ALGEBRAIC AND COMBINATORIAL STRUCTURES ON BAXTER PERMUTATIONS

SAMUELE GIRAUDO

Abstract. We construct a Hopf subalgebra of the Hopf algebra of Free quasi-symmetric functions whose bases are indexed by objects belonging to the Baxter combinatorial family (*i.e.*, Baxter permutation, pairs of twin binary trees, *etc.*). This construction relies on the definition of the Baxter monoid, analog of the plactic monoid and the sylvester monoid, and on a Robinson-Schensted-like insertion algorithm. The algebraic properties of this Hopf algebra are studied.

Contents

Date: January 4, 2011.
Key words and phrases. Hopf algebras, Robinson-Schensted algorithm, quotient monoid, Baxter permutations.

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. An 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], **Bell** from the Bell monoid [Rey07], **PBT** from the sylvester monoid [HNT02, HNT05], and **Sym** from the hypoplactic monoid [Nov98]. The richness of these constructions relies on the fact that, in addition to construct Hopf algebras, the definition of such monoids often brings partial orders, combinatorial algorithms and Robinson-Schensted-like algorithms, of independent interest.

In this paper, we propose to enrich this collection of Hopf algebras by providing a construction of a Hopf algebra whose bases are indexed by objects belonging to the Baxter combinatorial family. This combinatorial family admits various representations as Baxter permutations [Bax64], pairs of twin binary trees [DG94], quadrangulations [ABP04], plane bipolar orientations [BBMF08], *etc.* In [Rea05], Reading defines, as an example, a Hopf Algebra of twisted Baxter permutations in the context of lattice congruences [CS98]. He claims that twisted Baxter permutations are equinumerous with Baxter permutations up to order 15. Law and Reading point out in [LR10] that the first proof that Baxter permutations and twisted Baxter permutations are equinumerous occurs in unpublished notes of West and use generating trees [BM03]. Hence, the Reading's Hopf algebra can already be seen as a Hopf algebra on Baxter permutations.

Very recently, Law and Reading [LR10] have studied and detailed the construction of this Hopf algebra. Their starting point is very natural: It is well-known that the meet of two lattice congruences of the permutohedron related to the construction of **PBT** can be considered as the starting point of the construction of **Sym**. A natural question is to know what happens when the join, instead of the meet, of these two lattice congruences is considered. The minimal elements of the equivalence classes of the resulting lattice congruence are twisted Baxter permutations. We started independently the study of Baxter objects in a different way: We looked for a quotient of the free monoid analog to the plactic and the sylvester monoid. Surprisingly, the equivalence classes of the permutations under our monoid congruence are the same as the equivalence classes of Law and Reading's lattice congruence, 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 not 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 algorithm, the bidendriform bialgebra structure, the freeness, cofreeness, self-duality, primitive elements, and multiplicative bases of the Hopf algebra, and a few other properties.

We begin by recalling in Section 2 the preliminary notions 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 close to the Tamari lattice [Tam62, Knu06] since covering relations can be expressed by binary tree rotations. Finally, in Section 6, we define the Hopf algebra **Baxter** and study it. Using the order structure on pairs of twin binary trees, we provide multiplicative bases and show that **Baxter** is free as an algebra. Using the results of Foissy on bidendriform bialgebras [Foi05], we show that **Baxter** is also self-dual and that the Lie algebra of its primitive elements is free.

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$^+$10].

2. Preliminaries

2.1. **Words.** In the sequel, $A := \{a_1 < a_2 < \dots\}$ is a totally ordered infinite alphabet and A^* is the free monoid spanned by A. Let $u \in A^*$. We shall denote by $|u|$ the length of u and by ϵ the word of length 0. Let us denote by $u^\sim := u_{|u|} \dots u_1$ the *mirror image* of u. Denote by $\mathrm{Alph}(u) := \{u_i : 1 \leq i \leq |u|\}$ the smallest alphabet on which u is defined. Besides, for $S \subseteq A$, we denote by $u_{|S}$ the *restriction* of u on the alphabet S, that is the longest subword of u such that $\mathrm{Alph}(u) \subseteq S$. The *evaluation* $\mathrm{eval}(u)$ 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. We say that (i, j) is an *inversion* of u if $i < j$ and $u_i > u_j$. Besides, i is *descent* of u if $(i, i+1)$ is an inversion of u. Let $v \in A^*$. The *shuffle product* $\sqcup\!\sqcup$ is defined on $\mathbb{Z}\langle A^* \rangle$ recursively by:

$$(1) \qquad u \sqcup\!\sqcup v := \begin{cases} u & \text{if } v = \epsilon, \\ v & \text{if } u = \epsilon, \\ \mathtt{a}(u' \sqcup\!\sqcup \mathtt{b}v') \ + \ \mathtt{b}(\mathtt{a}u' \sqcup\!\sqcup v') & \text{where } u = \mathtt{a}u' \text{ and } v = \mathtt{b}v', \text{ otherwise.} \end{cases}$$

For example,

$$(2) \qquad \begin{aligned} a_1a_2 \sqcup\!\sqcup \mathbf{a_2a_1} &= a_1a_2\mathbf{a_2a_1} + a_1\mathbf{a_2}a_2\mathbf{a_1} + a_1\mathbf{a_2a_1}a_2 + \mathbf{a_2}a_1a_2\mathbf{a_1} + \mathbf{a_2}a_1\mathbf{a_1}a_2 + \mathbf{a_2a_1}a_1a_2 \\ &= a_1a_2a_1a_2 + 2a_1a_2a_2a_1 + 2a_2a_1a_1a_2 + a_2a_1a_2a_1. \end{aligned}$$

Let $A^\# := \{a_1^\# > a_2^\# > \dots\}$ be the alphabet A on which the order relations between its letters have been reversed. The *Schützenberger transformation* $\#$ is defined on words by $u^\# := u_{|u|}^\# \dots u_1^\#$; For example, $(a_5a_3a_1a_1a_5a_2)^\# = a_2^\# a_5^\# a_1^\# a_1^\# a_3^\# a_5^\#$. By setting $A^{\#^\#} := A$, $\#$ is also an involution on words.

2.2. **Permutations.** Denote by $\mathfrak{S}_n$ the set of permutations of size n and $\mathfrak{S} := \cup_{n \geq 0} \mathfrak{S}_n$. We shall call (i, j) a *co-inversion* of $\sigma \in \mathfrak{S}$ if (i, j) is an inversion of σ^{-1}. Besides, i is a *recoil* of σ if $(i, i+1)$ is a co-inversion of σ. Let us recall that the *(right) permutohedron order* is the partial order $\leq_{\mathrm{P}}$ defined on $\mathfrak{S}_n$ where σ is covered by ν if $\sigma = u\mathtt{ab}v$ and $\nu = u\mathtt{ba}v$ where $\mathtt{a} < \mathtt{b}$. 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, $\mathbf{312} \diagup 2314 = \mathbf{312}5647$ and $\mathbf{312} \diagdown 2314 = 5647\mathbf{312}$. The permutation σ is *connected* if $\sigma = \nu \diagup \pi$ implies $\nu = \sigma$ or $\pi = \sigma$. The permutation σ is *anti-connected* if $\sigma^{\sim}$ is connected. The *shifted shuffle product* $\overline{\sqcup\!\sqcup}$ of two permutations is defined by:

$$\sigma \,\overline{\sqcup\!\sqcup}\, \nu := \sigma \sqcup\!\sqcup \left(\nu_1 + |\sigma| \dots \nu_{|\nu|} + |\sigma|\right). \tag{3}$$

For example,

$$\mathbf{12} \,\overline{\sqcup\!\sqcup}\, 21 = \mathbf{12} \sqcup\!\sqcup 43 = \mathbf{12}43 + \mathbf{1}4\mathbf{2}3 + \mathbf{1}43\mathbf{2} + 4\mathbf{12}3 + 4\mathbf{1}3\mathbf{2} + 43\mathbf{12}. \tag{4}$$

The *standardized word* $\mathrm{std}(u)$ of $u \in A^*$ is the unique permutation having the same inversions as u; For example, $\mathrm{std}(a_3a_1a_4a_2a_5a_7a_4a_2a_3) = 416289735$.

2.3. **Binary trees.** Denote by $\mathcal{BT}_n$ the set of binary trees with n internal nodes and $\mathcal{BT} := \cup_{n\geq 0}\mathcal{BT}_n$. We use in the sequel the standard terminology (*i.e.*, *child*, *ancestor*, *arc*, *etc.*) about binary trees [AU94]. The only element of $\mathcal{BT}_0$ is the *leaf* or *empty tree*, denoted by $\perp$. We also shall call a *leaf* a node with empty left and right subtrees. Let us recall that the *Tamari order* [Tam62, Knu06] is the partial order $\leq_{\mathrm{T}}$ 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 transform T_0 into T_1 by performing a right rotation (see Figure 1).

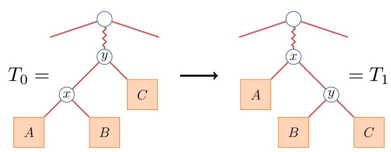
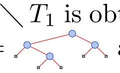
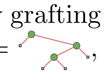
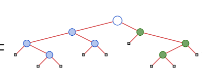
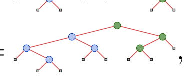
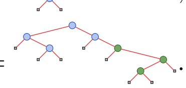


FIGURE 1. The right rotation around the arc $y \to x$.

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 $T_0 \diagup T_1$ is obtained by grafting T_0 from its root on the leftmost leaf of T_1; In the same way, the binary tree $T_0 \diagdown T_1$ is obtained by grafting T_1 from its root on the rightmost leaf of T_0. For example, for $T_0 :=$  and $T_1 :=$ , we have

(5) $T_0 \wedge T_1 =$ ,

(6) $T_0 \diagup T_1 =$ ,

(7) $T_0 \diagdown T_1 =$ .

Viennot defined in [Vie04] the *canopy* $\mathrm{cnp}(T)$ of $T \in \mathcal{BT}$, that 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 2). 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.

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

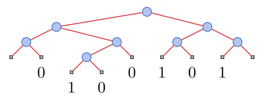


FIGURE 2. The canopy of this binary tree is 0100101.

of x. The *shape* of a labeled binary tree is the unlabeled binary tree obtained by forgetting its labels. Recall that the *infix traversal* of a binary tree T consists in recursively traversing the left subtree of T, then its root, and finally recursively traversing its right subtree. If T is labeled, its *infix reading* is the word $u_1 u_2 \dots u_{|u|}$ such that u_1 is the label of the first visited node by the infix traversal of T, u_2 the second, ..., and $u_{|u|}$ the last one.

