conversely, if σ is a twisted Baxter permutation, it avoids $2-41-3$ and $3-41-2$ and it is not possible to perform any rewriting $\to$, so that, by Proposition 3.8 and Lemma 3.9, it is minimal of its $\equiv_{\mathrm{B}}$-equivalence class. □

In a similar way, by calling *anti-twisted Baxter permutation* any permutation that avoids the generalized permutation patterns $2-14-3$ and $3-14-2$, an analogous proof to the one of Proposition 4.15 shows that anti-twisted Baxter permutations coincide with maximal elements of Baxter equivalence classes of permutations.

Proposition 4.15 implies that twisted Baxter permutations, anti-twisted Baxter permutations, and Baxter permutations are equinumerous since by Theorem 4.14 there is exactly one Baxter permutation by $\equiv_{\mathrm{B}}$-equivalence class of permutations and by Proposition 3.8, there is also exactly one twisted (and one anti-twisted) Baxter permutation. This suggests among other that there exists a bijection sending a Baxter permutation to the twisted Baxter permutation of its $\equiv_{\mathrm{B}}$-equivalence class.

As pointed out by Law and Reading, West has shown first a bijection between Baxter permutations and twisted Baxter permutations using generating trees [BM03]. In our setting, as in the setting of Law and Reading [LR12], this bijection is the one preserving the classes. Here follows an algorithm to compute this bijection.

Let us consider the following algorithm which allows, given an A-labeled pair of twin binary search trees (T_L, T_R), to compute the minimal permutation for the lexicographic order belonging to the $\equiv_{\mathrm{B}}$-equivalence class encoded by (T_L, T_R).

Algorithm: ExtractMin.
Input: An A-labeled pair of twin binary search trees (T_L, T_R).
Output: The minimal word for the lexicographic order of the class encoded by (T_L, T_R).

(1) Let $u := \epsilon$ be the empty word.
(2) Let $F := T_L$ be a rooted forest.
(3) While F is not empty and $T_R \neq \perp$:
   (a) Let i be the smallest index such that the i-th node of F is a root and the i-th node of T_R has no child.
   (b) Let $\mathtt{a}$ be the label of the i-th node of T_L.
   (c) Set $u := u\mathtt{a}$.
   (d) Suppress the i-th node of F and the i-th node of T_R.
(4) Return u.

End.

Note that, by choosing in the instruction (3a) the greatest index instead of the smallest, the previous algorithm would compute the maximal word for the lexicographic order of the $\equiv_{\mathrm{B}}$-equivalence class encoded by (T_L, T_R). Let us call this variant ExtractMax.

Figure 9 shows an example of application of ExtractMin.

Proposition 4.16. *For any A-labeled pair of twin binary search trees (T_L, T_R) as input, the algorithm* ExtractMin *(resp.* ExtractMax*) computes the minimal (resp. maximal) word for the lexicographic order of the $\equiv_{\mathrm{B}}$-equivalence class encoded by (T_L, T_R). Moreover, if (T_L, T_R) is labeled by a permutation, the computed word is the minimal (resp. maximal) permutation for the permutohedron order of its $\equiv_{\mathrm{B}}$-equivalence class.*

Proof. The output u of the algorithm ExtractMin (resp. ExtractMax) is both a linear extension of $\triangle(T_L)$ and a linear extension of $\triangledown(T_R)$. That implies by Proposition 4.10 that u belongs to the $\equiv_{\mathrm{B}}$-equivalence class encoded by the input pair of twin binary trees. Moreover, this algorithm terminates since by Theorem 4.14, each A-labeled pair of twin binary search trees (T_L, T_R) admits at least one word that is a common linear extension of $\triangle(T_L)$ and $\triangledown(T_R)$.

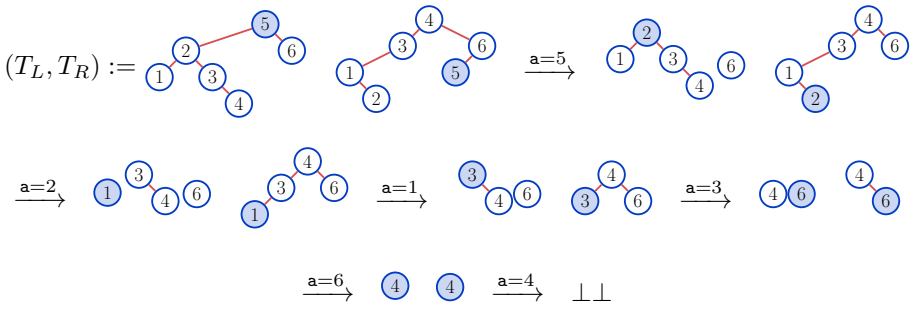


FIGURE 9. An execution of the algorithm EXTRACTMIN on (T_L, T_R). The computed permutation is 521364 and it is minimal in its $\equiv_{\mathrm{B}}$ -equivalence class.

The minimality (resp. maximality) for the lexicographic order of the computed word comes from the fact that at each step, the node that has the smallest (resp. greatest) label is chosen.

Finally, since the lexicographic order is a linear extension of the permutohedron order, and by Proposition 3.8, since Baxter equivalence classes are intervals of the permutohedron, EXTRACTMIN (resp. EXTRACTMAX) returns the minimal (resp. maximal) permutation for the permutohedron order of its Baxter equivalence class. □

By Proposition 4.16 and using our Robinson-Schensted-like algorithm, we can compute the bijection between Baxter permutations and twisted Baxter permutations in the following way: If σ is a Baxter permutation, apply EXTRACTMIN on $\mathsf{P}(\sigma)$ to obtain its corresponding twisted Baxter permutation. Conversely, if σ is a twisted Baxter permutation, apply EXTRACTBAXTER on $\mathsf{P}(\sigma)$ to obtain its corresponding Baxter permutation.

In the same way, we can compute a bijection between Baxter permutations and anti-twisted Baxter permutations using EXTRACTMAX instead of EXTRACTMIN. Moreover, these algorithms give a bijection between twisted Baxter permutations and anti-twisted Baxter permutations: If σ is a twisted (resp. anti-twisted) Baxter permutation, apply EXTRACTMAX (resp. EXTRACTMIN) on $\mathsf{P}(\sigma)$ to obtain its corresponding anti-twisted (resp. twisted) Baxter permutation.

4.4. **Definition and correctness of the iterative insertion algorithm.** In what follows, we shall revise our P-symbol algorithm that we have presented in Section 4.1 to make it iterative. Indeed, we propose an insertion algorithm such that, for any word u such that $\mathsf{P}(u) = (T_L, T_R)$ and any letter $\mathtt{a}$, the insertion of $\mathtt{a}$ into (T_L, T_R) is the pair of twin binary trees $\mathsf{P}(u\mathtt{a})$. This, besides being in agreement with the usual Robinson-Schensted-like algorithms, has the merit to allow to compute in the Baxter monoid. Indeed, this gives a simple way to compute the concatenation of two words u and v under the Baxter congruence simply by inserting the letters of the word uv into the pair $(\perp, \perp)$. Note that one can compute the product of two pairs of twin binary trees (T_L, T_R) and (T'_L, T'_R) by computing a word u' that belongs to the $\equiv_{\mathrm{B}}$ -equivalence class of (T'_L, T'_R) by applying the algorithm EXTRACTMIN (or EXTRACTBAXTER) with (T'_L, T'_R) as input, and then, by inserting the letters of u' from left to right into (T_L, T_R).

4.4.1. *Root insertion in binary search trees.* Let T be an A-labeled right binary search tree and $\mathtt{b}$ a letter of A. The *lower restricted binary tree* of T compared to $\mathtt{b}$, namely $T_{\leqslant \mathtt{b}}$, is the right binary search tree uniquely made of the nodes x of T labeled by letters $\mathtt{a}$ satisfying $\mathtt{a} \leqslant \mathtt{b}$ and such that for all nodes x and y of $T_{\leqslant \mathtt{b}}$, if x is ancestor of y in $T_{\leqslant \mathtt{b}}$, then x is also ancestor

of y in T. In the same way, we define the *higher restricted binary tree* of T compared to $\mathtt{b}$, namely $T_{>\mathtt{b}}$ (see Figure 10).

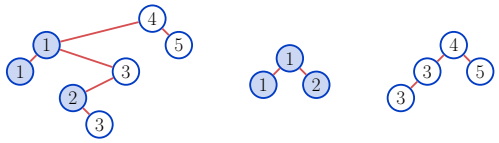


FIGURE 10. A right binary search tree T, $T_{\leqslant 2}$ and $T_{>2}$.

Let T be an A-labeled right binary search tree and $\mathtt{a}$ a letter of A. The *root insertion* of $\mathtt{a}$ into T consists in modifying T so that the root of T is a new node labeled by $\mathtt{a}$, its left subtree is $T_{\leqslant \mathtt{a}}$ and its right subtree is $T_{>\mathtt{a}}$.

4.4.2. *The iterative insertion algorithm.*

Definition 4.17. *Let (T_L, T_R) be an A-labeled pair of twin binary search trees and $\mathtt{a}$ be a letter. The* insertion *of $\mathtt{a}$ into (T_L, T_R) consists in making a leaf insertion of $\mathtt{a}$ into T_L and a root insertion of $\mathtt{a}$ into T_R. The* iterative Baxter P-symbol *(or simply* iterative P-symbol *if the context is clear) of a word $u \in A^*$ is the pair $\mathsf{P}(u) = (T_L, T_R)$ computed by iteratively inserting the letters of u, from left to right, into $(\perp, \perp)$. The* iterative Baxter Q-symbol *(or simply* iterative Q-symbol *if the context is clear) of $u \in A^*$ is the pair $\mathsf{Q}(u) = (S_L, S_R)$ of same shape as $\mathsf{P}(u)$ and such that each node is labeled by its date of creation in $\mathsf{P}(u)$.*

Figure 11 shows, step by step, the computation of the iterative Baxter P and Q-symbols of a word.

4.4.3. *Correctness of the iterative insertion algorithm.* To show that the iterative version of the Baxter P-symbol computes the same labeled pair of twin binary trees than its non-iterative version, we need the following lemma.

Lemma 4.18. *Let $u \in A^*$. Let T be the right binary search tree obtained by root insertions of the letters of u, from left to right. Let T' be the right binary search tree obtained by leaf insertions of the letters of u, from right to left. Then, $T = T'$.*

Proof. Let us proceed by induction on $|u|$. If $u = \epsilon$, the lemma is satisfied. Otherwise, assume that $u = v\mathtt{a}$ where $\mathtt{a} \in A$. Let S be the right binary search tree obtained by root insertions of the letters of v from left to right. By induction hypothesis, S also is the right binary tree obtained by leaf insertions of the letters of v from right to left. The right binary search tree T obtained by root insertions of u from left to right satisfies, by definition, $T = S_{\leqslant \mathtt{a}} \wedge_{\mathtt{a}} S_{>\mathtt{a}}$. The right binary search tree T' obtained by leaf insertions of u from right to left satisfies $T' = L' \wedge_{\mathtt{a}} R'$ where the subtree L' only depends on the subword $v_{\leqslant \mathtt{a}} := v_{]-\infty,\mathtt{a}]}$ and the subtree R' only depends on the subword $v_{>\mathtt{a}} := v_{]\mathtt{a},+\infty[}$, so that, by induction hypothesis, $L' = S_{\leqslant \mathtt{a}}$, $R' = S_{>\mathtt{a}}$ and thus, $T = T'$. □

Proposition 4.19. *For any $u \in A^*$, the Baxter P-symbol of u and the iterative Baxter P-symbol of u are equal.*

Proof. Let (T_L, T_R) be the P-symbol of u and (T'_L, T'_R) be the iterative P-symbol of u. By definition of these two insertion algorithms, we have $T_L = T'_L$. Moreover, T_R is obtained by leaf insertions of the letters of u from right to left and T'_R is obtained by root insertions of the letters of u from left to right. By Lemma 4.18, we have $T_R = T'_R$. □

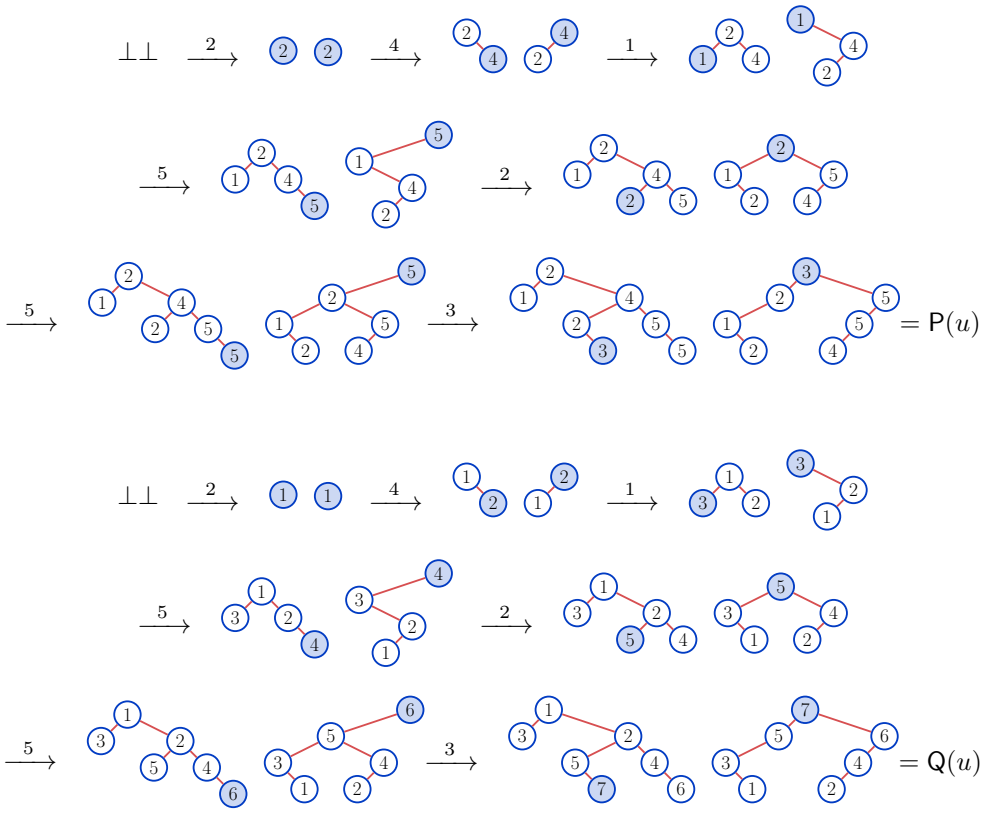


FIGURE 11. Steps of the computation of the P-symbol and the Q-symbol of $u := 2415253$.