2.4. **Baxter permutations and pairs of twin binary trees.** A permutation σ is a *Baxter permutation* if for any subword $u := u_1 u_2 u_3 u_4$ 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, **4**2**173**8**5**6 is not a Baxter permutation; On the other hand 436975128 is a Baxter permutation. Let us denote by $\mathfrak{S}^{\mathrm{B}}_n$ the set of Baxter permutations of size n and $\mathfrak{S}^{\mathrm{B}} := \cup_{n \geq 0} \mathfrak{S}^{\mathrm{B}}_n$.

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 $\mathrm{cnp}(T_L)_i \neq \mathrm{cnp}(T_R)_i$ for all $1 \leq i \leq n-1$ (see Figure 3). Denote by $\mathcal{TBT}_n$ the set of pairs of twin binary trees where each binary tree has n nodes and $\mathcal{TBT} := \cup_{n \geq 0} \mathcal{TBT}_n$. 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. The *shape* of an A-labeled pair of twin binary trees (T_L, T_R) is the unlabeled pair of twin binary trees (T'_L, T'_R) such that T'_L (resp. T'_R) is the shape of T_L (resp. T_R). In [DG94], Dulucq

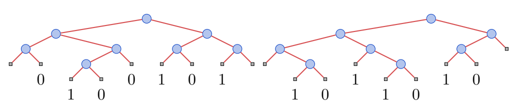


FIGURE 3. A pair of twin binary trees.

and Guibert have highlighted a bijection between Baxter permutations and pairs of unlabeled 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 $uv \equiv u'v'$.

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

$$\mathtt{c}u\mathtt{ad}v\mathtt{b} \leftrightharpoons_{\mathrm{B}} \mathtt{c}u\mathtt{da}v\mathtt{b} \qquad \text{where} \qquad \mathtt{a} \leq \mathtt{b} < \mathtt{c} \leq \mathtt{d}, \tag{8}$$

$$\mathtt{b}u\mathtt{da}v\mathtt{c} \rightleftharpoons_{\mathrm{B}} \mathtt{b}u\mathtt{ad}v\mathtt{c} \qquad \text{where} \qquad \mathtt{a} < \mathtt{b} \leq \mathtt{c} < \mathtt{d}. \tag{9}$$

For example, the $\equiv_{\mathrm{B}}$-equivalence class of 2114424 (see Figure 4) is

$$\{2114424,\ 2141424,\ 2144124, 2411424,\ 2414124,\ 2441124\}. \tag{10}$$

FIGURE 4. The Baxter equivalence class of the word $u := 2114424$ and of the permutation $3125647 = \mathrm{std}(u)$, with their adjacency relations.

If the Baxter congruence is applied on words without repetition, the two Baxter adjacency relations $\leftrightharpoons_{\mathrm{B}}$ and $\rightleftharpoons_{\mathrm{B}}$ can be replaced by the only adjacency relation $\rightleftarrows_{\mathrm{B}}$ defined by:

$$\mathtt{x}u\mathtt{ad}v\mathtt{y} \rightleftarrows_{\mathrm{B}} \mathtt{x}u\mathtt{da}v\mathtt{y} \qquad \text{where} \quad \mathtt{a} < \mathtt{x}, \mathtt{y} < \mathtt{d}. \tag{11}$$

An equivalence relation $\equiv$ defined on A^* is *compatible with the destandardization process* if for all $u, v \in A^*$, $u \equiv v$ iff $\mathrm{std}(u) \equiv \mathrm{std}(v)$ and $\mathrm{eval}(u) = \mathrm{eval}(v)$.

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 $u \leftrightharpoons_{\mathrm{B}} v$. We have $u = x\mathtt{c}y\mathtt{ad}z\mathtt{b}t$ and $v = x\mathtt{c}y\mathtt{da}z\mathtt{b}t$ for some $x, y, z, t \in A^*$ and $\mathtt{a} \leq \mathtt{b} < \mathtt{c} \leq \mathtt{d} \in A$. Since $\leftrightharpoons_{\mathrm{B}}$ acts by permuting letters, we have $\mathrm{eval}(u) = \mathrm{eval}(v)$. Moreover, the letters $\mathtt{a}'$, $\mathtt{b}'$, $\mathtt{c}'$ and $\mathtt{d}'$ of $\mathrm{std}(u)$ respectively at the same positions than 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 $\mathrm{std}(u) \leftrightharpoons_{\mathrm{B}} \mathrm{std}(v)$. The proof is analogous for the case $u \rightleftharpoons_{\mathrm{B}} v$.

Conversely, assume that v is a permutation of u and $\mathrm{std}(u) \leftrightharpoons_{\mathrm{B}} \mathrm{std}(v)$. We have $\mathrm{std}(u) = x\mathtt{c}y\mathtt{ad}z\mathtt{b}t$ and $\mathrm{std}(v) = x\mathtt{c}y\mathtt{da}z\mathtt{b}t$ for some $x, y, z, t \in A^*$ and $\mathtt{a} < \mathtt{b} < \mathtt{c} < \mathtt{d} \in A$. 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 than the letters $\mathtt{a}$, $\mathtt{b}$, $\mathtt{c}$ and $\mathtt{d}$ of $\mathrm{std}(u)$ satisfy $\mathtt{a}' \leq \mathtt{b}' < \mathtt{c}' \leq \mathtt{d}'$ 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 $u \leftrightharpoons_{\mathrm{B}} v$. The proof is analogous for the case $\mathrm{std}(u) \rightleftharpoons_{\mathrm{B}} \mathrm{std}(v)$. □

An equivalence relation $\equiv$ defined on A^* is *compatible with the restriction of alphabet intervals* if for all 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 $\sigma = t\mathtt{x}u\mathtt{ad}v\mathtt{y}w$ and $\nu = t\mathtt{x}u\mathtt{da}v\mathtt{y}w$ for some letters $\mathtt{a} < \mathtt{x}, \mathtt{y} < \mathtt{d}$. Let I be an interval of $\{1, \ldots, n\}$

and $R := I \cap \{\mathtt{a}, \mathtt{x}, \mathtt{y}, \mathtt{d}\}$. If $R = \{\mathtt{a}, \mathtt{x}, \mathtt{y}, \mathtt{d}\}$, $\sigma_{|I} = t_{|I}\mathtt{x}u_{|I}\mathtt{ad}v_{|I}\mathtt{y}w_{|I}$ and $\nu_{|I} = t_{|I}\mathtt{x}u_{|I}\mathtt{da}v_{|I}\mathtt{y}w_{|I}$ so that $\sigma_{|I} \rightleftarrows_{\mathrm{B}} \nu_{|I}$. Else, we have $\sigma_{|I} = \nu_{|I}$ and thus $\sigma_{|I} \equiv_{\mathrm{B}} \nu_{|I}$. □

An equivalence relation $\equiv$ defined on A^* 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 $\sigma = t\mathtt{x}u\mathtt{ad}v\mathtt{y}w$ and $\nu = t\mathtt{x}u\mathtt{da}v\mathtt{y}w$ for some letters $\mathtt{a} < \mathtt{x}, \mathtt{y} < \mathtt{d}$. We have $\sigma^\# = w^\#\mathtt{y}^\#v^\#\mathtt{d}^\#\mathtt{a}^\#u^\#\mathtt{x}^\#t^\#$ and $\nu^\# = w^\#\mathtt{y}^\#v^\#\mathtt{a}^\#\mathtt{d}^\#u^\#\mathtt{x}^\#t^\#$. Since $\mathtt{d}^\# < \mathtt{x}^\#, \mathtt{y}^\# < \mathtt{a}^\#$, 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 transitive closure of the *sylvester adjacency relation* $\leftrightharpoons_{\mathrm{S}}$ defined for $u \in A^*$ and $\mathtt{a}, \mathtt{b}, \mathtt{c} \in A$ by:

$$\mathtt{ac}u\mathtt{b} \leftrightharpoons_{\mathrm{S}} \mathtt{ca}u\mathtt{b} \qquad \text{where} \quad \mathtt{a} \leq \mathtt{b} < \mathtt{c}. \tag{12}$$

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

$$\mathtt{b}u\mathtt{ca} \leftrightharpoons_{\mathrm{S}^\#} \mathtt{b}u\mathtt{ac} \qquad \text{where} \quad \mathtt{a} < \mathtt{b} \leq \mathtt{c}. \tag{13}$$

It is plain, for all $u, v \in A^*$ that we have $u \equiv_{\mathrm{S}} v$ iff $u^\# \equiv_{\mathrm{S}^\#} v^\#$.

In [HNT05], Hivert, Novelli and Thibon have shown that two words are sylvester equivalent iff each of these gives the same right binary search tree by inserting their letters from right to left using the well-known binary search tree insertion algorithm [AU94]. In our setting, we call this process the *leaf insertion* and it declines 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, 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 $\mathtt{a} \leq \mathtt{b}$ (resp. $\mathtt{a} < \mathtt{b}$):
    (a) Then, recursively leaf insert $\mathtt{a}$ into the left subtree of T.
    (b) Else, 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:

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

In other words, the A-labeled right (resp. left) binary search trees encode the sylvester (resp. #-sylvester) equivalence classes of words of A^*. Note that the difference of treatment between right and left binary search trees in the instruction (3) of LeafInsertion comes from the effect of the Schützenberger involution on words and the fact that two words are $\equiv_{\mathrm{S}^\#}$-equivalent iff their image under the Schützenberger involution are $\equiv_{\mathrm{S}}$-equivalent. Indeed, if u is a word and T is the right binary search tree obtained by inserting the letters of u from right to left, the left binary search tree $T^{\#}$ obtained by inserting the word $u^{\#}$ from left to right is equals to T after swapping for each node its left and right subtrees and applying the Schützenberger involution on its labels.

Lemma 3.6. *Let $u = x\mathtt{ac}y$ and $v = x\mathtt{ca}y$ be two words such that $x, y \in A^*$, $\mathtt{a} < \mathtt{c} \in A$ 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 inserting them 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 the nodes labeled by $\mathtt{a}$ and $\mathtt{c}$ such that $\mathtt{a} \leq \mathtt{b} < \mathtt{c}$. Thus, $u \leftrightharpoons_{\mathrm{S}} v$. □

Lemma 3.6 proves also 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,*

$$u \equiv_{\mathrm{B}} v \Longleftrightarrow u \equiv_{\mathrm{S}} v \text{ and } u \equiv_{\mathrm{S}^{\#}} v. \tag{14}$$

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 $\sigma = x\mathtt{x}y\mathtt{ad}z\mathtt{y}t$ and $\nu = x\mathtt{x}y\mathtt{da}z\mathtt{y}t$ for some letters $\mathtt{a} < \mathtt{x}, \mathtt{y} < \mathtt{d}$. The presence of the letters $\mathtt{a}$, $\mathtt{d}$ and $\mathtt{y}$ with $\mathtt{a} < \mathtt{y} < \mathtt{d}$ ensures that $\sigma \leftrightharpoons_{\mathrm{S}} \nu$. Besides, the presence of the letters $\mathtt{x}$, $\mathtt{a}$ and $\mathtt{d}$ with $\mathtt{a} < \mathtt{x} < \mathtt{d}$ 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 $\tau := \inf_{\leq_{\mathrm{P}}} \{\sigma, \nu\}$. Since that the permutohedron is a lattice, τ is well-defined, and since that 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 $\tau \leq_{\mathrm{P}} \pi \leq_{\mathrm{P}} \ldots \leq_{\mathrm{P}} \sigma$ and $\tau \leq_{\mathrm{P}} \mu \leq_{\mathrm{P}} \ldots \leq_{\mathrm{P}} \nu$, there are sequences of adjacency relations $\tau \leftrightharpoons_{\mathrm{S}} \pi \leftrightharpoons_{\mathrm{S}} \ldots \leftrightharpoons_{\mathrm{S}} \sigma$, $\tau \leftrightharpoons_{\mathrm{S}^{\#}} \pi \leftrightharpoons_{\mathrm{S}^{\#}} \ldots \leftrightharpoons_{\mathrm{S}^{\#}} \sigma$, $\tau \leftrightharpoons_{\mathrm{S}} \mu \leftrightharpoons_{\mathrm{S}} \ldots \leftrightharpoons_{\mathrm{S}} \nu$ and $\tau \leftrightharpoons_{\mathrm{S}^{\#}} \mu \leftrightharpoons_{\mathrm{S}^{\#}} \ldots \leftrightharpoons_{\mathrm{S}^{\#}} \nu$. 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. This property is taken as a definition in Law and Reading's construction [LR10].

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 all $n \geq 0$, 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 $x, y \in A^*$, $\mathtt{a} < \mathtt{d} \in A$ 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 $\mathtt{y}$ in the factor y satisfying $\mathtt{a} \leq \mathtt{y} < \mathtt{d}$. In the same way, we have also $u \equiv_{\mathrm{S}^\#} v$ and thus $u \leftrightharpoons_{\mathrm{S}^\#} v$, hence the existence of a letter $\mathtt{x}$ in the factor x satisfying $\mathtt{a} < \mathtt{x} \leq \mathtt{d}$. 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 proves also that the $\leftrightharpoons_{\mathrm{B}}$ and $\rightleftharpoons_{\mathrm{B}}$-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 $\{\mathtt{b} \in A : \mathtt{a} < \mathtt{b} \leq \mathtt{c}\}$. In [NRT09], Novelli, Reutenauer and Thibon define for all $k \geq 0$ the congruence $\equiv_{\mathrm{R}^{(k)}}$. This congruence is the transitive closure of the *k-recoil adjacency relation*, defined for $\mathtt{a}, \mathtt{b} \in A$ by:

$$\mathtt{ab} \leftrightharpoons_{\mathrm{R}^{(k)}} \mathtt{ba} \qquad \text{where} \quad \mathtt{b} - \mathtt{a} \geq k. \tag{15}$$

The *k-recoil* monoid is the quotient of the free monoid by the congruence $\equiv_{\mathrm{R}^{(k)}}$. Note that the congruence $\equiv_{\mathrm{R}^{(2)}}$ restricted to permutations is the *hypoplactic congruence* [Nov98].

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

Proposition 3.10. *The $\equiv_{\mathrm{R}^{(3)}}$-equivalence classes of permutations are union of $\equiv_{\mathrm{B}}$-equivalence classes.*

Proof. This amounts to prove that for all permutations σ and ν, if $\sigma \equiv_{\mathrm{B}} \nu$ then $\sigma \equiv_{\mathrm{R}^{(3)}} \nu$. Assume that $\sigma \equiv_{\mathrm{B}} \nu$. It is enough to check this property on adjacency relations. Hence, assume that $\sigma \rightleftarrows_{\mathrm{B}} \nu$. We have $\sigma = t\mathtt{x}u\mathtt{ad}v\mathtt{y}w$ and $\nu = t\mathtt{x}u\mathtt{da}v\mathtt{y}w$ where $\mathtt{a} < \mathtt{x}, \mathtt{y} < \mathtt{d}$. Since σ and ν are permutations, $\mathtt{x} \neq \mathtt{y}$ and thus, we have $\mathtt{a} < \mathtt{x} < \mathtt{y} < \mathtt{d}$ or $\mathtt{a} < \mathtt{y} < \mathtt{x} < \mathtt{d}$, implying that $\mathtt{d} - \mathtt{a} \geq 3$. Hence, $\sigma \equiv_{\mathrm{R}^{(3)}} \nu$. □

Note that Proposition 3.10 is false for the congruence $\equiv_{\mathrm{R}^{(4)}}$. Indeed, we have $2143 \equiv_{\mathrm{B}} 2413$ but the permutations 2143 and 2413 are not $\equiv_{\mathrm{R}^{(4)}}$-equivalent. Moreover, it is plain, by definition of the congruence $\equiv_{\mathrm{R}^{(k)}}$, 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 $\equiv_{\mathrm{B}}$-equivalence classes.

4. A Robinson-Schensted-like algorithm

4.1. **Principle of the algorithm.** We describe an algorithm testing if two words are equivalent according to the Baxter congruence. Given a word $u \in A^*$, it computes its $\mathbb{P}$-symbol, that is an A-labeled pair (T_L, T_R) consisting in a left and a right binary search tree such that the non-decreasing rearrangement of u is the infix reading of both T_L and T_R. It computes also its $\mathbb{Q}$-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 infix 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.

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 pairs of unlabeled twin binary trees with n nodes, and thus, by the results of Dulucq and Guibert [DG94], also equinumerous with Baxter permutations of size n. To prove that, we shall first show that for any word u, the $\mathbb{P}$-symbol of u meets the conditions described above. Next, we will show that we have $u \equiv_{\mathrm{B}} v$ iff $\mathbb{P}(u) = \mathbb{P}(v)$, and finally, we will show that for each pair of unlabeled twin binary trees J, there is at least one permutation σ such that $\mathbb{P}(\sigma) = J$. Moreover, we shall prove that the correspondence $u \longmapsto (\mathbb{P}(u), \mathbb{Q}(u))$ yields a bijection between the words of A^* and the set of pairs $((T_L, T_R), (S_L, S_R))$ described above.

4.1.1. *The* $\mathbb{P}$*-symbol.* Recall that by Theorem 3.5 the left binary search tree obtained by leaf insertions of the letters of u from left to right encodes its $\equiv_{\mathrm{S}^\#}$-equivalence class and in the same way, the right binary search tree obtained by leaf insertions of the letters of u from right to left encodes its $\equiv_{\mathrm{S}}$-equivalence class. These two binary search trees form a pair that constitutes the $\mathbb{P}$-symbol $\mathbb{P}(u)$ of u. According to Proposition 3.7, two words u and v are $\equiv_{\mathrm{B}}$-equivalent iff they are $\equiv_{\mathrm{S}}$-equivalent and $\equiv_{\mathrm{S}^\#}$-equivalent, thus, the $\mathbb{P}$-symbol of u takes into account of both equivalence classes of u under the $\equiv_{\mathrm{S}}$ and the $\equiv_{\mathrm{S}^\#}$ congruences. Figure 5 shows the $\mathbb{P}$-symbol of $u := 5425424$.

4.1.2. *The* $\mathbb{Q}$*-symbol.* Let us first recall two algorithms. Let u be a word without repetition. Define $\mathrm{incr}(u)$, the *increasing binary tree of* u, by:

$$\mathrm{incr}(u) := \begin{cases} \perp & \text{if } u = \epsilon, \\ \mathrm{incr}(v) \wedge_{\mathtt{a}} \mathrm{incr}(w) & \text{where } u = v\mathtt{a}w \text{ and } \mathtt{a} = \min \mathrm{Alph}(u). \end{cases} \tag{16}$$

The *decreasing binary tree of* u, $\mathrm{decr}(u)$, is defined in the same way, splitting the word according to its greatest letter. The $\mathbb{Q}$-symbol of a word $u \in A^*$ is the pair (S_L, S_T) where $S_L := \mathrm{incr}\left(\mathrm{std}(u)^{-1}\right)$ and $S_R := \mathrm{decr}\left(\mathrm{std}(u)^{-1}\right)$. Figure 5 shows the $\mathbb{Q}$-symbol of $u := 5425424$, whose standardized word is 6317425, so that $\mathrm{std}(u)^{-1} = 3625714$.

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

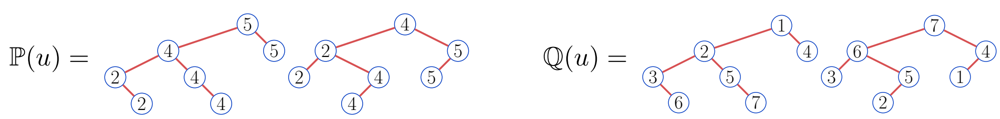


Figure 5. The $\mathbb{P}$-symbol and the $\mathbb{Q}$-symbol of $u := 5425424$.

4.2. **Correctness of the insertion algorithm.** In that follows, we say that a node x of a binary tree T is the *i-th node of T* if x is the i-th visited node by the infix traversal of T. In the same way, we say that a leaf y is the *i-th leaf of T* if y is the i-th leaf of T read from left to right.