The correctness of the iterative version of the Q-symbol algorithm comes from the correctness of the iterative P-algorithm.

5. The Baxter lattice

5.1. **The Baxter lattice congruence.** Recall that an equivalence relation $\equiv$ on the elements of a lattice $(L, \wedge, \vee)$ is a *lattice congruence* if for all $x, x', y, y' \in L$, $x \equiv x'$ and $y \equiv y'$ imply $x \wedge y \equiv x' \wedge y'$ and $x \vee y \equiv x' \vee y'$. The quotient $L/_{\equiv}$ of L by $\equiv$ is naturally a lattice. Indeed, by denoting by $\tau : L \to L/_{\equiv}$ the canonical projection, the set $L/_{\equiv}$ is endowed with meet and join operations defined by $\widehat{x} \wedge \widehat{y} := \tau(x \wedge y)$ and $\widehat{x} \vee \widehat{y} := \tau(x \vee y)$ for all $\widehat{x}, \widehat{y} \in L/_{\equiv}$ where x and y are any elements of L such that $\tau(x) = \widehat{x}$ and $\tau(y) = \widehat{y}$.

Lattices congruences admit the following very useful order-theoretic characterization [CS98, Rea05]. An equivalence relation $\equiv$ on the elements of a lattice $(L, \wedge, \vee)$ seen as a poset $(L, \leqslant)$ is a lattice congruence is the following three conditions hold.

(L1) Every $\equiv$-equivalence class is an interval of L;
(L2) For any $x, y \in L$, if $x \leqslant y$ then $x{\downarrow} \leqslant y{\downarrow}$ where $x{\downarrow}$ is the maximal element of the $\equiv$-equivalence class of x;

(L3) For any $x, y \in L$, if $x \leqslant y$ then $x\!\uparrow \leqslant y\!\uparrow$ where $x\!\uparrow$ is the minimal element of the $\equiv$-equivalence class of x.

For any permutation σ, let us denote by $\sigma\!\downarrow$ (resp. $\sigma\!\uparrow$) the maximal (resp. minimal) permutation of the $\equiv_{\mathrm{B}}$-equivalence class of σ for the permutohedron order. Note by Proposition 3.8 that $\sigma\!\downarrow$ and $\sigma\!\uparrow$ are well-defined.

Theorem 5.1. *The Baxter equivalence relation is a lattice congruence of the permutohedron.*

Proof. By Proposition 3.8, any Baxter equivalence class of permutations is an interval of the permutohedron, so that (L1) checks out. One just has to show that $\equiv_{\mathrm{B}}$ satisfies (L2) and (L3).

Let σ and ν two permutations such that $\sigma \leqslant_{\mathrm{P}} \nu$. Let us show that $\sigma\!\downarrow \leqslant_{\mathrm{P}} \nu\!\downarrow$. It is enough to check the property when $\nu = \sigma s_i$ where s_i is an elementary transposition and i is not a descent of σ. If $\sigma = \sigma\!\downarrow$, then $\sigma\!\downarrow \leqslant_{\mathrm{P}} \nu \leqslant_{\mathrm{P}} \nu\!\downarrow$ and the property holds. Otherwise, by Lemma 3.9, there exists an elementary transposition s_j and a permutation π such that π and σ are $\rightleftarrows_{\mathrm{B}}$-adjacent, $\pi = \sigma s_j$ and $\sigma \leqslant_{\mathrm{P}} \pi$. It then remains to prove that there exists a permutation μ such that $\nu \equiv_{\mathrm{B}} \mu$ and $\pi \leqslant_{\mathrm{P}} \mu$. Indeed, this leads to show, by applying iteratively this reasoning, that $\sigma\!\downarrow$ is smaller than a permutation belonging to the $\equiv_{\mathrm{B}}$-equivalence class of ν for the permutohedron order and hence, by transitivity, that $\sigma\!\downarrow \leqslant_{\mathrm{P}} \nu\!\downarrow$. We have four cases:

Case 1: If $j \leqslant i-2$, σ is of the form $\sigma = u\,\mathtt{ab}\,v\,\mathtt{cd}\,w$ where u, v, and w are some words and $\mathtt{a}$ (resp. $\mathtt{c}$) is the j-th (resp. i-th) letter of σ. One has $\mathtt{a} < \mathtt{b}$ and $\mathtt{c} < \mathtt{d}$ since i and j are not descents of σ. We have $\nu = u\,\mathtt{ab}\,v\,\mathtt{dc}\,w$ and $\nu s_j = u\,\mathtt{ba}\,v\,\mathtt{dc}\,w =: \mu$. Moreover, since $\pi \rightleftarrows_{\mathrm{B}} \sigma$, there are some letters $\mathtt{x} \in \mathrm{Alph}(u)$ and $\mathtt{y} \in \mathrm{Alph}(v\,\mathtt{cd}\,w)$ such that $\mathtt{a} < \mathtt{x}, \mathtt{y} < \mathtt{b}$. Thus, $\mu \rightleftarrows_{\mathrm{B}} \nu$. Finally, since $\pi = u\,\mathtt{ba}\,v\,\mathtt{cd}\,w$, $\pi \leqslant_{\mathrm{P}} \mu$, so that μ is appropriate.
Case 2: If $j \geqslant i+2$, this is analogous to the previous case.
Case 3: If $j = i+1$, σ is of the form $\sigma = u\,\mathtt{abc}\,v$ where u and v are some words and $\mathtt{a}$ is the i-th letter of σ. One has $\mathtt{a} < \mathtt{b} < \mathtt{c}$ since i and j are not descents of σ. Since $\sigma \rightleftarrows_{\mathrm{B}} \pi$, there are some letters $\mathtt{x} \in \mathrm{Alph}(u)$ and $\mathtt{y} \in \mathrm{Alph}(v)$ such that $\mathtt{b} < \mathtt{x}, \mathtt{y} < \mathtt{c}$. Thus, since $\nu = u\,\mathtt{bac}\,v$ and $\mathtt{a} < \mathtt{b} < \mathtt{x}, \mathtt{y} < \mathtt{c}$, we have $\nu s_j = u\,\mathtt{bca}\,v \rightleftarrows_{\mathrm{B}} \nu$. Moreover, $\nu s_j s_i = u\,\mathtt{cba}\,v =: \mu$ and $\nu s_j \rightleftarrows_{\mathrm{B}} \nu s_j s_i$ since $\mathtt{b} < \mathtt{x}, \mathtt{y} < \mathtt{c}$ and thus, $\mu \equiv_{\mathrm{B}} \nu$. Finally, since $\pi = u\,\mathtt{acb}\,v$, we have $\pi \leqslant_{\mathrm{P}} \mu$, and hence μ is appropriate.
Case 4: If $j = i-1$, this is analogous to the previous case.

Hence, the Baxter equivalence relation satisfies (L2). The proof that $\equiv_{\mathrm{B}}$ satisfies (L3) is analogous. □

5.2. **A lattice structure over the set of pairs of twin binary trees.** Recall that by Theorem 4.13, the Baxter equivalence classes of permutations are in correspondence with unlabeled pairs of twin binary trees. Thus, the quotient of the permutohedron of order n by the Baxter congruence is a lattice $(\mathcal{TBT}_n, \leqslant_{\mathrm{B}})$ where the *Baxter order relation* $\leqslant_{\mathrm{B}}$ satisfies, for any $J_0, J_1 \in \mathcal{TBT}_n$,

$$J_0 \leqslant_{\mathrm{B}} J_1 \quad \text{if and only if} \quad \begin{array}{c}\text{there are } \sigma, \nu \in \mathfrak{S}_n \text{ such that} \\ \sigma \leqslant_{\mathrm{P}} \nu,\ \mathsf{P}(\sigma) = J_0 \text{ and } \mathsf{P}(\nu) = J_1.\end{array} \tag{5.1}$$

Let us call *Baxter lattice* the lattice $(\mathcal{TBT}_n, \leqslant_{\mathrm{B}})$. Figure 12 shows the $\equiv_{\mathrm{B}}$-equivalence classes in the permutohedron of order 4 that form the Baxter lattice $(\mathcal{TBT}_4, \leqslant_{\mathrm{B}})$.

5.3. **Covering relations of the Baxter lattice.** Let us describe the covering relations of the lattice $(\mathcal{TBT}_n, \leqslant_{\mathrm{B}})$ in terms of operations on pairs of twin binary trees. Consider a Baxter equivalence class $\widehat{\sigma}$ of permutations encoded by a pair of twin binary trees (T_L, T_R). Let σ by the maximal element of $\widehat{\sigma}$. If i is a descent of σ, the permutation σs_i is not in $\widehat{\sigma}$, and, by

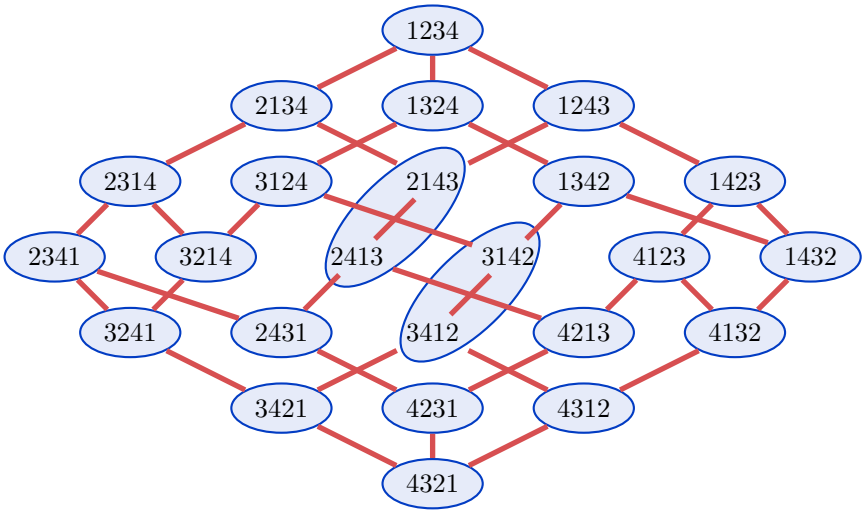


FIGURE 12. The permutohedron of order 4 cut into Baxter equivalence classes.

definition of the Baxter lattice, the pair of twin binary trees $\mathsf{P}(\sigma s_i) =: (T'_L, T'_R)$ covers (T_L, T_R). The permutations σ and σs_i satisfy

$$\sigma = u\,\mathtt{ad}\,v \qquad \text{and} \qquad \sigma s_i = u\,\mathtt{da}\,v, \tag{5.2}$$

where $\mathtt{a} < \mathtt{d}$. There are three cases whether the factor u or v contains a letter $\mathtt{b}$ satisfying $\mathtt{a} < \mathtt{b} < \mathtt{d}$. Since the quotient of the permutohedron by the sylvester congruence is the Tamari lattice [HNT05] and that covering relations in the Tamari lattice are binary tree rotations, the covering relations of the Baxter lattice are the following:

- (C1) If there is a letter $\mathtt{b}$ in v such that $\mathtt{a} < \mathtt{b} < \mathtt{d}$, then $T'_R = T_R$ and T'_L is obtained from T_L by performing a left rotation that does not change its canopy;
- (C2) If there is a letter $\mathtt{b}$ in u such that $\mathtt{a} < \mathtt{b} < \mathtt{d}$, then $T'_L = T_L$ and T'_R is obtained from T_R by performing a right rotation that does not change its canopy;
- (C3) If for any letter $\mathtt{b}$ of u and v, one has $\mathtt{b} < \mathtt{a}$ or $\mathtt{d} < \mathtt{b}$, then T'_L (resp. T'_R) is obtained from T_L (resp. T_R) by performing a left (resp. right) rotation that changes its canopy.

Hence, according to this characterization of the covering relations of the Baxter lattice and the definition of the Tamari lattice, we have, for any pairs of twin binary trees (T_L, T_R) and (T'_L, T'_R),

$$(T_L, T_R) \leqslant_{\mathrm{B}} (T'_L, T'_R) \qquad \text{if and only if} \qquad T'_L \leqslant_{\mathrm{T}} T_L \text{ and } T_R \leqslant_{\mathrm{T}} T'_R. \tag{5.3}$$

Note that a right rotation at root y in a binary tree T changes its canopy if and only if the right subtree B of the left child x of y is empty (see Figure 1). Similarly, a left rotation at root y changes the canopy of T if and only if the left subtree B of y is empty. Moreover, if y is the i-th node of T, by Lemma 4.3, one can see that B is the i-th leaf of T. Hence, the right (resp. left) rotation at root y changes the orientation of the i-th leaf of T formerly on the right to the left (resp. left to the right).

5.4. **Twin Tamari diagrams.** The purpose of this section is to introduce *twin Tamari diagrams*. These diagrams are in bijection with pairs of twin binary trees and provide a useful realization of the Baxter lattice since it appears that testing if two twin Tamari diagrams are comparable under the Baxter order relation is immediate.

5.4.1. *Tamari diagrams and the Tamari order relation.* Pallo introduced in [Pal86] words in bijection with binary trees (see also [Knu06]). We call *Tamari diagrams* these words and to compute the Tamari diagram $\mathrm{td}(T)$ of a binary tree T, just label each node x of T by the number of nodes in the right subtree of x and then, consider its inorder reading.

Any Tamari diagram δ of length n satisfies the following two inequalities:

(1) $0 \leqslant \delta_i \leqslant n - i$, for all $1 \leqslant i \leqslant n$;
(2) $\delta_{i+j} \leqslant \delta_i - j$, for all $1 \leqslant i \leqslant n$ and $1 \leqslant j \leqslant \delta_i$.

The main interest of Tamari diagrams is that they offer a very simple way to test if two binary trees are comparable in the Tamari lattice [Knu06]. Indeed, if T and T' are two binary trees with n nodes, one has

$$T \leqslant_{\mathrm{T}} T' \quad \text{if and only if} \quad \mathrm{td}(T)_i \leqslant \mathrm{td}(T')_i \quad \text{for all } 1 \leqslant i \leqslant n. \tag{5.4}$$

5.4.2. *Twin Tamari diagrams and the Baxter order relation.*

Definition 5.2. *A* twin Tamari diagram *of size n is a pair $\left(\delta^L, \delta^R\right)$ such that δ^L and δ^R are Tamari diagrams of length n and for all index $1 \leqslant i \leqslant n-1$, exactly one letter among δ^L_i and δ^R_i is zero.*

Note that we can represent any twin Tamari diagram $\delta := \left(\delta^L, \delta^R\right)$ in a more compact way by a word $\omega(\delta)$ were

$$\omega(\delta)_i := \begin{cases} -\delta^L_i & \text{if } \delta^L_i \neq 0, \\ \delta^R_i & \text{otherwise,} \end{cases} \tag{5.5}$$

for all $1 \leqslant i \leqslant n$ where n is the size of δ. We graphically represent a twin Tamari diagram δ by drawing for each index i a column of $|\omega(\delta)_i|$ boxes facing up if $\omega(\delta)_i \geqslant 0$ and facing down otherwise. First twin Tamari diagrams are drawn in Figure 13.

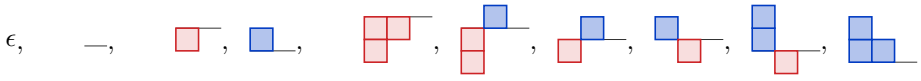


FIGURE 13. First twin Tamari diagrams of size 0, 1, 2, and 3.

Proposition 5.3. *For any $n \geqslant 0$, the set of twin Tamari diagrams of size n is in bijection with the set of pairs of twin binary trees with n nodes. Moreover, this bijection is expressed as follows: If $J := (T_L, T_R)$ is a pair of twin binary trees, the twin Tamari diagram in bijection with J is $\mathrm{ttd}(J) := (\mathrm{td}(T_L), \mathrm{td}(T_R))$.*

Proof. Let us show that the application ttd is well-defined, that is $\mathrm{ttd}(J) =: \left(\delta^L, \delta^R\right)$ is a twin Tamari diagram. Fix an index $1 \leqslant i \leqslant n-1$. By contradiction, assume first that $\delta^L_i = \delta^R_i = 0$. By definition of td, this implies that the i-th nodes of T_L and T_R have no right child. Hence, by Lemma 4.3, the i+1-st leaves of T_L and T_R are attached to its i-th nodes and are right-oriented. Since $i \leqslant n-1$, these leaves are not the rightmost leaves of T_L and T_R, implying that T_L and T_R have not complementary canopies, and hence that (T_L, T_R) is not a pair of twin binary trees. Assume now that $\delta^L_i \neq 0$ and $\delta^R_i \neq 0$. By definition of td, this implies that the i-th nodes of T_L and T_R have a right child. Hence, by Lemma 4.3, the i+1-st leaves of T_L and T_R are attached to its i+1-st nodes and are left-oriented. This implies again that (T_L, T_R) is not a pair of twin binary trees. Thus, ttd computes twin Tamari diagrams.

Now, since td is a bijection between the set of binary trees with n nodes and Tamari diagrams of size n [Pal86], for any twin Tamari diagram δ, there is a unique pair of binary trees J such that $\mathrm{ttd}(J) = \delta$. Using very similar arguments as above, one can prove that the canopies of the trees of J are complementary, and hence, that J is a pair of twin binary trees. □

Figure 14 shows an example of a pair of twin binary trees with the corresponding twin Tamari diagram.

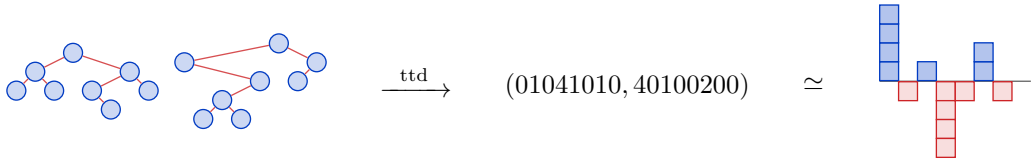


FIGURE 14. A pair of twin binary trees, the corresponding twin Tamari diagram via the bijection ttd and its graphical representation.

Proposition 5.4. *Let J_0 and J_1 two pairs of twin binary trees with n nodes. We have*

$$J_0 \leqslant_{\mathrm{B}} J_1 \quad \text{if and only if} \quad \omega\left(\mathrm{ttd}\left(J_0\right)\right)_i \leqslant \omega\left(\mathrm{ttd}\left(J_1\right)\right)_i \quad \text{for all } 1 \leqslant i \leqslant n. \tag{5.6}$$

Proof. This result is a direct consequence of the characterization of the Baxter order relation (5.3) using the Tamari order relation, the characterization furnished by (5.4) to compare two binary trees in the Tamari lattice with Tamari diagrams, and the bijection between pairs of twin binary trees and Twin Tamari diagrams provided by Proposition 5.3. □

Figure 15 shows an interval of the Baxter lattice.

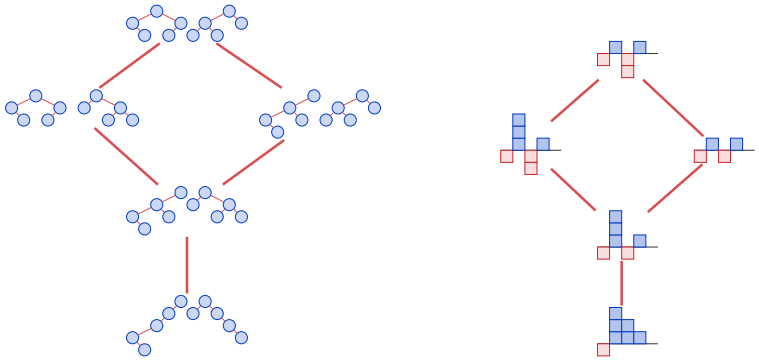


FIGURE 15. An interval of the Baxter lattice of order 5 where vertices are seen as pairs of twin binary trees and as Twin Tamari diagrams.

6. The Hopf algebra of pairs of twin binary trees

In the sequel, all the algebraic structures have a field of characteristic zero $\mathbb{K}$ as ground field.

6.1. The Hopf algebra FQSym and construction of Hopf subalgebras.

6.1.1. *The Hopf algebra* **FQSym**. Recall that the family $\{\mathbf{F}_\sigma\}_{\sigma\in\mathfrak{S}}$ forms the *fundamental* basis of **FQSym**, the Hopf algebra of Free quasi-symmetric functions [MR95, DHT02]. Its product and its coproduct are defined by

$$\mathbf{F}_\sigma \cdot \mathbf{F}_\nu := \sum_{\pi \in \sigma \,\overline{\sqcup}\, \nu} \mathbf{F}_\pi, \tag{6.1}$$

$$\Delta(\mathbf{F}_\sigma) := \sum_{\sigma=uv} \mathbf{F}_{\mathrm{std}(u)} \otimes \mathbf{F}_{\mathrm{std}(v)}. \tag{6.2}$$

For example,

$$\begin{aligned}\mathbf{F}_{132}\cdot\mathbf{F}_{12} = {} & \mathbf{F}_{13245}+\mathbf{F}_{13425}+\mathbf{F}_{13452}+\mathbf{F}_{14325}+\mathbf{F}_{14352}\\ & +\mathbf{F}_{14532}+\mathbf{F}_{41325}+\mathbf{F}_{41352}+\mathbf{F}_{41532}+\mathbf{F}_{45132},\end{aligned} \tag{6.3}$$

$$\begin{aligned}\Delta(\mathbf{F}_{35142}) = {} & 1\otimes\mathbf{F}_{35142}+\mathbf{F}_1\otimes\mathbf{F}_{4132}+\mathbf{F}_{12}\otimes\mathbf{F}_{132}\\ & +\mathbf{F}_{231}\otimes\mathbf{F}_{21}+\mathbf{F}_{2413}\otimes\mathbf{F}_1+\mathbf{F}_{35142}\otimes 1.\end{aligned} \tag{6.4}$$

Set $\mathbf{G}_\sigma := \mathbf{F}_{\sigma^{-1}}$. Recall that **FQSym** is isomorphic to its dual $\mathbf{FQSym}^\star$ through the map $\psi : \mathbf{FQSym} \to \mathbf{FQSym}^\star$ defined by $\psi(\mathbf{F}_\sigma) := \mathbf{F}^\star_{\sigma^{-1}} = \mathbf{G}^\star_\sigma$.

Recall also that **FQSym** admits a *polynomial realization* [DHT02], that is an injective algebra morphism $r_A : \mathbf{FQSym} \hookrightarrow \mathbb{K}\langle A\rangle$. Furthermore, this map should be compatible with the coalgebra structure in the sense that the coproduct of an element can be computed by taking its image by r_A, and then by applying the *alphabet doubling trick* [DHT02, Hiv07]. This map is defined by

$$r_A(\mathbf{G}_\sigma) := \sum_{\substack{u\in A^*\\ \mathrm{std}(u)=\sigma}} u. \tag{6.5}$$

For example,

$$r_A(\mathbf{G}_\epsilon) = 1, \tag{6.6}$$

$$r_A(\mathbf{G}_1) = \sum_i a_i = a_1 + a_2 + a_3 + \cdots, \tag{6.7}$$

$$r_A(\mathbf{G}_{231}) = \sum_{k<i\leqslant j} a_i a_j a_k = a_2a_2a_1 + a_2a_3a_1 + a_2a_4a_1 + \cdots. \tag{6.8}$$

6.1.2. *Construction of Hopf subalgebras of* **FQSym**. If $\equiv$ is an equivalence relation on $\mathfrak{S}$ and $\sigma\in\mathfrak{S}$, let us denote by $\widehat{\sigma}$ the $\equiv$-equivalence class of σ.

The following theorem contained in an unpublished note of Hivert and Nzeutchap [HN07] (see also [DHT02, Hiv07]) shows that an equivalence relation on A^* satisfying some properties can be used to define Hopf subalgebras of **FQSym**:

Theorem 6.1. *Let $\equiv$ be an equivalence relation defined on A^*. If $\equiv$ is a congruence, compatible with the restriction of alphabet intervals and compatible with the destandardization process, then the family $\{\mathbf{P}_{\widehat{\sigma}}\}_{\widehat{\sigma}\in\mathfrak{S}/_\equiv}$ defined by*

$$\mathbf{P}_{\widehat{\sigma}} := \sum_{\nu\in\widehat{\sigma}} \mathbf{F}_\nu, \tag{6.9}$$

spans a Hopf subalgebra of **FQSym**.

The compatibility with the destandardization process and with the restriction of alphabet intervals imply that for any $\mathbf{F}_\pi$ appearing in a product $\mathbf{P}_{\widehat{\sigma}} \cdot \mathbf{P}_{\widehat{\nu}}$ and any permutation $\pi' \equiv \pi$, $\mathbf{F}_{\pi'}$ also appears in the product. Moreover, the compatibility with the destandardization process and the fact that $\equiv$ is a congruence imply that for any $\mathbf{F}_\sigma \otimes \mathbf{F}_\nu$ appearing in a coproduct $\Delta\left(\mathbf{P}_{\widehat{\pi}}\right)$ and any permutations $\sigma' \equiv \sigma$ and $\nu' \equiv \nu$, $\mathbf{F}_{\sigma'} \otimes \mathbf{F}_{\nu'}$ also appears in the coproduct.

In the sequel, we shall call $\{\mathbf{P}_{\widehat{\sigma}}\}_{\widehat{\sigma} \in \mathfrak{S}/_\equiv}$ the *fundamental basis* of the corresponding Hopf subalgebra of **FQSym**.