Lemma 4.1. *Let T be a non-empty binary tree and x be the i-th leaf of T. If x is left-oriented, it is attached to the i-th node of T. If x 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. Else, we have $T = A \wedge B$. Let x be the i-th leaf of T and y be the node where x is attached. If x is also in A and $A = \perp$, x is left-oriented and is attached to the root of T (that is the first node of T) so that the lemma is satisfied. If x is in A and $A \neq \perp$, x is also the i-th leaf of A and the lemma follows by induction hypothesis on A. Otherwise, x is in B. If $B = \perp$, x is right-oriented and is attached to the root of T (that is the last node of T) so that the lemma is satisfied. Else, x is the $i-n-1$-st leaf of B where n is the number of nodes of A. Assume that the node y is the j-st node of T, then, y becomes the $j-n-1$-th node of B. Hence, the lemma follows by induction hypothesis on B. ☐

The following lemma is the key of our construction:

Lemma 4.2. *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 iff i is a recoil of σ.*

Proof. Set $\mathtt{a} := i$ and $\mathtt{b} := i + 1$. Assume that $\mathtt{a}$ is a recoil of σ. We have $\sigma = u\mathtt{b}v\mathtt{a}w$. Since that no letter $\mathtt{x}$ of u and v satisfies $\mathtt{a} < \mathtt{x} < \mathtt{b}$, the node of T labeled by $\mathtt{b}$ has a node labeled by $\mathtt{a}$ in its left subtree, itself having no right child and thus contributes, by Lemma 4.1, to a right-oriented leaf in position $i + 1$.

Conversely, assume that $\mathtt{a}$ is not a recoil of σ. We have $\sigma = u\mathtt{a}v\mathtt{b}w$. For the same reason as before, the node of T labeled by $\mathtt{a}$ has a node labeled by $\mathtt{b}$ in its right subtree, itself having no left child and thus contributes, by Lemma 4.1, to a left-oriented leaf in position $i + 1$. ☐

4.2.1. *The $\mathbb{P}$-symbol.*

Proposition 4.3. *For all word $u \in A^*$, the $\mathbb{P}$-symbol (T_L, T_R) of u is a pair of twin binary search trees where T_L (resp. T_R) is a left (resp. right) binary search tree, and the infix reading of both T_L and T_R is the non-decreasing 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 infix reading of both T_L and T_R is the non-decreasing 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 $\sigma := u$ is a permutation. Lemma 4.2 implies that the canopies of T_L and T_R are complementary because i is a recoil of σ iff i is not a recoil of $\sigma^\sim$. Thus, the shapes of T_L and T_R consist in twin binary trees. ☐

Theorem 4.4. *Let $u, v \in A^*$. Then, $u \equiv_{\mathrm{B}} v$ iff $\mathbb{P}(u) = \mathbb{P}(v)$.*

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 $\mathbb{P}$-symbol, so that $\mathbb{P}(u) = \mathbb{P}(v)$.

Conversely assume that $\mathbb{P}(u) = \mathbb{P}(v) =: (T_L, T_R)$. Since the leaf insertion of both u and v from left to right gives T_L, we have, by Theorem 3.5, $u \equiv_{\mathrm{S}\#} v$. Besides, the leaf insertion of both u and v from right to left gives T_R, so that, by the pre-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 $\{1, \ldots, n\}$ such that it is a binary search tree.

4.2.2. *The $\mathbb{Q}$-symbol.* Let us recall the following lemma of [HNT05] restated in our setting:

Lemma 4.5. *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 4.6. *For all word $u \in A^*$, the shape of the $\mathbb{Q}$-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 and their infix reading is both $\mathrm{std}(u)^{-1}$.*

Proof. By definition of the $\mathbb{Q}$-symbol, S_L and S_R are respectively the increasing and the decreasing binary trees of $\sigma := \mathrm{std}(u)^{-1}$. By Lemma 4.5, 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 Lemma 4.2, 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 infix reading. □

Theorem 4.7. *There is a bijection between the elements of A^* and the set formed by the pairs $((T_L, T_R), (S_L, S_R))$ where:*

1. *(T_L, T_R) and (S_L, S_R) are pairs of twin binary trees of same shape;*
2. *(T_L, T_R) is an A-labeled pair of twin binary search trees where T_L (resp. T_R) is a left (resp. right) binary search tree;*
3. *T_L and T_R have the same infix reading;*
4. *S_L (resp. S_R) is an increasing (resp. decreasing) binary tree;*
5. *S_L and S_R have the same infix reading.*

Moreover, the correspondence $u \longmapsto (\mathbb{P}(u), \mathbb{Q}(u))$ realizes such a bijection.

Proof. Let us first show that for all $u \in A^*$, the pair $(\mathbb{P}(u), \mathbb{Q}(u))$ satisfies the assertions of the theorem. Points (2) and (3) follow from Proposition 4.3. Points (4) and (5) follow from Proposition 4.6. Moreover, by Lemma 4.5, the assertion (1) checks out. Besides, as already mentioned, it is possible to reconstruct from the pair $(\mathbb{P}(u), \mathbb{Q}(u))$ the word u and such a word is unique. That shows that the correspondence is well-defined and injective.

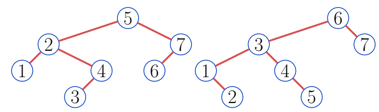
Conversely, assume that $((T_L, T_R), (S_L, S_R))$ satisfies the five assertions of the theorem. According to [HNT02], there is a bijection between the elements of A^* and the pairs (T_R, S_R) where T_R is a right binary search tree and S_R a decreasing binary tree of same shape. Let u be the word in 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). The assumption (3) implies $\mathrm{eval}(u) = \mathrm{eval}(v)$. Besides, the assumption (5) implies $\mathrm{std}(u)^{-1} = \mathrm{std}(v)^{-1}$. 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 $\mathbb{P}$-symbol of u and the pair (T_L, S_R) is the #-sylvester $\mathbb{P}$-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. □

4.2.3. *Baxter equivalence classes as linear extensions of posets.* Let T be an A-labeled binary tree. We shall denote by $\triangle(T)$ (resp. $\bigtriangledown(T)$) the poset $(N, \leq)$ where N is the set of nodes of T and $\leq$ is defined, for $x, y \in N$, by:

$$x \leq y \quad \text{if } x \text{ is an ancestor (resp. a descendant) of } y. \tag{17}$$

Let n be the number of nodes of T. If the sequence $x_1, \ldots, x_n$ is a linear extension of $\triangle(T)$ (resp. $\bigtriangledown(T)$), we shall also say that the word $u = u_1 \ldots u_n$ is a *linear extension* of $\triangle(T)$ (resp. $\bigtriangledown(T)$) if for all $1 \leq i \leq n$, the label of the node x_i is u_i.

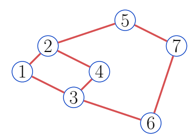
Note 4 of [HNT05] says that the words of a sylvester equivalence class encoded by a labeled right binary search tree T are exactly the linear extensions of $\bigtriangledown(T)$. Additionally, this also says that the words of a #-sylvester equivalence class encoded by a labeled left binary search tree T are exactly the linear extensions of $\triangle(T)$. Moreover, by Proposition 3.7, the words of a Baxter equivalence class are both sylvester and #-sylvester equivalent, thus the words of a Baxter equivalence class encoded by a labeled pair of twin binary search trees (T_L, T_R) are the words that are both linear extensions of $\triangle(T_L)$ and $\bigtriangledown(T_R)$. For example, consider the following labeled pair of twin binary trees,

(18) $(T_L, T_R) :=$  .

The set of words v satisfying these conditions is

(19)
$$\{5214376,\ 5214736,\ 5217436,\ 5241376,\ 5241736,\ 5247136,\ 5271436,\ 5274136,\ 5721436,\ 5724136\}.$$

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

(20)  .

4.3. **Distinguished permutations from a pair of twin binary trees.** We give in this section algorithms to read some distinguished permutations from a pair of twin binary trees.

4.3.1. *Baxter permutations.* The following algorithm allows, given a pair of twin binary search trees (T_L, T_R) labeled by a permutation, to compute the Baxter permutation belonging to the $\equiv_B$-equivalence class encoded by (T_L, T_R). This algorithm is a version adapted to our setting of the algorithm used by Dulucq and Guibert to prove their bijection between pairs of twin binary trees and Baxter permutations [DG94].

Algorithm: ExtractBaxter.
Input: A pair of twin binary search trees (T_L, T_R) labeled by a permutation.
Output: The Baxter permutation of the class encoded by (T_L, T_R).

(1) Let $\sigma := \epsilon$ be the empty permutation.
(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) Set $\sigma := \sigma.\mathtt{a}$.
  (c) Let A (resp. B) be the left (resp. right) subtree of T_L.
  (d) If the node labeled by $\mathtt{a}$ is a left child in T_R:
    (i) Then, set $T_L := A \diagup B$.
    (ii) Else, set $T_L := A \diagdown B$.
  (e) Suppress the leaf labeled by $\mathtt{a}$ in T_R.
(3) Return σ.

End.

Figure 6 shows an execution of this algorithm.

By the results of Dulucq and Guibert, ExtractBaxter terminates and computes a Baxter permutation. The only thing to prove is that the computed permutation belongs to the $\equiv_B$-equivalence class encoded by the pair of twin binary trees as input. For that, let us first prove the following lemma:

Lemma 4.8. *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 is a leaf.*

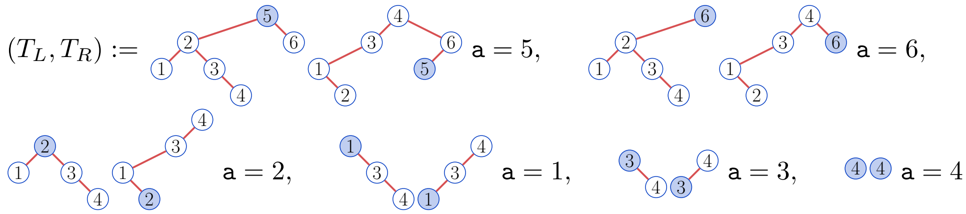


FIGURE 6. An execution of the EXTRACTBAXTER algorithm on (T_L, T_R). The computed Baxter permutation is 562134.

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.