6.2. **Construction of the Hopf algebra Baxter.** By Theorem 4.13, the $\equiv_{\mathrm{B}}$-equivalence classes of permutations can be encoded by unlabeled pairs of twin binary trees. Moreover, in the sequel, the P-symbols of permutations are regarded as unlabeled pairs of twin binary trees since there is only one way to label a pair of twin binary trees with a permutation so that it is a pair of twin binary search trees. Hence, in our graphical representations we will only represent their shape.

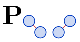
Since by definition $\equiv_{\mathrm{B}}$ is a congruence, since by Propositions 3.2 and 3.3, $\equiv_{\mathrm{B}}$ satisfies the conditions of Theorem 6.1, and since by Theorem 4.6, the permutations σ such that $\mathsf{P}(\sigma) = J$ coincide with the Baxter equivalence class represented by the pair of twin binary trees J, we have the following theorem.

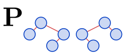
Theorem 6.2. *The family* $\{\mathbf{P}_J\}_{J \in \mathcal{TBT}}$ *defined by*

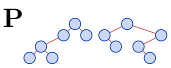
$$\mathbf{P}_J := \sum_{\substack{\sigma \in \mathfrak{S} \\ \mathsf{P}(\sigma) = J}} \mathbf{F}_\sigma, \tag{6.10}$$

spans a Hopf subalgebra of **FQSym**, *namely the Hopf algebra* **Baxter**.

For example,

(6.11) $\mathbf{P}$  $= \mathbf{F}_{12},$

(6.12) $\mathbf{P}$  $= \mathbf{F}_{2143} + \mathbf{F}_{2413},$

(6.13) $\mathbf{P}$  $= \mathbf{F}_{542163} + \mathbf{F}_{542613} + \mathbf{F}_{546213}.$

The Hilbert series of **Baxter** is

$$B(z) := 1 + z + 2z^2 + 6z^3 + 22z^4 + 92z^5 + 422z^6 + 2074z^7 + 10754z^8 + 58202z^9 + \cdots, \tag{6.14}$$

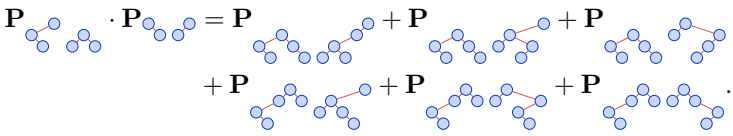
the generating series of Baxter permutations (sequence **A001181** of [Slo]).

By Theorem 6.1, the product of **Baxter** is well-defined. We deduce it from the product of **FQSym**, and, since by Theorem 4.14 there is exactly one Baxter permutation in any $\equiv_{\mathrm{B}}$-equivalence class of permutations, we obtain

$$\mathbf{P}_{J_0} \cdot \mathbf{P}_{J_1} = \sum_{\substack{\mathsf{P}(\sigma) = J_0,\ \mathsf{P}(\nu) = J_1 \\ \pi \in \sigma \,\overline{\sqcup\!\sqcup}\, \nu \,\cap\, \mathfrak{S}^{\mathrm{B}}}} \mathbf{P}_{\mathsf{P}(\pi)}. \tag{6.15}$$

For example,

(6.16)

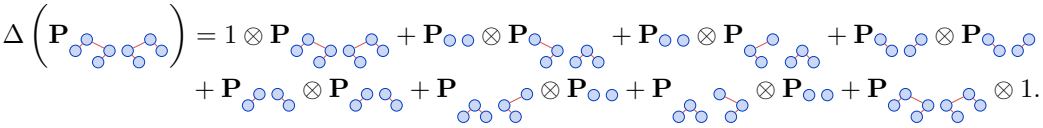


In the same way, we deduce the coproduct of **Baxter** from the coproduct of **FQSym** and by Theorem 4.14, we obtain

$$\Delta(\mathbf{P}_J) = \sum_{\substack{uv \in \mathfrak{S} \\ \mathsf{P}(uv)=J \\ \sigma:=\mathrm{std}(u),\ \nu:=\mathrm{std}(v) \in \mathfrak{S}^{\mathrm{B}}}} \mathbf{P}_{\mathsf{P}(\sigma)} \otimes \mathbf{P}_{\mathsf{P}(\nu)}. \tag{6.17}$$

For example,

(6.18)



6.3. **Properties of the Hopf algebra Baxter.**

6.3.1. *A polynomial realization.* We deduce a polynomial realization of **Baxter** from the one of **FQSym**. In this section, we shall use the notation $J_0 \simeq J_1$ to say that the labeled pairs of twin binary trees J_0 and J_1 have same shape.

Theorem 6.3. *The map* $r_A : \mathbf{Baxter} \to \mathbb{K}\langle A\rangle$ *defined by*

$$r_A(\mathbf{P}_J) := \sum_{\substack{u \in A^* \\ (\mathrm{incr}(u),\ \mathrm{decr}(u)) \simeq J}} u, \tag{6.19}$$

for any $J \in \mathcal{TBT}$ *provides a polynomial realization of* **Baxter**.

Proof. Let us apply the polynomial realization r_A of **FQSym** defined in (6.5) on elements of the fundamental basis of **Baxter**:

$$r_A(\mathbf{P}_J) = \sum_{\substack{\sigma \in \mathfrak{S} \\ \mathsf{P}(\sigma)=J}} r_A(\mathbf{F}_\sigma), \tag{6.20}$$

$$= \sum_{\substack{\sigma \in \mathfrak{S} \\ \mathsf{P}(\sigma^{-1})=J}} r_A(\mathbf{G}_\sigma), \tag{6.21}$$

$$= \sum_{\substack{\sigma \in \mathfrak{S} \\ (\mathrm{incr}(\sigma),\ \mathrm{decr}(\sigma)) \simeq J}} r_A(\mathbf{G}_\sigma), \tag{6.22}$$

$$= \sum_{\substack{\sigma \in \mathfrak{S} \\ (\mathrm{incr}(\sigma),\ \mathrm{decr}(\sigma)) \simeq J}} \sum_{\substack{u \in A^* \\ \mathrm{std}(u)=\sigma}} u, \tag{6.23}$$

The equality between (6.21) and (6.22) follows from Lemma 4.7. The equality between (6.23) and the right member of (6.19) follows from the fact that $\mathrm{incr}(\sigma) \simeq \mathrm{incr}(u)$ (resp. $\mathrm{decr}(\sigma) \simeq \mathrm{decr}(u)$) whenever $\mathrm{std}(u) = \sigma$. □

6.3.2. *The dual Hopf algebra.* We denote by $\{\mathbf{P}^\star_J\}_{J\in\mathcal{TBT}}$ the dual basis of the basis $\{\mathbf{P}_J\}_{J\in\mathcal{TBT}}$. The Hopf algebra $\mathbf{Baxter}^\star$, dual of $\mathbf{Baxter}$, is a quotient Hopf algebra of $\mathbf{FQSym}^\star$. More precisely,

$$\mathbf{Baxter}^\star = \mathbf{FQSym}^\star / I, \tag{6.24}$$

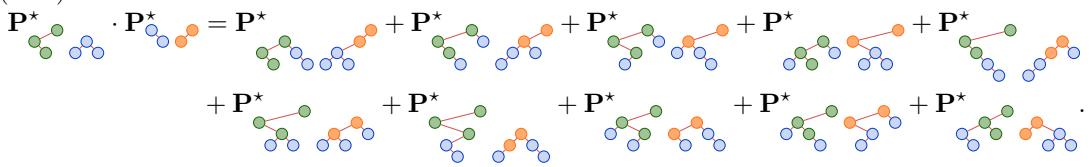
where I is the Hopf ideal of $\mathbf{FQSym}^\star$ spanned by the elements $(\mathbf{F}^\star_\sigma - \mathbf{F}^\star_\nu)$ whenever $\sigma \equiv_{\mathrm{B}} \nu$.

Let $\phi : \mathbf{FQSym}^\star \twoheadrightarrow \mathbf{Baxter}^\star$ be the canonical projection, mapping $\mathbf{F}^\star_\sigma$ on $\mathbf{P}^\star_{\mathsf{P}(\sigma)}$. By definition, the product of $\mathbf{Baxter}^\star$ is

$$\mathbf{P}^\star_{J_0} \cdot \mathbf{P}^\star_{J_1} = \phi\left(\mathbf{F}^\star_\sigma \cdot \mathbf{F}^\star_\nu\right), \tag{6.25}$$

where σ and ν are any permutations such that $\mathsf{P}(\sigma) = J_0$ and $\mathsf{P}(\nu) = J_1$. Note that due to the fact that $\mathbf{Baxter}^\star$ is a quotient of $\mathbf{FQSym}^\star$, the number of terms occurring in a product $\mathbf{P}^\star_{J_0} \cdot \mathbf{P}^\star_{J_1}$ only depends on the number m (resp. n) of nodes of J_0 (resp. J_1) and is $\binom{m+n}{m}$. For example,

(6.26)

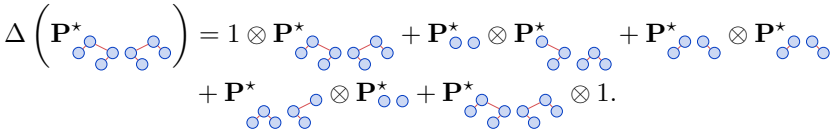


In the same way, the coproduct of $\mathbf{Baxter}^\star$ is

$$\Delta(\mathbf{P}_J) = (\phi\otimes\phi)\left(\Delta\left(\mathbf{F}^\star_\sigma\right)\right), \tag{6.27}$$

where σ is any permutation such that $\mathsf{P}(\sigma) = J$. Note that the number of terms occurring in a coproduct $\Delta\left(\mathbf{P}_J\right)$ only depends on the number n of nodes of each binary trees of J and is $n+1$. For example,

(6.28)



Following Fomin [Fom94] (see also [BLL08]), we can build a *pair of graded graphs in duality* $(G_{\mathbf{P}}, G_{\mathbf{P}^\star})$. The set of vertices of $G_{\mathbf{P}}$ and $G_{\mathbf{P}^\star}$ is the set of pairs of twin binary trees. There is an edge between the vertices J and J' in $G_{\mathbf{P}}$ (resp. in $G_{\mathbf{P}^\star}$) if $\mathbf{P}_{J'}$ (resp. $\mathbf{P}^\star_{J'}$) appears in the product $\mathbf{P}_J \cdot \mathbf{P}_{\circ\circ}$ (resp. in the product $\mathbf{P}^\star_J \cdot \mathbf{P}^\star_{\circ\circ}$). Figure 16 (resp. Figure 17) shows the graded graph $G_{\mathbf{P}}$ (resp. $G_{\mathbf{P}^\star}$) restricted to vertices of order smaller than 5.

6.3.3. *A boolean basis.* We shall call a basis of an algebra (resp. coalgebra) a *boolean algebra basis* (resp. *boolean coalgebra basis*) if each element of the basis (resp. tensor square of the basis) only occurs with coefficient 0 or 1 in any product (resp. coproduct) involving two (resp. one) elements of the basis.

Proposition 6.4. *If $\equiv$ is an equivalence relation defined on A^* satisfying the conditions of Theorem 6.1 and additionally, for all $\pi, \mu \in \mathfrak{S}$,*

$$\sigma, \nu \in \pi \boxplus \mu \quad \text{and} \quad \sigma^{-1} \equiv \nu^{-1} \qquad \text{imply} \qquad \sigma = \nu, \tag{6.29}$$

then, the family $\{\mathbf{P}_{\widehat{\sigma}}\}_{\widehat{\sigma}\in\mathfrak{S}/_\equiv}$ defined in (6.9) is both an algebra and a coalgebra boolean basis of the corresponding Hopf subalgebra of $\mathbf{FQSym}$.

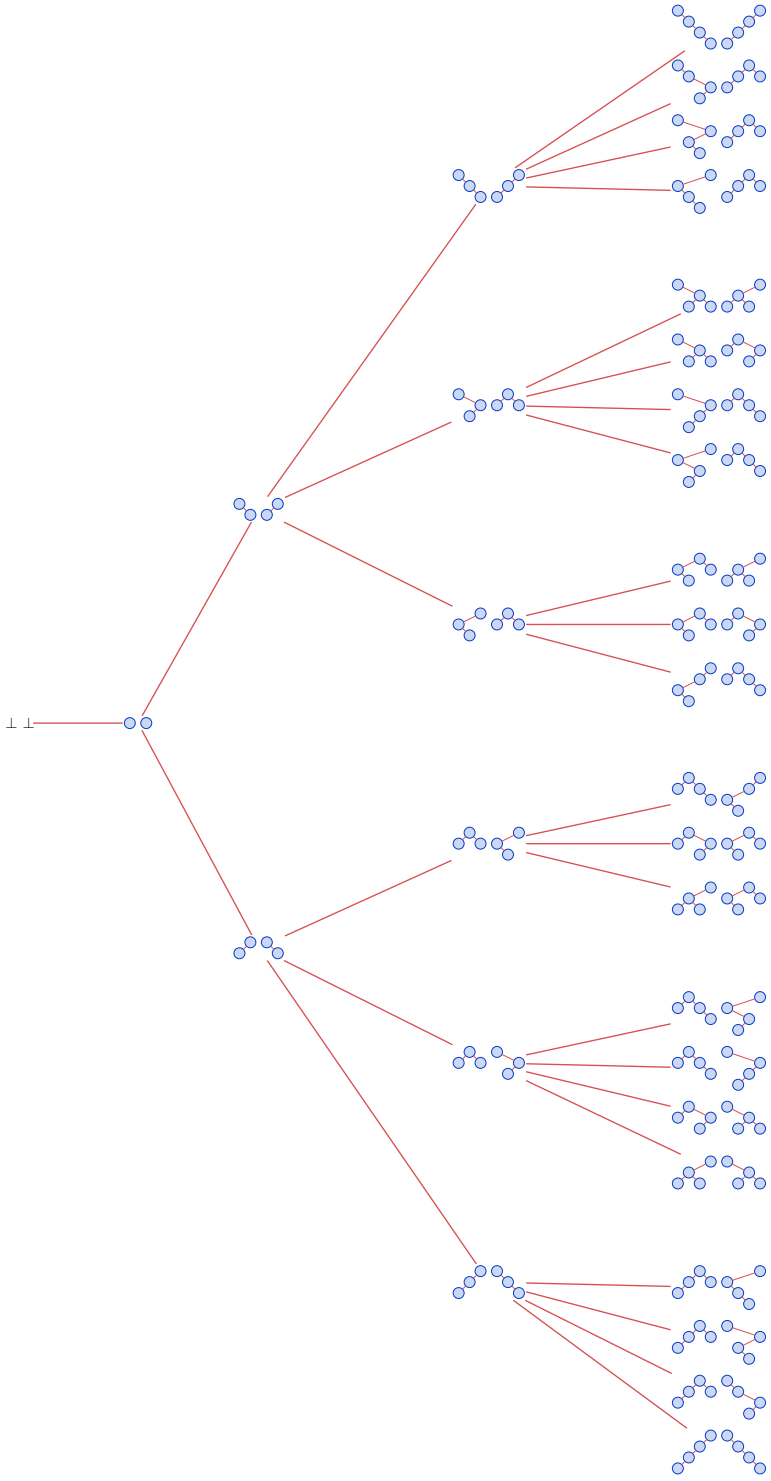


FIGURE 16. The graded graph $G_{\mathbf{P}}$ restricted to vertices of order smaller than 5.

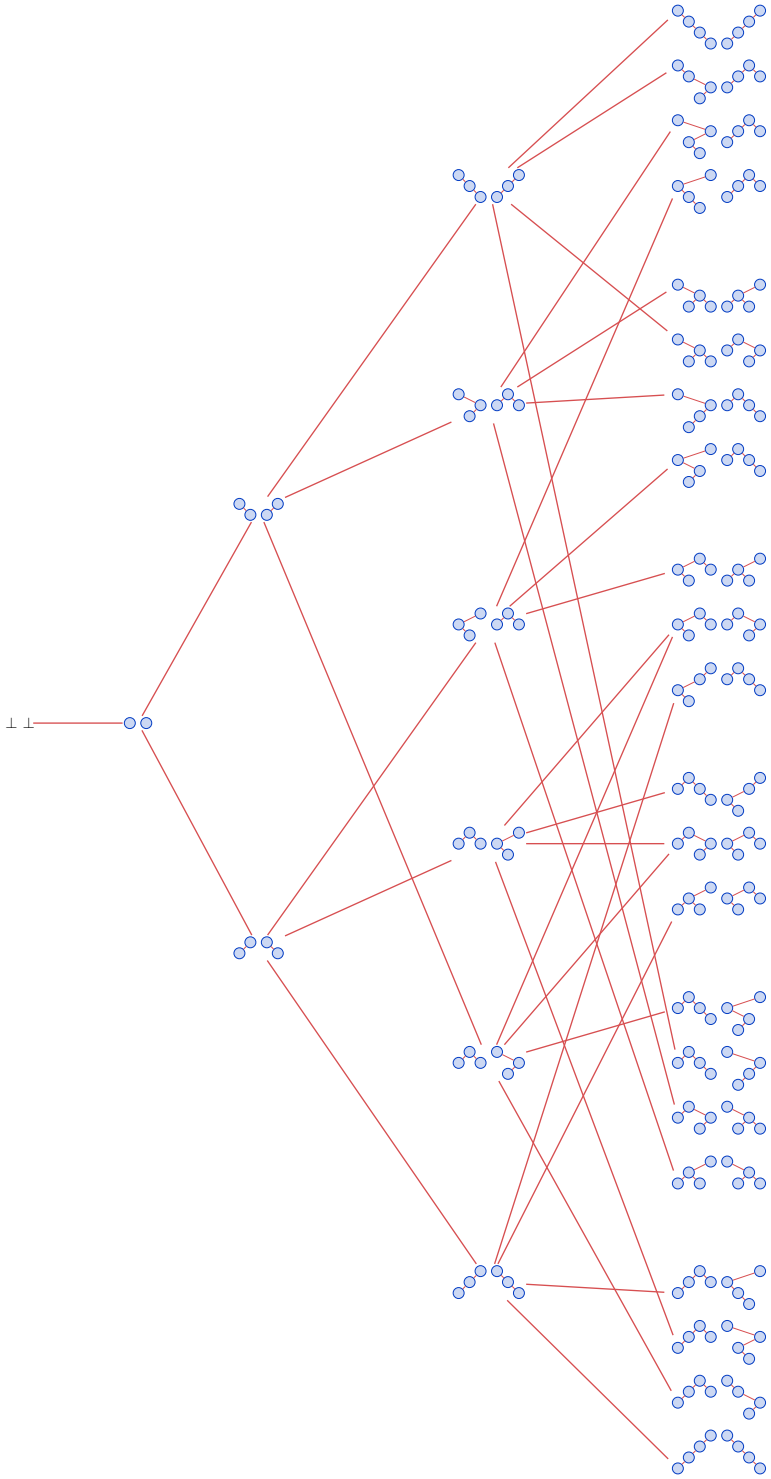


FIGURE 17. The graded graph $G_{\mathbf{P}^\star}$ restricted to vertices of order smaller than 5.

Proof. It is immediate from the definition of the product of **FQSym** that $\{\mathbf{P}_{\widehat{\sigma}}\}_{\widehat{\sigma}\in\mathfrak{S}/_{\equiv}}$ is a boolean algebra basis, regardless of (6.29).

By duality, $\{\mathbf{P}_{\widehat{\sigma}}\}_{\widehat{\sigma}\in\mathfrak{S}/_{\equiv}}$ is a boolean coalgebra basis if and only if its dual basis $\{\mathbf{P}^{\star}_{\widehat{\sigma}}\}_{\widehat{\sigma}\in\mathfrak{S}/_{\equiv}}$ is a boolean algebra basis. One has

$$\mathbf{P}^{\star}_{\widehat{\pi}}\cdot\mathbf{P}^{\star}_{\widehat{\mu}} = \phi\left(\mathbf{F}^{\star}_{\pi}\cdot\mathbf{F}^{\star}_{\mu}\right) \tag{6.30}$$

$$= \phi\left(\psi\left(\psi^{-1}\left(\mathbf{F}^{\star}_{\pi}\right)\cdot\psi^{-1}\left(\mathbf{F}^{\star}_{\mu}\right)\right)\right) \tag{6.31}$$

$$= \phi\left(\psi\left(\mathbf{F}_{\pi^{-1}}\cdot\mathbf{F}_{\mu^{-1}}\right)\right) \tag{6.32}$$

$$= \sum_{\sigma\,\in\,\pi^{-1}\,\overline{\sqcup\!\sqcup}\,\nu^{-1}} \phi\left(\mathbf{F}^{\star}_{\sigma^{-1}}\right), \tag{6.33}$$

where ϕ is the canonical projection mapping $\mathbf{F}^{\star}_{\sigma}$ on $\mathbf{P}^{\star}_{\widehat{\sigma}}$ for any permutation σ, ψ is the Hopf isomorphism mapping $\mathbf{F}_{\sigma}$ on $\mathbf{F}^{\star}_{\sigma^{-1}}$ for any permutation σ, and $\pi\in\widehat{\pi}$ and $\mu\in\widehat{\mu}$. One can easily see that if $\equiv$ satisfies the hypothesis of the proposition, then there are no multiplicities in (6.33). □

Law and Reading have proved in [LR12] that the basis of their Baxter Hopf algebra, analog to our basis $\{\mathbf{P}_J\}_{J\in\mathcal{TBT}}$, is both a boolean algebra basis and a boolean coalgebra basis. We re-prove this result in our setting:

Proposition 6.5. *The basis $\{\mathbf{P}_J\}_{J\in\mathcal{TBT}}$ is both a boolean algebra basis and a boolean coalgebra basis of* **Baxter**.

Proof. Let us prove that the sylvester equivalence relation satisfies the assumptions of Proposition 6.4. Indeed, the result directly follows from the fact that, by Proposition 3.7, the Baxter equivalence relation is finer than the sylvester equivalence relation.

Let us start with a useful result: Let x and y be two words without repetition of same length and $u,v\in x\,\overline{\sqcup\!\sqcup}\,y$ (here, the letters of y are shifted by $\max(x)$). Let us prove by induction on $|x|+|y|$ that if $\mathrm{decr}(u)$ and $\mathrm{decr}(v)$ have same shape, then $u=v$. It is obvious if $|x|+|y|=0$. Otherwise, one has $u=u'\,\mathtt{b}\,u''$ and $v=v'\,\mathtt{b}\,v''$ where $\mathtt{b}:=\max(u)=\max(v)$. Since the shape of the left subtree of $\mathrm{decr}(u)$ is equal to the shape of the left subtree of $\mathrm{decr}(v)$, the position of $\mathtt{b}$ in u and v is the same. Moreover, the word y is of the form $y=y'\,\mathtt{a}\,y''$ where $\mathtt{a}:=\max(y)$, and x is of the form $x=x'x''$, where $u',v'\in x'\,\overline{\sqcup\!\sqcup}\,y'$ and $u'',v''\in x''\,\overline{\sqcup\!\sqcup}\,y''$. Since the left (resp. right) subtree of $\mathrm{decr}(u)$ is equal to the left (resp. right) subtree of $\mathrm{decr}(v)$, by induction hypothesis, $u'=v'$ and $u''=v''$, showing that $u=v$.

Now, let $\pi,\mu\in\mathfrak{S}$ and $\sigma\neq\nu\in\pi\,\overline{\sqcup\!\sqcup}\,\mu$ and assume that $\sigma^{-1}\equiv_{\mathrm{S}}\nu^{-1}$. Then, by Theorem 3.5, the permutations σ^{-1} and ν^{-1} give the same right binary search tree when inserted from right to left. By Lemma 4.7, that implies that $\mathrm{decr}(\sigma)$ and $\mathrm{decr}(\nu)$ have same shape. That implies $\sigma=\nu$, contradicting our hypothesis. □

By duality, Proposition 6.5 also shows that the basis $\{\mathbf{P}^{\star}_J\}_{J\in\mathcal{TBT}}$ is a boolean algebra and coalgebra basis.

6.3.4. *A lattice interval description of the product.* If $\equiv$ is an equivalence relation of $\mathfrak{S}$ and σ a permutation, denote by $\widehat{\sigma}\!\uparrow$ (resp. $\widehat{\sigma}\!\downarrow$) the minimal (resp. maximal) permutation of the $\equiv$-equivalence class of σ for the permutohedron order.

Proposition 6.6. *If $\equiv$ is an equivalence relation defined on A^* satisfying the conditions of Theorem 6.1 and additionally, the $\equiv$-equivalence classes of permutations are intervals of the*

permutohedron, then the product on the family defined in (6.9) can be expressed as:

$$\mathbf{P}_{\widehat{\sigma}} \cdot \mathbf{P}_{\widehat{\nu}} = \sum_{\substack{\widehat{\sigma}\uparrow \,/\, \widehat{\nu}\uparrow \leqslant_{\mathrm{P}} \pi \leqslant_{\mathrm{P}} \widehat{\sigma}\downarrow \,\backslash\, \widehat{\nu}\downarrow \\ \pi = \min \widehat{\pi}}} \mathbf{P}_{\widehat{\pi}}. \tag{6.34}$$

Proof. It is well-known that the shifted shuffle product of two permutohedron intervals is still a permutohedron interval. Restating this fact in **FQSym**, we have

$$\left(\sum_{\sigma \leqslant_{\mathrm{P}} \mu \leqslant_{\mathrm{P}} \sigma'} \mathbf{F}_{\mu} \right) \cdot \left(\sum_{\nu \leqslant_{\mathrm{P}} \tau \leqslant_{\mathrm{P}} \nu'} \mathbf{F}_{\tau} \right) = \sum_{\sigma \,/\, \nu \leqslant_{\mathrm{P}} \pi \leqslant_{\mathrm{P}} \sigma' \,\backslash\, \nu'} \mathbf{F}_{\pi}. \tag{6.35}$$

By (6.35) and since that every $\equiv$-equivalence class is an interval of the permutohedron, we obtain

$$\mathbf{P}_{\widehat{\sigma}} \cdot \mathbf{P}_{\widehat{\nu}} = \sum_{\widehat{\sigma}\uparrow \,/\, \widehat{\nu}\uparrow \leqslant_{\mathrm{P}} \pi \leqslant_{\mathrm{P}} \widehat{\sigma}\downarrow \,\backslash\, \widehat{\nu}\downarrow} \mathbf{F}_{\pi}. \tag{6.36}$$

By Theorem 6.1, the expression (6.36) can be expressed as a sum of $\mathbf{P}_{\widehat{\pi}}$ elements and the proposition follows. □

Let $J_0 := (T_L^0, T_R^0)$ and $J_1 := (T_L^1, T_R^1)$ be two pairs of twin binary trees. Let us define the pair of twin binary trees $J_0 \,/\, J_1$ by

$$J_0 \,/\, J_1 := (T_L^0 \,\backslash\, T_L^1,\ T_R^0 \,/\, T_R^1). \tag{6.37}$$

In the same way, the pair of twin binary trees $J_0 \,\backslash\, J_1$ is defined by

$$J_0 \,\backslash\, J_1 := (T_L^0 \,/\, T_L^1,\ T_R^0 \,\backslash\, T_R^1). \tag{6.38}$$

Proposition 6.6 leads to the following expression for the product of **Baxter**.