If $A \neq \perp$, 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 4.1, 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 $A \neq \perp$. Hence, by Lemma 4.1, the i-th leaf of T_L is also attached to its $i+1$-st node and is right-oriented. Since there is a $i+1$-st node in T_L and T_R, 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 4.1, 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 4.1, 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 is a leaf. The case $B \neq \perp$ is analogous. □

Proposition 4.9. *For all pair of twin binary search trees (T_L, T_R) labeled by a permutation as input, The algorithm* EXTRACTBAXTER *computes a permutation belonging to the $\equiv_{\mathrm{B}}$-equivalence class encoded by (T_L, T_R).*

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 a pair of twin binary search trees both labeled by a same word without repetition, then EXTRACTBAXTER returns a word that is a linear extension of $\triangle(T_L)$ and a linear extension of $\triangledown(T_R)$, *i.e.*, a word belonging to the $\equiv_{\mathrm{B}}$-equivalence class encoded by (T_L, T_R). This property straightforwardly holds for $n \leq 1$. Now, assume that $T_L = A \wedge_{\mathtt{a}} B$. 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, and thus, by Lemma 4.8, there is a leaf x labeled by $\mathtt{a}$ in T_R. 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. 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 $T'_L = A \diagup B$ and T'_R is obtained by suppressing the leaf labeled by $\mathtt{a}$ into T_R. 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, 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. □

Theorem 4.10. *For all $n \geq 0$, there is a bijection between the set of equivalence classes of $\mathfrak{S}_n/_{\equiv_{\mathrm{B}}}$ and the set of unlabeled pairs of twin binary trees with n nodes.*

Proof. By Proposition 4.3 and Theorem 4.4, the $\mathbb{P}$-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 4.9, the algorithm ExtractBaxter exhibits a surjection between these two sets. Hence, these two sets are in bijection. □

Theorem 4.11. *For all $n \geq 0$, 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 4.10, C can be encoded by an unlabeled pair of twin binary trees J. By Proposition 4.9, the algorithm ExtractBaxter computes a permutation belonging to the $\equiv_{\mathrm{B}}$-equivalence class encoded by J. The theorem follows from the fact that Baxter permutations are equinumerous with unlabeled pairs of twin binary trees. □

4.3.2. *Minimal and maximal permutations.* Reading defines in [Rea05] *twisted Baxter permutations*, that are the permutations avoiding the generalized permutation patterns $2-41-3$ and $3-41-2$. These permutations are the minimal permutations of our $\equiv_{\mathrm{B}}$-equivalence classes. Indeed, assume that σ is minimal of its $\equiv_{\mathrm{B}}$-equivalence class. Then, it is not possible to perform any rewriting of the form $\mathtt{x}u\mathtt{da}v\mathtt{y} \to \mathtt{x}u\mathtt{ad}v\mathtt{y}$ where $\mathtt{a} < \mathtt{x}, \mathtt{y} < \mathtt{d}$, so that σ avoids the patterns $2-41-3$ and $3-41-2$. 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 Lemma 3.9, it is minimal. That implies that twisted Baxter permutations and Baxter permutations are equinumerous since by Theorem 4.11 there is exactly one Baxter permutation by $\equiv_{\mathrm{B}}$-equivalence class and by Proposition 3.8, there is also exactly one twisted Baxter permutation. This suggests 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 Law and Reading's setting [LR10], this bijection is the natural one. We give in that follows an algorithm to compute this bijection.

Let us consider the following algorithm which allows, given a pair of twin binary trees (T_L, T_R) labeled by a permutation, to compute the minimal permutation for the lexicographic order belonging to the $\equiv_{\mathrm{B}}$-equivalence class encoded by (T_L, T_R). By Proposition 3.8, the $\equiv_{\mathrm{B}}$-equivalence classes of permutations are intervals of the permutohedron so that the permutation computed by the following algorithm is also the minimal element for the permutohedron order of its $\equiv_{\mathrm{B}}$-equivalence class.

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

(1) Let $\sigma := \epsilon$ be the empty permutation and $F := T_L$ be a rooted forest.
(2) While F is not empty and $T_R \neq \perp$:
  (a) Let $\mathtt{a}$ be the smallest value which is both a label of a root in F and of a leaf in T_R.
  (b) Set $\sigma := \sigma.\mathtt{a}$.
  (c) Suppress the nodes labeled by $\mathtt{a}$ in F and T_R.

(3) Return σ.

End.

Note that, by choosing in the instruction (2a) the greatest label instead of the smallest, the previous algorithm computes the maximal permutation of the $\equiv_B$-equivalence class encoded by (T_L, T_R). Figure 7 shows an example of application of this algorithm.

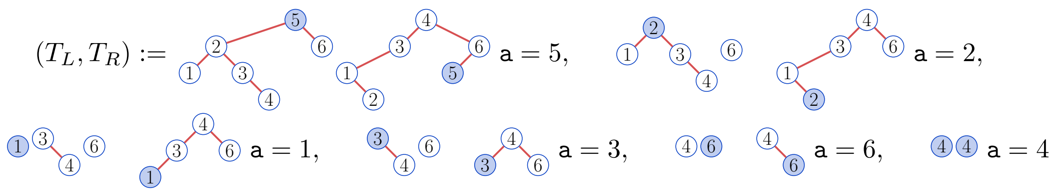


FIGURE 7. An execution of the EXTRACTMIN algorithm on (T_L, T_R). The computed permutation is 521364 and it is minimal in its $\equiv_B$-equivalence class.

Proposition 4.12. *For all pair of twin binary search trees (T_L, T_R) labeled by a permutation as input, the algorithm* EXTRACTMIN *computes the minimal permutation for the lexicographic order of the $\equiv_B$-equivalence class encoded by (T_L, T_R).*

Proof. The output σ of the algorithm EXTRACTMIN is both a linear extension of $\triangle(T_L)$ and a linear extension of $\triangledown(T_R)$. That implies that σ belongs to the $\equiv_B$-equivalence class encoded by the input pair of twin binary trees. Moreover, this algorithm terminates since by Theorem 4.11, each pair of twin binary trees (T_L, T_R) admits at least one permutation that is a common linear extension of $\triangle(T_L)$ and $\triangledown(T_R)$. Finally, the proposition comes from the fact that the smallest label is chosen at each step. □

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 $\mathbb{P}(\sigma)$ to obtain its corresponding twisted Baxter permutation. Conversely, if σ is a twisted Baxter permutation, apply EXTRACTBAXTER on $\mathbb{P}(\sigma)$ to obtain its corresponding Baxter permutation.

In the same way, there is a simple bijection between Baxter permutations and permutations that avoid the generalized permutation patterns $2-14-3$ and $3-14-2$, sending a Baxter permutation to the maximal element of its $\equiv_B$-equivalence class and conversely.

4.4. **Definition and correctness of the iterative insertion algorithm.** In what follows, we shall revise our $\mathbb{P}$-symbol algorithm that we have presented to make it iterative. Indeed, for all word u such that $\mathbb{P}(u) = (T_L, T_R)$, we propose an algorithm to insert a letter $\mathtt{a}$ into the pair of twin binary search trees (T_L, T_R) satisfying $(T_L, T_R) \leftarrow \mathtt{a} = \mathbb{P}(u\mathtt{a})$. 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, or equivalently, the product of the pairs of twin binary search trees $\mathbb{P}(u)$ and $\mathbb{P}(v)$, simply by inserting the letters of the word uv into the pair $(\perp, \perp)$.

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 $T_{\leq \mathtt{b}}$, is the right binary search tree uniquely made of the nodes x of T labeled by letters $\mathtt{a}$ satisfying $\mathtt{a} \leq \mathtt{b}$ and such that for all nodes x and y of

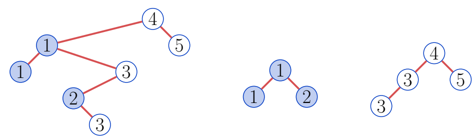


FIGURE 8. A right binary search tree T, $T_{\leq 2}$ and $T_{>2}$.

$T_{\leq \mathtt{b}}$, if x is ancestor of y in $T_{\leq \mathtt{b}}$, 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 8).

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 $T_{\leq \mathtt{a}}$ and its right subtree is $T_{>\mathtt{a}}$.

Given an A-labeled pair of twin binary search trees (T_L, T_R) where T_L (resp. T_R) is a left (resp. right) binary search tree, the *insertion* of the letter $\mathtt{a}$ of 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* $\mathbb{P}$*-symbol* (T_L, T_R) of a word $u \in A^*$ is computed by iteratively inserting the letters of u, from left to right, into $(\perp, \perp)$. The *iterative* $\mathbb{Q}$*-symbol* (S_L, S_R) is constructed by recording in S_L (resp. S_R) the dates of creation of each node of T_L (resp. T_R) (see Figure 9).

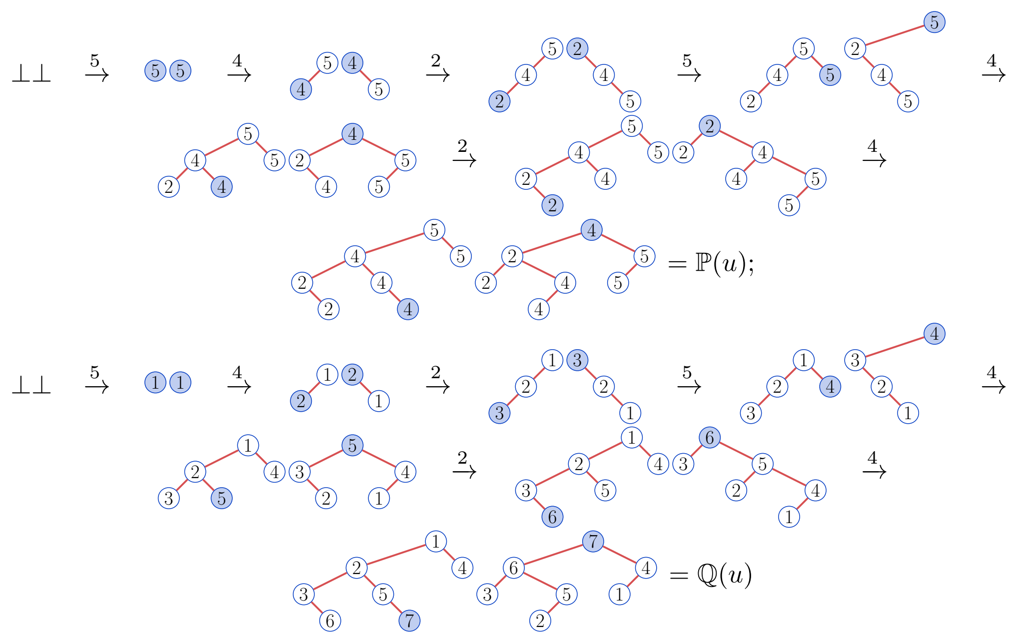


FIGURE 9. Steps of the computation of the $\mathbb{P}$-symbol and the $\mathbb{Q}$-symbol of $u := 5425424$.