Corollary 6.7. *For all pairs of twin binary trees J_0 and J_1, the product of* **Baxter** *satisfies*

$$\mathbf{P}_{J_0} \cdot \mathbf{P}_{J_1} = \sum_{J_0 \,/\, J_1 \leqslant_{\mathrm{B}} J \leqslant_{\mathrm{B}} J_0 \,\backslash\, J_1} \mathbf{P}_{J}. \tag{6.39}$$

Proof. Let σ and ν two permutations. It is immediate, from the definition of the P-symbol algorithm, that the P-symbol of the permutation $\sigma \,/\, \nu$ (resp. $\sigma \,\backslash\, \nu$) is the pair of twin binary trees $\mathsf{P}(\sigma) \,/\, \mathsf{P}(\nu)$ (resp. $\mathsf{P}(\sigma) \,\backslash\, \mathsf{P}(\nu)$). The expression (6.39) follows from the fact that $\equiv_{\mathrm{B}}$-equivalence classes of permutations are intervals of the permutohedron (Proposition 3.8) and from Proposition 6.6. □

6.3.5. *Multiplicative bases and free generators.* Recall that the *elementary* family $\{\mathbf{E}^{\sigma}\}_{\sigma \in \mathfrak{S}}$ and the *homogeneous* family $\{\mathbf{H}^{\sigma}\}_{\sigma \in \mathfrak{S}}$ of **FQSym** respectively defined by

$$\mathbf{E}^{\sigma} := \sum_{\sigma \leqslant_{\mathrm{P}} \sigma'} \mathbf{F}_{\sigma'}, \tag{6.40}$$

$$\mathbf{H}^{\sigma} := \sum_{\sigma' \leqslant_{\mathrm{P}} \sigma} \mathbf{F}_{\sigma'}, \tag{6.41}$$

form multiplicative bases of **FQSym** (see [AS05, DHNT11] for an exposition of some known bases of **FQSym**). Indeed, for all $\sigma, \nu \in \mathfrak{S}$, the product satisfies

$$\mathbf{E}^{\sigma} \cdot \mathbf{E}^{\nu} = \mathbf{E}^{\sigma \,/\, \nu}, \tag{6.42}$$

$$\mathbf{H}^{\sigma} \cdot \mathbf{H}^{\nu} = \mathbf{H}^{\sigma \,\backslash\, \nu}. \tag{6.43}$$

Mimicking these definitions, let us define the *elementary* family $\{\mathbf{E}_J\}_{J\in\mathcal{TBT}}$ and the *homogeneous* family $\{\mathbf{H}_J\}_{J\in\mathcal{TBT}}$ of **Baxter** respectively by

$$\mathbf{E}_J := \sum_{J \leqslant_{\mathrm{B}} J'} \mathbf{P}_{J'}, \tag{6.44}$$

$$\mathbf{H}_J := \sum_{J' \leqslant_{\mathrm{B}} J} \mathbf{P}_{J'}. \tag{6.45}$$

These families are bases of **Baxter** since they are defined by triangularity.

Proposition 6.8. *Let J be a pair of twin binary trees and $\sigma\uparrow$ (resp. $\sigma\downarrow$) be the minimal (resp. maximal) permutation such that $\mathsf{P}(\sigma\uparrow) = J$ (resp. $\mathsf{P}(\sigma\downarrow) = J$). Then,*

$$\mathbf{E}_J = \mathbf{E}^{\sigma\uparrow}, \tag{6.46}$$

$$\mathbf{H}_J = \mathbf{H}^{\sigma\downarrow}. \tag{6.47}$$

Proof. Using the fact that, by Theorem 5.1, the $\equiv_{\mathrm{B}}$-equivalence relation is a lattice congruence of the permutohedron, one successively has

$$\mathbf{E}_J = \sum_{J \leqslant_{\mathrm{B}} J'} \mathbf{P}_{J'} = \sum_{J \leqslant_{\mathrm{B}} J'} \sum_{\substack{\nu \in \mathfrak{S} \\ \mathsf{P}(\nu)=J'}} \mathbf{F}_\nu = \sum_{\substack{\nu \in \mathfrak{S} \\ J \leqslant_{\mathrm{B}} \mathsf{P}(\nu)}} \mathbf{F}_\nu = \sum_{\substack{\nu \in \mathfrak{S} \\ \sigma\uparrow \leqslant_{\mathrm{P}} \nu}} \mathbf{F}_\nu = \mathbf{E}^{\sigma\uparrow}. \tag{6.48}$$

The proof for the homogeneous family is analogous. □

Corollary 6.9. *For all pairs of twin binary trees J_0 and J_1, we have*

$$\mathbf{E}_{J_0} \cdot \mathbf{E}_{J_1} = \mathbf{E}_{J_0 \,/\, J_1}, \tag{6.49}$$

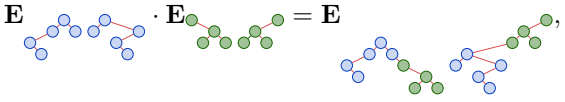
$$\mathbf{H}_{J_0} \cdot \mathbf{H}_{J_1} = \mathbf{H}_{J_0 \,\backslash\, J_1}. \tag{6.50}$$

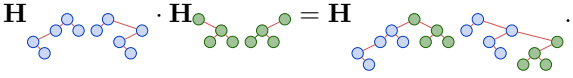
Proof. Let σ and ν be the minimal permutations of the $\equiv_{\mathrm{B}}$-equivalence classes respectively encoded by J_0 and J_1. By Proposition 6.8, we have

$$\mathbf{E}_{J_0} \cdot \mathbf{E}_{J_1} = \mathbf{E}^{\sigma} \cdot \mathbf{E}^{\nu} = \mathbf{E}^{\sigma \,/\, \nu}. \tag{6.51}$$

The permutation $\sigma \,/\, \nu$ is obviously the minimal element of its $\equiv_{\mathrm{B}}$-equivalence class, and, by the definition of the P-symbol algorithm, the P-symbol of $\sigma \,/\, \nu$ is the pair of twin binary trees $\mathsf{P}(\sigma) \,/\, \mathsf{P}(\nu) = J_0 \,/\, J_1$. The proof of the second part of the proposition is analogous. □

For example,

(6.52) 

(6.53) 

Corollary 6.9 also shows that the $\{\mathbf{E}_J\}_{J\in\mathcal{TBT}}$ and $\{\mathbf{H}_J\}_{J\in\mathcal{TBT}}$ bases of **Baxter** are boolean algebra bases. However, these are not boolean coalgebra bases since one has

$$\Delta\left(\mathbf{E}_{\bullet\bullet\bullet}\right) = 1 \otimes \mathbf{E}_{\bullet\bullet\bullet} + 2\,\mathbf{E}_{\bullet\bullet} \otimes \mathbf{E}_{\bullet\bullet} + \mathbf{E}_{\bullet\bullet\bullet} \otimes 1, \tag{6.54}$$

and

$$\Delta\left(\mathbf{H}_{\bullet\bullet\bullet}\right) = 1 \otimes \mathbf{H}_{\bullet\bullet\bullet} + 2\,\mathbf{H}_{\bullet\bullet} \otimes \mathbf{H}_{\bullet\bullet} + \mathbf{H}_{\bullet\bullet\bullet} \otimes 1. \tag{6.55}$$

Let us say that a pair of twin binary trees J is *connected* (resp. *anti-connected*) if all the permutations σ such that $\mathsf{P}(\sigma) = J$ are connected (resp. anti-connected). Since for any connected (resp. anti-connected) permutation σ and a permutation ν such that $\sigma \leqslant_{\mathrm{P}} \nu$ (resp. $\nu \leqslant_{\mathrm{P}} \sigma$) the permutation ν is also connected (resp. anti-connected), it is enough to check if the minimal (resp. maximal) permutation of the $\equiv_{\mathrm{B}}$-equivalence class encoded by J is connected (resp. anti-connected) to decide if J is connected (resp. anti-connected).

Lemma 6.10. *For any pair of twin binary trees J, there exists a sequence of connected (resp. anti-connected) pairs of twin binary trees J_1, ..., J_k such that*

$$J = J_1 \diagup \cdots \diagup J_k \quad (\text{resp. } J = J_1 \diagdown \cdots \diagdown J_k). \tag{6.56}$$

Proof. Let σ be the minimal permutation of the $\equiv_{\mathrm{B}}$-equivalence class encoded by J (recall that the existence of this element is ensured by Proposition 3.8). One can write σ as

$$\sigma = \sigma^{(1)} \diagup \cdots \diagup \sigma^{(k)}, \tag{6.57}$$

where the permutations $\sigma^{(i)}$ are connected for all $1 \leqslant i \leqslant k$. Since σ is the minimal permutation of its $\equiv_{\mathrm{B}}$-equivalence class, all the permutations $\sigma^{(i)}$ are also minimal of their $\equiv_{\mathrm{B}}$-equivalence classes. Hence, the pairs of twin binary trees $\mathsf{P}\left(\sigma^{(i)}\right)$ are connected and we can write

$$J = \mathsf{P}\left(\sigma^{(1)}\right) \diagup \cdots \diagup \mathsf{P}\left(\sigma^{(k)}\right). \tag{6.58}$$

The proof for the respective part is analogous. □

Theorem 6.11. *The algebra* **Baxter** *is free on the elements $\mathbf{E}_J$ (resp. $\mathbf{H}_J$) such that J is a connected (resp. anti-connected) pair of twin binary trees.*

Proof. By Corollary 6.9 and Lemma 6.10, each element $\mathbf{E}_J$ can be expressed as

$$\mathbf{E}_J = \mathbf{E}_{J_1} \cdot \ldots \cdot \mathbf{E}_{J_k}, \tag{6.59}$$

where the pairs of twin binary trees J_i are connected for all $1 \leqslant i \leqslant k$.

Now, since for all permutations σ and ν one has $\mathbf{E}^\sigma \cdot \mathbf{E}^\nu = \mathbf{E}^{\sigma \diagup \nu}$ in **FQSym**, and since any permutation σ admits a unique expression

$$\sigma = \sigma^{(1)} \diagup \cdots \diagup \sigma^{(k)}, \tag{6.60}$$

where $\sigma^{(1)}$, ..., $\sigma^{(k)}$ are connected permutations, there is no relation in **FQSym** between the elements $\mathbf{E}^\sigma$ where σ is a connected permutation.

Hence, by Proposition 6.8 and Corollary 6.9, there is also no relation in **Baxter** between the elements $\mathbf{E}_J$ where J is a connected pair of twin binary trees. The proof for the respective part is analogous. □

Let us denote by $B_C(z)$ the generating series of connected (resp. anti-connected) pairs of twin binary trees. It follows, from Theorem 6.11, that the Hilbert series $B(z)$ of **Baxter** satisfies $B(z) = 1/(1 - B_C(z))$. Hence, the generating series $B_C(z)$ satisfies

$$B_C(z) = 1 - \frac{1}{B(z)}. \tag{6.61}$$

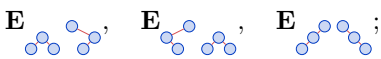
First dimensions of algebraic generators of **Baxter** are

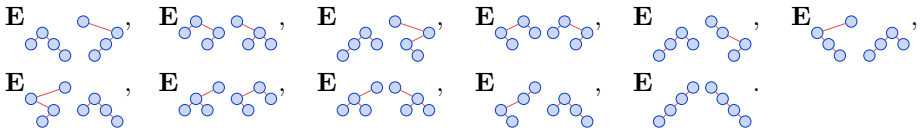
$$0, 1, 1, 3, 11, 47, 221, 1113, 5903, 32607, 186143, 1092015. \tag{6.62}$$

Here follows algebraic generators of **Baxter** of order 1 to 4:

$$\mathbf{E}_{\circ\,\circ}; \tag{6.63}$$

(6.64) 

(6.65) 

(6.66) 

Proposition 6.12. *If σ is a connected (resp. anti-connected) Baxter permutation, then any permutation ν such that $\sigma \equiv_{\mathrm{B}} \nu$ is also connected (resp. anti-connected).*

Proof. As any permutation, every Baxter permutation σ can be uniquely expressed as

$$\sigma = \sigma^{(1)} \diagup \cdots \diagup \sigma^{(k)}, \tag{6.67}$$

where the permutations $\sigma^{(i)}$ are connected for all $1 \leqslant i \leqslant k$. Moreover, since σ avoids the permutation patterns $2-41-3$ and $3-14-2$, the permutations $\sigma^{(i)}$ also does, and hence, the $\sigma^{(i)}$ are Baxter permutations. This shows that the generating series of connected Baxter permutations is $B_C(z)$ and thus, that connected Baxter permutations, connected pairs of twin binary trees, and connected minimal permutations of Baxter equivalence classes are equinumerous.