To show that the iterative version of the $\mathbb{P}$-symbol computes the same labeled pair of twin binary trees than its non-iterative version, we need the following lemma:

Lemma 4.13. *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 is also 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, $T = S_{\leq \mathtt{a}} \wedge_{\mathtt{a}} S_{>\mathtt{a}}$. 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 $v_{\leq \mathtt{a}} := v_{|]-\infty,\mathtt{a}]}$ and the subtree R' only depends on the subword $v_{>\mathtt{a}} := v_{|]\mathtt{a},+\infty[}$, so that, by induction hypothesis, $L' = S_{\leq \mathtt{a}}$, $R' = S_{>\mathtt{a}}$ and thus, $T = T'$. □

Proposition 4.14. *For all $u \in A^*$, the $\mathbb{P}$-symbol of u and the iterative $\mathbb{P}$-symbol of u are equal.*

Proof. Let (T_L, T_R) be the $\mathbb{P}$-symbol of u and (T'_L, T'_R) be the iterative $\mathbb{P}$-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 4.13, we have $T_R = T'_R$. □

The correctness of the iterative version of the $\mathbb{Q}$-symbol algorithm comes from the correctness of the iterative $\mathbb{P}$-algorithm.

5. The Baxter lattice

5.1. **The Baxter lattice congruence.** Recall that an equivalence relation $\equiv$ on the elements of a lattice $(L, \leq)$ is a *lattice congruence* [Rea05, CS98] if the following three conditions hold.

(1) Every $\equiv$-equivalence class is an interval of L;
(2) For all $x, y \in L$, if $x \leq y$ then $x\downarrow \leq y\downarrow$ where $x\downarrow$ is the maximal element of the $\equiv$-equivalence class of x;
(3) For all $x, y \in L$, if $x \leq y$ then $x\uparrow \leq y\uparrow$ where $x\uparrow$ is the minimal element of the $\equiv$-equivalence class of x.

In this section, we shall prove that the Baxter monoid congruence is also a lattice congruence of the permutohedron. Proposition 3.8 says that the $\equiv_{\mathrm{B}}$-equivalence classes of permutations are intervals of the permutohedron, so that $\equiv_{\mathrm{B}}$ satisfies point (1). Figure 10 shows the $\equiv_{\mathrm{B}}$-equivalence classes in the permutohedron of order 4.

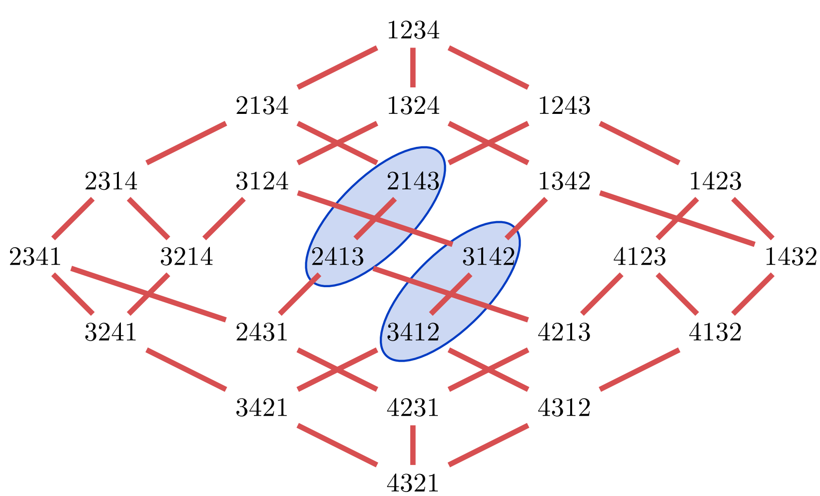


Figure 10. The permutohedron of order 4 and the two non-singleton $\equiv_{\mathrm{B}}$-equivalence classes.

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

Proposition 5.1. *Let σ and ν be two permutations such that $\sigma \leq_{\mathrm{P}} \nu$. Then, $\sigma\uparrow \leq_{\mathrm{P}} \nu\uparrow$ and $\sigma\downarrow \leq_{\mathrm{P}} \nu\downarrow$.*

Proof. We shall only prove that $\sigma\uparrow \leq_{\mathrm{P}} \nu\uparrow$, the proof of $\sigma\downarrow \leq_{\mathrm{P}} \nu\downarrow$ being analogous. 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 $\sigma = \sigma\uparrow$, then $\sigma\uparrow \leq_{\mathrm{P}} \nu \leq_{\mathrm{P}} \nu\uparrow$ and the property holds. Else, 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 $\sigma \leq_{\mathrm{P}} \pi$. It then remains to prove that there exists a permutation μ such that $\nu \equiv_{\mathrm{B}} \mu$ and $\pi \leq_{\mathrm{P}} \mu$ since this leads to show, by applying iteratively this reasoning, that $\sigma\uparrow$ is smaller than a permutation belonging to the $\equiv_{\mathrm{B}}$-equivalence class of ν for the permutohedron order and hence, by transitivity, that $\sigma\uparrow \leq_{\mathrm{P}} \nu\uparrow$. We have three cases:

(1) If $j \leq i-2$, σ is of the form $\sigma = u\mathtt{ab}v\mathtt{cd}w$ where $\mathtt{a}$ (resp. $\mathtt{c}$) is the j-th (resp. i-th) letter of σ and $\mathtt{a} < \mathtt{b}$ 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, $\pi \leq_{\mathrm{P}} \mu$, so that μ is appropriate.
(2) If $j \geq i+2$, this is analogous to the previous case.
(3) If $j = i+1$, σ is of the form $\sigma = u\mathtt{abc}v$ where $\mathtt{a}$ is the i-th letter of σ and $\mathtt{a} < \mathtt{b} < \mathtt{c}$ 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 $\pi \leq_{\mathrm{P}} \mu$, hence, μ is appropriate.
(4) If $j = i-1$, this is analogous to the previous case.

□

5.2. A lattice structure on the set of pairs of twin binary trees.

Definition 5.2. *For all $n \geq 0$, define the order relation $\leq_{\mathrm{B}}$ on the set $\mathcal{TBT}_n$ setting $J_0 \leq_{\mathrm{B}} J_1$, where $J_0, J_1 \in \mathcal{TBT}_n$, if there exists $\sigma, \nu \in \mathfrak{S}_n$ such that $\mathbb{P}(\sigma) = J_0$, $\mathbb{P}(\nu) = J_1$ and $\sigma \leq_{\mathrm{P}} \nu$.*

To describe the covering relations of the poset $(\mathcal{TBT}_n, \leq_{\mathrm{B}})$ in terms of transformations on pairs of twin binary trees, it suffices to consider a pair of twin binary trees J and the maximal permutation σ of the $\equiv_{\mathrm{B}}$-equivalence class encoded by J. The coverings of J are the pairs of twin binary trees $\mathbb{P}(\nu)$ where $\nu = \sigma s_i$ and i is not a descent of σ. In this way, the pair of twin binary trees (T_L, T_R) is covered by (T'_L, T'_R) if one of the three following conditions is satisfied:

(1) T'_L is obtained by performing a left rotation into T_L such that $\mathrm{cnp}(T_L) = \mathrm{cnp}(T'_L)$ and $T'_R = T_R$;
(2) T'_R is obtained by performing a right rotation into T_R such that $\mathrm{cnp}(T_R) = \mathrm{cnp}(T'_R)$ and $T'_L = T_L$;
(3) T'_L (resp. T'_R) is obtained by performing a left (resp. right) rotation into T_L (resp. T_R) such that $\mathrm{cnp}(T_L) \neq \mathrm{cnp}(T'_L)$ (resp. $\mathrm{cnp}(T_R) \neq \mathrm{cnp}(T'_R)$).

Figure 11 shows an interval of the poset of the pair of twin binary trees.

Moreover, it is possible to compare two pairs of twin binary trees $J_0 := (T^0_L, T^0_R)$ and $J_1 := (T^1_L, T^1_R)$ very easily by computing the *Tamari vector* [Knu06] of each binary tree. Recall that

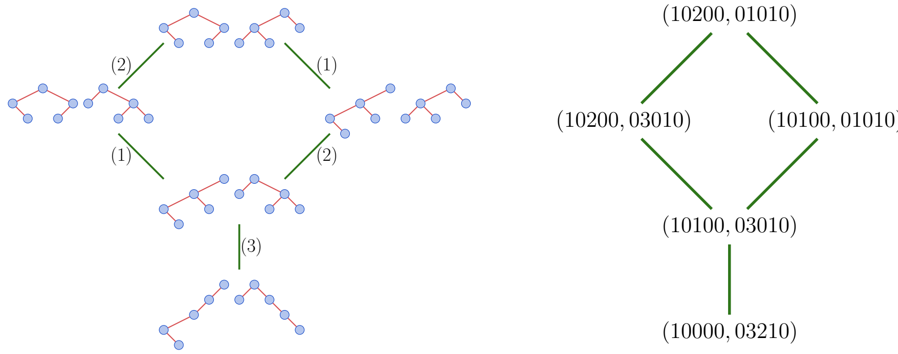


FIGURE 11. An interval of the lattice of the pairs of twin binary trees of order 5 and the corresponding Tamari vectors. The edges are labeled according to the type of the covering relation.

the Tamari vector of a binary tree T is computed by labeling each node x of T by the number of nodes of the right subtree of x and by considering its infix reading. Two binary trees T_0 and T_1 satisfy $T_0 \leq_{\mathrm{T}} T_1$ in the Tamari lattice iff the Tamari vector of T_0 is not greater component by component than the Tamari vector of T_1 [Knu06]. Hence, by the nature of the covering relations in $(\mathcal{TBT}_n, \leq_{\mathrm{B}})$, we have $J_0 \leq_{\mathrm{B}} J_1$ iff $T_L^1 \leq_{\mathrm{T}} T_L^0$ and $T_R^0 \leq_{\mathrm{T}} T_R^1$.

Proposition 5.3. *For all $n \geq 0$, the poset $(\mathcal{TBT}_n, \leq_{\mathrm{B}})$ is a lattice.*

Proof. The equivalence relation $\equiv_{\mathrm{B}}$ is also a lattice congruence of the permutohedron since it satisfies, by Propositions 3.8 and 5.1 the points (1), (2) and (3). Besides, by Theorem 4.10, the unlabeled pairs of twin binary trees encode exactly the equivalence classes of permutations under the Baxter congruence so that $(\mathcal{TBT}_n, \leq_{\mathrm{B}})$ is a lattice. □

6. The Baxter Hopf Algebra

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

6.1. **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 [DHT02]. Its product and its coproduct are defined by:

$$\mathbf{F}_\sigma \cdot \mathbf{F}_\nu := \sum_{\pi \,\in\, \sigma \overline{\sqcup\!\sqcup} \nu} \mathbf{F}_\pi, \tag{21}$$

$$\Delta(\mathbf{F}_\sigma) := \sum_{\sigma = uv} \mathbf{F}_{\mathrm{std}(u)} \otimes \mathbf{F}_{\mathrm{std}(v)}. \tag{22}$$

For example,

$$\begin{aligned} \mathbf{F}_{132} \cdot \mathbf{F}_{12} = \mathbf{F}_{13245} + \mathbf{F}_{13425} + \mathbf{F}_{13452} + \mathbf{F}_{14325} + \mathbf{F}_{14352} \\ + \mathbf{F}_{14532} + \mathbf{F}_{41325} + \mathbf{F}_{41352} + \mathbf{F}_{41532} + \mathbf{F}_{45132}, \end{aligned} \tag{23}$$

$$\begin{aligned} \Delta(\mathbf{F}_{35142}) = 1 \otimes \mathbf{F}_{35142} + \mathbf{F}_1 \otimes \mathbf{F}_{4132} + \mathbf{F}_{12} \otimes \mathbf{F}_{132} \\ + \mathbf{F}_{231} \otimes \mathbf{F}_{21} + \mathbf{F}_{2413} \otimes \mathbf{F}_1 + \mathbf{F}_{35142} \otimes 1. \end{aligned} \tag{24}$$

If $\equiv$ is an equivalence relation on $\mathfrak{S}$ and $\sigma \in \mathfrak{S}$, denote by $\widehat{\sigma}$ the $\equiv$-equivalence class of σ.

The following theorem due to Hivert and Nzeutchap [HN07] 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:*

$$\mathbf{P}_{\widehat{\sigma}} := \sum_{\sigma\in\widehat{\sigma}} \mathbf{F}_\sigma \tag{25}$$

spans a Hopf subalgebra of **FQSym**.

6.2. **The Hopf algebra Baxter.**

6.2.1. *A construction from the Baxter monoid.* By definition, $\equiv_{\mathrm{B}}$ is a congruence, and, by Propositions 3.2 and 3.3, $\equiv_{\mathrm{B}}$ satisfies the conditions of Theorem 6.1. Moreover, by Theorem 4.10, the $\equiv_{\mathrm{B}}$-equivalence classes of permutations can be encoded by pairs of unlabeled twin binary trees. Hence, we have the following theorem:

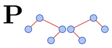
Theorem 6.2. *The family $\{\mathbf{P}_J\}_{J\in\mathcal{TBT}}$ defined by:*

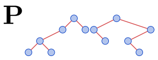
$$\mathbf{P}_J := \sum_{\substack{\sigma\in\mathfrak{S}\\ \mathbb{P}(\sigma)=J}} \mathbf{F}_\sigma \tag{26}$$

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

For example,

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

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

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

The Hilbert series of **Baxter** is

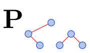
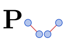
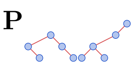
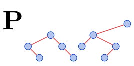
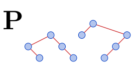
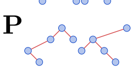
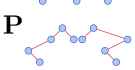
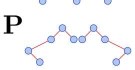
$$B(z) := 1 + z + 2z^2 + 6z^3 + 22z^4 + 92z^5 + 422z^6 + 2074z^7 + 10754z^8 + 58202z^9 + \ldots, \tag{30}$$

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 obtain

$$\mathbf{P}_{J_0} \cdot \mathbf{P}_{J_1} = \sum_{\substack{\mathbb{P}(\sigma)=J_0,\ \mathbb{P}(\nu)=J_1\\ \pi\ \in\ \sigma \,\overline{\sqcup\!\sqcup}\, \nu\ \cap\ \mathfrak{S}^{\mathrm{B}}}} \mathbf{P}_{\mathbb{P}(\pi)}. \tag{31}$$

For example,

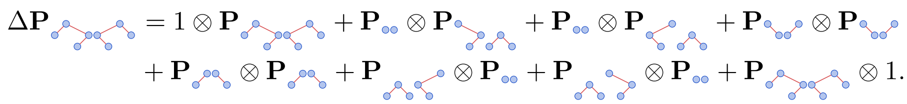
(32) $\mathbf{P}$  $\cdot\, \mathbf{P}$  $= \mathbf{P}$  $+ \mathbf{P}$  $+ \mathbf{P}$ 
$+ \mathbf{P}$  $+ \mathbf{P}$  $+ \mathbf{P}$  .

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

$$\Delta(\mathbf{P}_J) = \sum_{\substack{\mathbb{P}(\pi)=J \\ \pi=u.v \\ \sigma:=\mathrm{std}(u),\ \nu:=\mathrm{std}(v)\in\mathfrak{S}^{\mathrm{B}}}} \mathbf{P}_{\mathbb{P}(\sigma)} \otimes \mathbf{P}_{\mathbb{P}(\nu)}. \tag{33}$$

For example,

(34)



6.2.2. *Set-bases.* We shall call a basis of an algebra (resp. coalgebra) a *set-algebra basis* (resp. *set-coalgebra basis*) if each element of the basis (resp. tensorial square of the basis) occurs only with coefficient 0 or 1 in any product (resp. coproduct) involving two (resp. one) elements of the basis. Law and Reading have proved in [LR10] that the basis of their Baxter Hopf algebra, analog to our basis $\{\mathbf{P}_J\}_{J\in\mathcal{TBT}}$, is both a set-algebra basis and a set-coalgebra basis. We re-prove this result in our setting:

Proposition 6.3. *The basis* $\{\mathbf{P}_J\}_{J\in\mathcal{TBT}}$ *is both a set-algebra basis and a set-coalgebra basis of* **Baxter***.*

Proof. It is immediate from (31) that $\{\mathbf{P}_J\}_{J\in\mathcal{TBT}}$ is a set-algebra basis.

To prove that $\{\mathbf{P}_J\}_{J\in\mathcal{TBT}}$ is also a set-coalgebra basis, it is enough to prove that the inverse of two different permutations σ and ν arising in a same shifted shuffle $\pi \,\overline{\sqcup\!\sqcup}\, \mu$ are not $\equiv_{\mathrm{B}}$-equivalent. We shall prove this property for the sylvester congruence. Indeed, the result will follow from the fact that, by Proposition 3.7, the Baxter congruence is finer than the sylvester congruence. Assume that $\sigma^{-1}\equiv_{\mathrm{S}}\nu^{-1}$. Then, by Theorem 3.5, the permutations σ^{-1} and ν^{-1} give the same binary search tree when inserted from right to left. By Lemma 4.5, that implies that the shape of $\mathrm{decr}(\sigma)$ and $\mathrm{decr}(\nu)$ are the same. Writing $\sigma = t\mathtt{a}u$ and $\nu = v\mathtt{a}w$, where $\mathtt{a}$ is the maximal letter of both σ and ν, it is plain that the position of $\mathtt{a}$ in σ and ν are the same. The permutation μ is of the form $\mu = \mu'\mathtt{b}\mu''$ where $\mathtt{b}$ is the maximal letter of μ, and then, π is of the form $\pi = \pi'\pi''$ where $|\pi'| = |t| - |\mu'|$. Now, we have $t, v \in \pi' \,\overline{\sqcup\!\sqcup}\, \mu'$ and $u, w \in \pi'' \,\overline{\sqcup\!\sqcup}\, \mu''$. It follows by induction on $|\sigma| + |\nu|$ that $\sigma = \nu$, contradicting our hypothesis. □

6.2.3. *Polynomial realization of* **Baxter**. Set $\mathbf{G}_\sigma := \mathbf{F}_{\sigma^{-1}}$. Recall that the *free quasi-ribbon* $r(\mathbf{G}_\sigma)$ of $\mathbf{G}_\sigma$ is the polynomial of $\mathbb{K}\langle A^*\rangle$ defined by:

$$r(\mathbf{G}_\sigma) := \sum_{\substack{u\in A^* \\ \mathrm{std}(u)=\sigma}} u. \tag{35}$$

For example,

$$r(\mathbf{G}_\epsilon) = 1, \tag{36}$$

$$r(\mathbf{G}_1) = \sum_i a_i = a_0 + a_1 + a_2 + \ldots, \tag{37}$$

$$r(\mathbf{G}_{231}) = \sum_{k<i\leq j} a_i a_j a_k = a_1a_1a_0 + a_1a_2a_0 + a_1a_3a_0 + \ldots. \tag{38}$$

These polynomials provide a realization of **FQSym** as an algebra [DHT02]; Indeed, r is an injective algebra morphism from **FQSym** to $\mathbb{K}\langle A^*\rangle$. We deduce from (35) and Lemma 4.5 the following realization of **Baxter**:

Theorem 6.4. *The map r defined in (35), restricted to* **Baxter**, *satisfies*

$$r\left(\mathbf{P}_{(T_L,T_R)}\right) = \sum_{\substack{u \in A^* \\ \operatorname{incr}(\operatorname{std}(u)) \simeq T_L \\ \operatorname{decr}(\operatorname{std}(u)) \simeq T_R}} u \tag{39}$$

and provides a realization of **Baxter** *as an algebra, where $\simeq$ stands for the equality of binary tree shapes.*

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

Proposition 6.5. *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 (25) can be expressed as:*

$$\mathbf{P}_{\widehat{\sigma}} \cdot \mathbf{P}_{\widehat{\nu}} = \sum_{\substack{\widehat{\sigma}\downarrow \,\diagup\, \widehat{\nu}\downarrow \;\leq_{\mathrm{P}}\; \pi \;\leq_{\mathrm{P}}\; \widehat{\sigma}\uparrow \,\diagdown\, \widehat{\nu}\uparrow \\ \pi = \min \widehat{\pi}}} \mathbf{P}_{\widehat{\pi}}. \tag{40}$$

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

$$\left(\sum_{\sigma \;\leq_{\mathrm{P}}\; \mu \;\leq_{\mathrm{P}}\; \sigma'} \mathbf{F}_{\mu}\right) \cdot \left(\sum_{\nu \;\leq_{\mathrm{P}}\; \tau \;\leq_{\mathrm{P}}\; \nu'} \mathbf{F}_{\tau}\right) = \sum_{\sigma \,\diagup\, \nu \;\leq_{\mathrm{P}}\; \pi \;\leq_{\mathrm{P}}\; \sigma' \,\diagdown\, \nu'} \mathbf{F}_{\pi}. \tag{41}$$

By (41) and since that every $\equiv$-equivalence class is an interval of the permutohedron, we obtain

$$\mathbf{P}_{\widehat{\sigma}} \cdot \mathbf{P}_{\widehat{\nu}} = \sum_{\widehat{\sigma}\downarrow \,\diagup\, \widehat{\nu}\downarrow \;\leq_{\mathrm{P}}\; \pi \;\leq_{\mathrm{P}}\; \widehat{\sigma}\uparrow \,\diagdown\, \widehat{\nu}\uparrow} \mathbf{F}_{\pi}. \tag{42}$$