The proposition follows from Theorem 4.14 saying that each $\equiv_{\mathrm{B}}$-equivalence class of permutations contains exactly one Baxter permutation. The proof for the respective part is analogous. □

Corollary 6.13. *The algebra* **Baxter** *is free on the elements $\mathbf{E}_J$ (resp. $\mathbf{H}_J$) where the Baxter permutation belonging to the $\equiv_{\mathrm{B}}$-equivalence class encoded by J is connected (resp. anti-connected).*

6.3.6. *Bidendriform bialgebra structure and self-duality.* A Hopf algebra $(H, \cdot, \Delta)$ can be fit into a bidendriform bialgebra structure [Foi07] if $(H^+, \prec, \succ)$ is a dendriform algebra [Lod01] and $(H^+, \Delta_\prec, \Delta_\succ)$ a codendriform coalgebra, where H^+ is the augmentation ideal of H. The operators $\prec$, $\succ$, $\Delta_\prec$ and $\Delta_\succ$ have to fulfill some compatibility relations. In particular, for all $x, y \in H^+$, the product $\cdot$ of H is retrieved by $x \cdot y = x \prec y + x \succ y$ and the coproduct Δ of H is retrieved by $\Delta(x) = 1 \otimes x + \Delta_\prec(x) + \Delta_\succ(x) + x \otimes 1$. Recall that an element $x \in H^+$ is *totally primitive* if $\Delta_\prec(x) = 0 = \Delta_\succ(x)$.

The Hopf algebra **FQSym** admits a bidendriform bialgebra structure [Foi07]. Indeed, for all $\sigma, \nu \in \mathfrak{S}_n$ with $n \geqslant 1$, set

$$\mathbf{F}_\sigma \prec \mathbf{F}_\nu := \sum_{\substack{\pi \,\in\, \sigma \,\boxbar\, \nu \\ \pi_{|\pi|} = \sigma_{|\sigma|}}} \mathbf{F}_\pi, \tag{6.68}$$

$$\mathbf{F}_\sigma \succ \mathbf{F}_\nu := \sum_{\substack{\pi \,\in\, \sigma \,\boxbar\, \nu \\ \pi_{|\pi|} = \nu_{|\nu|} + |\sigma|}} \mathbf{F}_\pi, \tag{6.69}$$

$$\Delta_\prec(\mathbf{F}_\sigma) := \sum_{\substack{\sigma = uv \\ \max(u) = \max(\sigma)}} \mathbf{F}_{\mathrm{std}(u)} \otimes \mathbf{F}_{\mathrm{std}(v)}, \tag{6.70}$$

$$\Delta_{\succ}(\mathbf{F}_\sigma) := \sum_{\substack{\sigma=uv \\ \max(v)=\max(\sigma)}} \mathbf{F}_{\mathrm{std}(u)} \otimes \mathbf{F}_{\mathrm{std}(v)}. \tag{6.71}$$

Proposition 6.14. *If $\equiv$ is an equivalence relation defined on A^* satisfying the conditions of Theorem 6.1 and additionally, for all $u, v \in A^*$, the relation $u \equiv v$ implies $u_{|u|} = v_{|v|}$, then, the family defined in (6.9) spans a bidendriform sub-bialgebra of* **FQSym** *that is free as an algebra, cofree as a coalgebra, self-dual, free as a dendriform algebra on its totally primitive elements, and the Lie algebra of its primitive elements is free.*

Proof. It is enough to show that the operators $\prec$, $\succ$, $\Delta_\prec$ and $\Delta_\succ$ of **FQSym** are well-defined in the Hopf subalgebra H of **FQSym** spanned by the elements $\{\mathbf{P}_{\widehat{\sigma}}\}_{\widehat{\sigma} \in \mathfrak{S}/_\equiv}$. In this way, H is endowed with a structure of bidendriform bialgebra and the results of Foissy [Foi07] imply the rest of the proposition.

Fix $\widehat{\sigma}, \widehat{\nu} \in \mathfrak{S}/_\equiv$ and an element $\mathbf{F}_\pi$ appearing in the product $\mathbf{P}_{\widehat{\sigma}} \prec \mathbf{P}_{\widehat{\nu}}$. Hence, there is a permutation $\sigma \in \widehat{\sigma}$ such that $\pi_{|\pi|} = \sigma_{|\sigma|}$. Let π' a permutation such that $\pi \equiv \pi'$. By Theorem 6.1, the element $\mathbf{F}_{\pi'}$ appears in the product $\mathbf{P}_{\widehat{\sigma}} \cdot \mathbf{P}_{\widehat{\nu}}$, and hence, it also appears in $\mathbf{P}_{\widehat{\sigma}} \prec \mathbf{P}_{\widehat{\nu}}$ or in $\mathbf{P}_{\widehat{\sigma}} \succ \mathbf{P}_{\widehat{\nu}}$. Assume by contradiction that $\mathbf{F}_{\pi'}$ appears in $\mathbf{P}_{\widehat{\sigma}} \succ \mathbf{P}_{\widehat{\nu}}$. There are two permutations $\sigma' \in \widehat{\sigma}$ and $\nu' \in \widehat{\nu}$ such that $\pi'_{|\pi'|} = \nu'_{|\nu'|} + |\sigma'|$. That implies that $\pi_{|\pi|} \neq \pi'_{|\pi'|}$ and contradicts the fact that all permutations of a same $\equiv$-equivalence class end with a same letter. Hence, the element $\mathbf{F}_{\pi'}$ appears in $\mathbf{P}_{\widehat{\sigma}} \prec \mathbf{P}_{\widehat{\nu}}$, showing that the product $\prec$ is well-defined in H. Then so is $\succ$ since $\prec + \succ$ is the whole product.

Fix $\widehat{\sigma} \in \mathfrak{S}/_\equiv$ and an element $\mathbf{F}_\nu \otimes \mathbf{F}_\pi$ appearing in the coproduct $\Delta_\prec(\mathbf{P}_{\widehat{\sigma}})$. Hence, there is a permutation $\sigma \in \widehat{\sigma}$ such that $\sigma = uv$, $\nu = \mathrm{std}(u)$, $\pi = \mathrm{std}(v)$ and the maximal letter of uv is in the factor u. Now, let ν' and π' be two permutations such that $\nu \equiv \nu'$, $\pi \equiv \pi'$. Let us show that the element $\mathbf{F}_{\nu'} \otimes \mathbf{F}_{\pi'}$ also appears in $\Delta_\prec(\mathbf{P}_{\widehat{\sigma}})$. For that, let u' be a permutation of u such that $\mathrm{std}(u') = \nu'$, and v' be a permutation of v such that $\mathrm{std}(v') = \pi'$. Since $\mathrm{ev}(u') = \mathrm{ev}(u)$, $\mathrm{std}(u') \equiv \mathrm{std}(u)$, and $\equiv$ is compatible with the destandardization process, one has $u \equiv u'$. For the same reason, $v \equiv v'$, and since $\equiv$ is a congruence, one has $uv \equiv u'v'$. Finally, since the maximal letter of uv is in u, the maximal letter of $u'v'$ is in u', showing that the element $\mathbf{F}_{\nu'} \otimes \mathbf{F}_{\pi'}$ appears in $\Delta_\prec(\mathbf{P}_{\widehat{\sigma}})$. Thus, the coproduct $\Delta_\prec$ is well-defined in H. The proof for the coproduct $\Delta_\succ$ is analogous. □

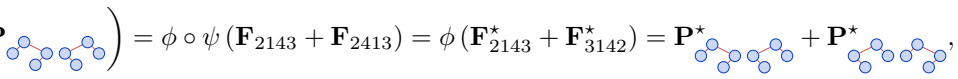
Corollary 6.15. *The Hopf algebra* **Baxter** *is free as an algebra, cofree as a coalgebra, self-dual, free as a dendriform algebra on its totally primitive elements, and the Lie algebra of its primitive elements is free.*

Proof. Since all words of a same $\equiv_{\mathrm{B}}$-equivalence class end with a same letter, $\equiv_{\mathrm{B}}$ satisfies the premises of Proposition 6.14 and hence, **Baxter** satisfies all stated properties. □

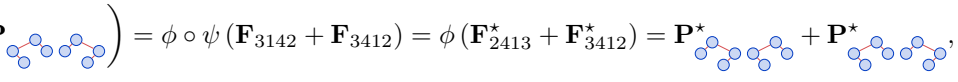
Considering the map $\theta' : \mathbf{PBT} \hookrightarrow \mathbf{FQSym}$ that is the injection from **PBT** to **FQSym** and $\phi' : \mathbf{FQSym}^\star \twoheadrightarrow \mathbf{PBT}^\star$ the surjection from $\mathbf{FQSym}^\star$ to $\mathbf{PBT}^\star$, it is well-known (see [HNT05]) that the map $\phi' \circ \psi \circ \theta'$ induces an isomorphism between **PBT** and $\mathbf{PBT}^\star$. Hence, since by Corollary 6.15, the Hopf algebras **Baxter** and $\mathbf{Baxter}^\star$ are isomorphic, it is natural to test if an analogous map is still an isomorphism between **Baxter** and $\mathbf{Baxter}^\star$. However, denoting by $\theta : \mathbf{Baxter} \hookrightarrow \mathbf{FQSym}$ the injection from **Baxter** to **FQSym**, the map $\phi \circ \psi \circ \theta : \mathbf{Baxter} \to$

Baxter* is not an isomorphism. Indeed

(6.72)



(6.73)



showing that $\phi \circ \psi \circ \theta$ is not injective.

6.3.7. *Primitive and totally primitive elements.* Since the family $\{\mathbf{E}_J\}_{J\in C}$ (resp. $\{\mathbf{H}_J\}_{J\in C}$), where C is the set of connected (resp. anti-connected) pairs of twin binary trees are indecomposable elements of **Baxter**, its dual family $\{\mathbf{E}^\star_J\}_{J\in C}$ (resp. $\{\mathbf{H}^\star_J\}_{J\in C}$) forms a basis of the Lie algebra of the primitive elements of **Baxter***. By Corollary 6.15, this Lie algebra is free.

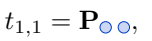
Following [Foi07], the generating series $B_T(z)$ of the totally primitive elements of **Baxter** is

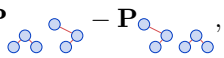
$$B_T(z) = \frac{B(z)-1}{B(z)^2}. \tag{6.74}$$

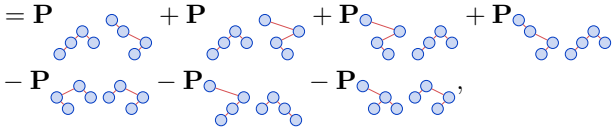
First dimensions of totally primitive elements of **Baxter** are

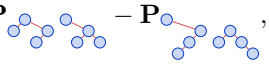
$$0, 1, 0, 1, 4, 19, 96, 511, 2832, 16215, 95374, 573837. \tag{6.75}$$

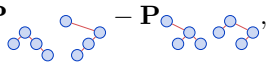
Here follows a basis of the totally primitive elements of **Baxter** of order 1, 3 and 4:

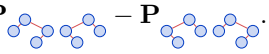
(6.76) 

(6.77) 

(6.78) 

(6.79) 

(6.80) 

(6.81) 

6.3.8. *Compatibility with the # product.* Aval and Viennot [AV10] endowed **PBT** with a new associative product called the # *product*. The product of two elements of **PBT** of degrees n and m is an element of degree $n+m-1$. Aval, Novelli, and Thibon [ANT11] generalized the # product at the level of the associative algebra and showed that it is still well-defined in **FQSym**.

Let for all $k \geqslant 1$ the linear maps $d_k : \mathbf{FQSym} \to \mathbf{FQSym}$ defined for any permutation σ of $\mathfrak{S}_n$ by

(6.82)

$$d_k(\mathbf{F}_\sigma) := \begin{cases} \mathbf{F}_{\mathrm{std}(\sigma_1\dots\sigma_i\sigma_{i+2}\dots\sigma_n)} & \text{if there is } 1 \leqslant i \leqslant n-1 \text{ such that } \sigma_i = k \text{ and } \sigma_{i+1} = k+1, \\ 0 & \text{otherwise.} \end{cases}$$

Now, for any permutations σ and ν, the #-product is defined in **FQSym** by

$$\mathbf{F}_\sigma \# \mathbf{F}_\nu := d_n\left(\mathbf{F}_\sigma \cdot \mathbf{F}_\nu\right), \tag{6.83}$$

where n is the size of σ.

Proposition 6.16. *The linear maps d_k are well-defined in* **Baxter**. *More precisely, one has for any pair of twin binary trees $J := (T_0, T_1)$,*

$$d_k(\mathbf{P}_J) = \begin{cases} \mathbf{P}_{J'} & \text{if the } k{+}1\text{-st (resp. } k\text{-th) node is a child of the} \\ & \quad k\text{-th (resp. } k{+}1\text{-st) node in } T_L \text{ (resp. } T_R), \\ 0 & \text{otherwise,} \end{cases} \tag{6.84}$$

where $J' := (T'_L, T'_R)$ is the pair of twin binary trees obtained by contracting in T_L and T_R the edges connecting the k-th and the $k{+}1$-st nodes.

Proof. This proof relies on the fact that, according to Proposition 4.10, the permutations of a Baxter equivalence class coincide with linear extensions of the posets $\triangle(T_L)$ and $\triangledown(T_R)$.

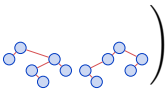
We have two cases to consider whether the $k{+}1$-st (resp. k-th) node is a child of the k-th (resp. $k{+}1$-st) node in T_L (resp. T_R).

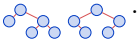
Case 1. If so, there is in the Baxter equivalence class represented by J some permutations with a factor $k.(k{+}1)$. The map d_k deletes letters $k{+}1$ in these permutations and standardizes them. The obtained permutations coincide with linear extensions of the posets $\triangle(T'_L)$ and $\triangledown(T'_R)$.

Case 2. If this is not the case, since the k-th and $k+1$-st nodes of a binary tree are on a same path starting from the root, no permutation of the Baxter class represented by J has a factor $k.(k{+}1)$. Hence, $d_k(\mathbf{P}_J) = 0$. □

One has for example

$$d_3\left(\mathbf{P}\right) = \mathbf{P} \tag{6.85}$$





Proposition 6.16 shows in particular that the # product in well-defined in **Baxter**.

6.4. Connections with other Hopf subalgebras of FQSym.

6.4.1. *Connection with the Hopf algebra* **PBT**. We already recalled that the sylvester congruence leads to the construction of the Hopf subalgebra **PBT** [LR98] of **FQSym**, whose fundamental basis

$$\{\mathbf{P}_T : T \in \mathcal{BT}\} \tag{6.86}$$

is defined in accordance with (6.9) (see [HNT02] and [HNT05]). By Proposition 3.7, every $\equiv_{\mathrm{S}}$-equivalence class is a union of some $\equiv_{\mathrm{B}}$-equivalence classes. Hence, we have the following injective Hopf map:

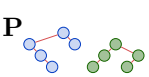
$$\rho : \mathbf{PBT} \hookrightarrow \mathbf{Baxter}, \tag{6.87}$$

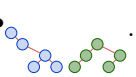
satisfying

$$\rho\left(\mathbf{P}_T\right) = \sum_{\substack{T' \in \mathcal{BT} \\ J:=(T',T) \in \mathcal{TBT}}} \mathbf{P}_J, \tag{6.88}$$

for any binary tree T. For example,

$$\rho\left(\mathbf{P}\right) = \mathbf{P} + \mathbf{P} + \mathbf{P}. \tag{6.89}$$





6.4.2. *Connection with the Hopf algebra* $\mathbf{DSym}^{(3)}$. The congruence $\equiv_{\mathrm{R}^{(3)}}$ leads to the construction of the Hopf subalgebra $\mathbf{DSym}^{(3)}$ of **FQSym**, whose fundamental basis

$$\left\{\mathbf{P}_{\widehat{\sigma}} : \widehat{\sigma} \in \mathfrak{S}/_{\equiv_{\mathrm{R}^{(3)}}}\right\} \tag{6.90}$$

is defined in accordance with (6.9) (see [NRT11]). By Proposition 3.10, every $\equiv_{\mathrm{R}^{(3)}}$-equivalence class of permutations is a union of some $\equiv_{\mathrm{B}}$-equivalence classes. Hence, we have the following injective Hopf map:

$$\alpha : \mathbf{DSym}^{(3)} \hookrightarrow \mathbf{Baxter}, \tag{6.91}$$

satisfying

$$\alpha\left(\mathbf{P}_{\widehat{\sigma}}\right) = \sum_{\sigma \in \widehat{\sigma} \cap \mathfrak{S}^{\mathrm{B}}} \mathbf{P}_{\mathsf{P}(\sigma)}, \tag{6.92}$$

for any $\equiv_{\mathrm{R}^{(3)}}$-equivalence class $\widehat{\sigma}$ of permutations.

6.4.3. *Connection with the Hopf algebra* **Sym**. The hypoplactic congruence [Nov98] leads to the construction of the Hopf subalgebra **Sym** of **FQSym**. As already mentioned, the hypoplactic congruence is the same as the congruence $\equiv_{\mathrm{R}^{(2)}}$ when both are restricted on permutations. Moreover, the hypoplactic equivalence classes of permutations can be encoded by binary words. Indeed, if $\widehat{\sigma}$ is such an equivalence class, $\widehat{\sigma}$ contains all the permutations having a given recoil set. Thus, the class $\widehat{\sigma}$ can be encoded by the binary word b of length $n-1$ where n is the length of the elements of $\widehat{\sigma}$ and $b_i = 1$ if and only if i is a recoil of the elements of $\widehat{\sigma}$. We denote by

$$\{\mathbf{P}_b : b \in \{0,1\}^*\} \tag{6.93}$$

the fundamental basis of **Sym** indexed by binary words.

Since **PBT** is a Hopf subalgebra of **Baxter** and **Sym** is a Hopf subalgebra of **PBT** [HNT05], **Sym** is itself a Hopf subalgebra of **Baxter**. The injective Hopf map

$$\beta : \mathbf{Sym} \hookrightarrow \mathbf{PBT}, \tag{6.94}$$

satisfies, thanks to the fact that the hypoplactic equivalence classes are union of $\equiv_{\mathrm{S}}$-equivalence classes and Proposition 4.4,

$$\beta\left(\mathbf{P}_b\right) = \sum_{\substack{T \in \mathcal{BT} \\ \mathrm{cnp}(T)=b}} \mathbf{P}_T, \tag{6.95}$$

for any binary word b. From a combinatorial point of view, given a binary word b, the map β computes the sum of the binary trees having b as canopy. The composition $\rho \circ \beta$ is an injective Hopf map from **Sym** to **Baxter**. From a combinatorial point of view, given a binary word b, the map $\rho \circ \beta$ computes the sum of the pairs of twin binary trees (T_L, T_R) where the canopy of T_R is b and the canopy of T_L is the complementary of b.

6.4.4. *Full diagram of embeddings.* Figure 18 summarizes the relations between known Hopf algebras related to **Baxter**.

References

[ABP04] E. Ackerman, G. Barequet, and R. Y. Pinter. On the Number of Rectangular Partitions. *Proc. 15th ACM-SIAM Symp. on Discrete Algorithms*, pages 736–745, 2004.

[ANT11] J.-C. Aval, J.-C. Novelli, and J.-Y. Thibon. The # product in combinatorial Hopf algebras. `arXiv:1007.1901v2 [math.CO]`, 2011.

[AS05] M. Aguiar and F. Sottile. Structure of the Malvenuto-Reutenauer Hopf algebra of permutations. *Adv. Math.*, 191(2):225–275, 2005.

[AU94] A. Aho and J. Ullman. *Foundations of Computer Science*. W. H. Freeman, 1994.

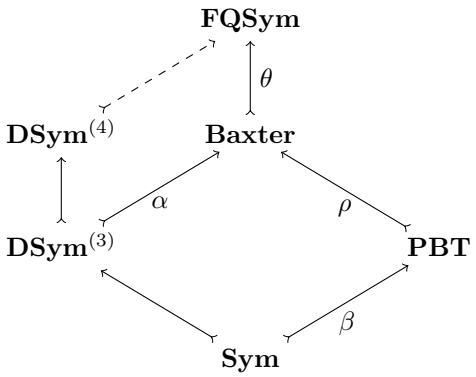


FIGURE 18. Diagram of injective Hopf maps between some Hopf algebras related to **Baxter**. Arrows $\rightarrowtail$ are injective Hopf maps.

[AV10] J.-C. Aval and X. Viennot. The product of trees in the Loday-Ronco algebra through Catalan alternative tableaux. *Sém. Lothar. Combin.*, 63, 2010.

[Bax64] G. Baxter. On fixed points of the composite of commuting functions. *Proceedings of the American Mathematical Society*, 15:851–855, 1964.

[BBMF08] N. Bonichon, M. Bousquet-Mélou, and É. Fusy. Baxter permutations and plane bipolar orientations. *Electronic Notes in Discrete Mathematics*, 31:69–74, 2008.

[BLL08] N. Bergeron, T. Lam, and H. Li. Combinatorial Hopf Algebras and Towers of Algebras. *Formal Power Series and Algebraic Combinatorics*, 2008.

[BM03] M. Bousquet-Mélou. Four classes of pattern-avoiding permutations under one roof: generating trees with two labels. *The electronic journal of combinatorics*, 9(2), 2003.

[BS00] E. Babson and E. Steingrímsson. Generalized permutation patterns and a classification of the Mahonian statistic. *Sém. Lothar. Combin.*, 44, 2000.

[CS98] I. Chajda and V. Snášel. Congruences in Ordered Sets. *Mathematica Bohemica*, 123(1):95–100, 1998.

[DG94] S. Dulucq and O. Guibert. Mots de piles, tableaux standards et permutations de Baxter. *Formal Power Series and Algebraic Combinatorics*, 1994.

[DHNT11] G. Duchamp, F. Hivert, J.-C. Novelli, and J.-Y. Thibon. Noncommutative symmetric functions VII: free quasi-symmetric functions revisited. *Ann. Comb.*, 15:655–673, 2011.

[DHT02] G. Duchamp, F. Hivert, and J.-Y. Thibon. Noncommutative Symmetric Functions VI: Free Quasi-Symmetric Functions and Related Algebras. *Int. J. Algebr. Comput.*, 12(5):671–717, 2002.

[Foi07] L. Foissy. Bidendriform bialgebras, trees, and free quasi-symmetric functions. *J. Pure Appl. Algebra*, 209(2):439–459, 2007.

[Fom94] S. Fomin. Duality of Graded Graphs. *J. Algebr. Comb.*, 3(4):357–404, 1994.

[Gir11] S. Giraudo. Algebraic and combinatorial structures on Baxter permutations. *Formal Power Series and Algebraic Combinatorics*, 23:387–398, 2011.

[GKL+94] I.M. Gelfand, D. Krob, A. Lascoux, B. Leclerc, V.S. Retakh, and J.-Y. Thibon. Noncommutative symmetric functions I. `arXiv:hep-th/9407124v1`, 1994.

[Hiv07] F. Hivert. An introduction to Combinatorial Hopf Algebras—examples and realizations. *IOS Press*, 7:253–274, 2007.

[HN07] F. Hivert and J. Nzeutchap. Dual graded graphs in combinatorial Hopf algebras. *unpublished*, 2007.

[HNT02] F. Hivert, J.-C. Novelli, and J.-Y. Thibon. An analogue of the plactic monoid for binary search trees. *C. R. Math.*, 335(7):577–580, 2002.

[HNT05] F. Hivert, J.-C. Novelli, and J.-Y. Thibon. The Algebra of Binary Search Trees. *Theor. Comput. Sci.*, 339(1):129–165, 2005.

[Knu06] D. Knuth. *The art of computer programming. Vol. 4, Fasc. 4*. Addison-Wesley, 2006. Generating all trees—history of combinatorial generation.

[KT97] D. Krob and J.-Y. Thibon. Noncommutative symmetric functions IV: Quantum linear groups and Hecke algebras at $q = 0$. *J. Algebr. Comb.*, 6(4):339–376, 1997.

[Lod01] J.-L. Loday. Dialgebras. *Lect. Notes Math.*, 1763:7–66, 2001.

[Lot02] M. Lothaire. *Algebraic combinatorics on words*. Cambridge University Press, 2002.

[LR98] J.-L. Loday and M. O. Ronco. Hopf Algebra of the Planar Binary Trees. *Advances in Mathematics*, 139:293–309, 1998.

[LR02] J.-L. Loday and M. O. Ronco. Order Structure on the Algebra of Permutations and of Planar Binary Trees. *J. Algebr. Comb.*, 15(3):253–270, 2002.

[LR12] S. Law and N. Reading. The Hopf algebra of diagonal rectangulations. *J. Combin. Theory Ser. A*, 119(3):788–824, 2012.

[LS81] A. Lascoux and M.-P. Schützenberger. Le monoïde plaxique. *Noncommutative Structures in Algebra and Geometric Combinatorics*, 109:129–156, 1981.

[MR95] C. Malvenuto and C. Reutenauer. Duality between quasi-symmetric functions and Solomon descent algebra. *J. Algebra*, 177:967–982, 1995.

[Nov98] J.-C. Novelli. On the hypoplactic monoid. *Discrete Math.*, 217(1-3):315–336, 1998.

[NRT11] J.-C. Novelli, C. Reutenauer, and J.-Y. Thibon. Generalized descent patterns in permutations and associated Hopf algebras. *European J. Combin.*, 32(4):618–627, 2011.

[Pal86] J. M. Pallo. Enumerating, Ranking and Unranking Binary Trees. *Computer Journal*, 29(2):171–175, 1986.

[PR95] S. Poirier and C. Reutenauer. Algèbres de Hopf de tableaux. *Ann. Sci. Math. Québec*, 19(1):79–90, 1995.

[Rea05] N. Reading. Lattice congruences, fans and Hopf algebras. *Journal of Combinatorial Theory Series A*, 110(2):237–273, 2005.

[Rey07] M. Rey. Algebraic constructions on set partitions. *Formal Power Series and Algebraic Combinatorics*, 2007.

[S$^+$11] W. A. Stein et al. *Sage Mathematics Software (Version 4.7.2)*. The Sage Development Team, 2011. http://www.sagemath.org.

[Slo] N. J. A. Sloane. The On-Line Encyclopedia of Integer Sequences. http://www.research.att.com/~njas/sequences/.

[Tam62] D. Tamari. The algebra of bracketings and their enumeration. *Nieuw Arch. Wisk. (3)*, 10:131–146, 1962.

[Vie04] X. Viennot. *Up-down sequences of permutations, paths and canopy of binary trees*, 2004. invited talk at the Lascouxfest, 52nd SLC, Otrott.

Institut Gaspard-Monge, université Paris-Est Marne-la-Vallée, 5 Boulevard Descartes, Champs-sur-Marne, 77454, Marne-la-Vallée cedex 2, France

E-mail address: samuele.giraudo@univ-mlv.fr