By Theorem 6.1, the expression (42) can be expressed as a sum of $\mathbf{P}_{\widehat{\pi}}$ elements. □

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 $J_0 \diagup J_1$ by:

$$J_0 \diagup J_1 := (T_L^0 \diagdown T_L^1,\; T_R^0 \diagup T_R^1). \tag{43}$$

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

$$J_0 \diagdown J_1 := (T_L^0 \diagup T_L^1,\; T_R^0 \diagdown T_R^1). \tag{44}$$

Proposition 6.5 leads to the following expression for the product of **Baxter**:

Corollary 6.6. *For all pairs of twin binary trees J_0 and J_1, the product of* **Baxter** *satisfies*

$$\mathbf{P}_{J_0} \cdot \mathbf{P}_{J_1} = \sum_{J_0 \,\diagup\, J_1 \;\leq_{\mathrm{B}}\; J \;\leq_{\mathrm{B}}\; J_0 \,\diagdown\, J_1} \mathbf{P}_{J}. \tag{45}$$

6.3. **Connections with other Hopf subalgebras of FQSym.**

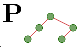
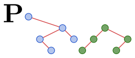
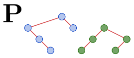
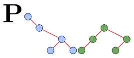
6.3.1. *Connection with the Hopf algebra* **PBT**. It is well-known that the sylvester congruence leads to the construction of the Hopf subalgebra **PBT** [LR98] of **FQSym**, whose fundamental basis $\{\mathbf{P}_T\}_{T \in \mathcal{BT}}$ is defined in accordance with (25) (see [HNT02] and [HNT05]). By Proposition 3.7, every $\equiv_{\mathrm{S}}$-equivalence class is an union of some $\equiv_{\mathrm{B}}$-equivalence classes. Hence, we have the following injective Hopf map:

$$\rho : \mathbf{PBT} \hookrightarrow \mathbf{Baxter} \tag{46}$$

satisfying

$$\rho\left(\mathbf{P}_T\right) = \sum_{\substack{T' \in \mathcal{BT} \\ J := (T', T) \,\in\, \mathcal{TBT}}} \mathbf{P}_J. \tag{47}$$

For example,

(48) $\rho\left(\mathbf{P}\right.$  $\left.\right) = \mathbf{P}$  $+\, \mathbf{P}$  $+\, \mathbf{P}$  .

6.3.2. *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 $\{\mathbf{P}_{\widehat{\sigma}}\}_{\widehat{\sigma} \in \mathfrak{S}/_{\equiv_{\mathrm{R}^{(3)}}}}$ is defined in accordance with (25) (see [NRT09]). By Proposition 3.10, every $\equiv_{\mathrm{R}^{(3)}}$-equivalence class of permutations is an union of some $\equiv_{\mathrm{B}}$-equivalence classes. Hence, we have the following injective Hopf map:

$$\alpha : \mathbf{DSym}^{(3)} \hookrightarrow \mathbf{Baxter} \tag{49}$$

satisfying

$$\alpha\left(\mathbf{P}_{\widehat{\sigma}}\right) = \sum_{\sigma \,\in\, \widehat{\sigma} \cap \mathfrak{S}^{\mathrm{B}}} \mathbf{P}_{\mathbb{P}(\sigma)}. \tag{50}$$

6.3.3. *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. This shows that 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 $b_i = 1$ iff i is a recoil of the elements of $\widehat{\sigma}$. We denote by $\{\mathbf{P}_b\}_{b \in \{0,1\}^*}$ 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:

$$\beta : \mathbf{Sym} \hookrightarrow \mathbf{PBT} \tag{51}$$

satisfies, thanks to the fact that the hypoplactic equivalence classes are union of $\equiv_{\mathrm{S}}$-equivalence classes and Lemma 4.2,

$$\beta\left(\mathbf{P}_b\right) = \sum_{\substack{T \in \mathcal{BT} \\ \mathrm{cnp}(T) = b}} \mathbf{P}_T. \tag{52}$$

Combinatorially, the map β associates to a binary word b the sum of the binary trees having b as canopy. The composition $\rho \circ \beta$ is an injective Hopf map from **Sym** to **Baxter**. Combinatorially, it associates to a binary word b the sum of the pairs of twin binary trees (T_L, T_R) where the canopy of T_R is b.

Figure 12 summarizes the relations between these considered Hopf algebras close to **Baxter**.

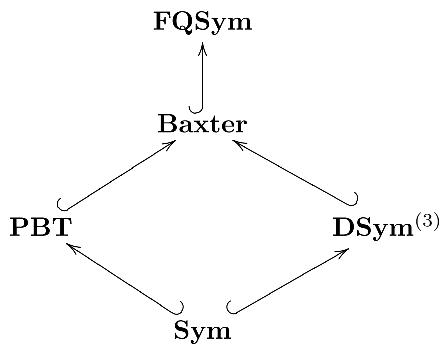


FIGURE 12. Diagram of Hopf maps between some Hopf algebras related to **Baxter**.

6.4. **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:

$$\mathbf{E}^\sigma := \sum_{\sigma \leq_{\mathrm{P}} \sigma'} \mathbf{F}_\sigma, \tag{53}$$

$$\mathbf{H}^\sigma := \sum_{\sigma' \leq_{\mathrm{P}} \sigma} \mathbf{F}_\sigma, \tag{54}$$

form multiplicative bases of **FQSym** [DHNT08]. Indeed, for all $\sigma, \nu \in \mathfrak{S}$, the product satisfies

$$\mathbf{E}^\sigma \cdot \mathbf{E}^\nu = \mathbf{E}^{\sigma \,\diagup\, \nu}, \tag{55}$$

$$\mathbf{H}^\sigma \cdot \mathbf{H}^\nu = \mathbf{H}^{\sigma \,\diagdown\, \nu}. \tag{56}$$

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:

$$\mathbf{E}_J := \sum_{J \leq_{\mathrm{B}} J'} \mathbf{P}_{J'}, \tag{57}$$

$$\mathbf{H}_J := \sum_{J' \leq_{\mathrm{B}} J} \mathbf{P}_{J'}. \tag{58}$$

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

Proposition 6.7. *Let J be a pair of twin binary trees and $\sigma\!\downarrow$ (resp. $\sigma\!\uparrow$) be the minimal (resp. maximal) permutation such that $\mathbb{P}(\sigma\!\downarrow) = J$ (resp. $\mathbb{P}(\sigma\!\uparrow) = J$). Then,*

$$\mathbf{E}_J = \mathbf{E}^{\sigma\downarrow}, \tag{59}$$

$$\mathbf{H}_J = \mathbf{H}^{\sigma\uparrow}. \tag{60}$$

Proof. By definition, we have

$$\mathbf{E}_J = \sum_{\substack{\nu\in\mathfrak{S} \\ J \leq_{\mathrm{B}} \mathbb{P}(\nu)}} \mathbf{F}_\nu. \tag{61}$$

Assume that the element $\mathbf{F}_\nu$ arises in (61). That implies that $J \leq_{\mathrm{B}} \mathbb{P}(\nu)$. By Propositions 3.8 and 5.1, we have $\sigma\!\downarrow \leq_{\mathrm{P}} \nu\!\downarrow \leq_{\mathrm{P}} \nu$, and hence, the element $\mathbf{F}_\nu$ arises also in $\mathbf{E}^{\sigma\downarrow}$. Conversely, assume that the element $\mathbf{F}_\nu$ arises in $\mathbf{E}^{\sigma\downarrow}$. That implies that $\sigma\!\downarrow \leq_{\mathrm{P}} \nu$. By definition of the order relation $\leq_{\mathrm{B}}$, we have $\mathbb{P}(\sigma\!\downarrow) \leq_{\mathrm{B}} \mathbb{P}(\nu)$. Since $J = \mathbb{P}(\sigma\!\downarrow)$, the element $\mathbf{F}_\nu$ arises also in (61). The proof for the homogeneous family is analogous. □

Corollary 6.8. *For all $J_0, J_1 \in \mathcal{TBT}$, we have*

$$\mathbf{E}_{J_0} \cdot \mathbf{E}_{J_1} = \mathbf{E}_{J_0 \,\diagup\, J_1}, \tag{62}$$

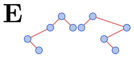
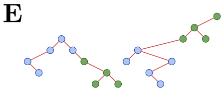
$$\mathbf{H}_{J_0} \cdot \mathbf{H}_{J_1} = \mathbf{H}_{J_0 \,\diagdown\, J_1}. \tag{63}$$

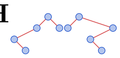
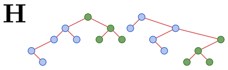
Proof. Let σ and ν be the minimal permutations of the $\equiv_{\mathrm{B}}$-equivalence classes respectively encoded by J_0 and J_1. By Proposition 6.7, we have

$$\mathbf{E}_{J_0} \cdot \mathbf{E}_{J_1} = \mathbf{E}^{\sigma} \cdot \mathbf{E}^{\nu} = \mathbf{E}^{\sigma \,\diagup\, \nu}. \tag{64}$$

The permutation $\sigma \diagup \nu$ is obviously the minimal element of its $\equiv_{\mathrm{B}}$-equivalence class, and, by the definition of the insertion algorithm, the $\mathbb{P}$-symbol of $\sigma \diagup \nu$ is the pair of twin binary trees $(T_L^0 \diagdown T_L^1, T_R^0 \diagup T_R^1) = J_0 \diagup J_1$. The proof of the second part of the proposition is analogous. □

For example,

(65) $\mathbf{E}$  $\cdot\, \mathbf{E}$  $= \mathbf{E}$  ,

(66) $\mathbf{H}$  $\cdot\, \mathbf{H}$  $= \mathbf{H}$  .

Let us say that a pair of twin binary trees J is *connected* (resp. *anti-connected*) if the minimal (resp. maximal) permutation of the $\equiv_{\mathrm{B}}$-equivalence class encoded by J is connected (resp. anti-connected).

Corollary 6.9. *The algebra* **Baxter** *is free on the elements $\mathbf{E}_J$ (resp. $\mathbf{H}_J$) where J is a connected (resp. anti-connected) pair of twin binary trees.*

Proof. First, since every permutation σ can be expressed as $\sigma = \sigma' \diagup \dots \diagup \sigma''$ where the permutations $\sigma', \dots, \sigma''$ are connected, by Corollary 6.8, it is possible to express any element $\mathbf{E}_J$ as a product of $\mathbf{E}_{J'} \cdot \dots \cdot \mathbf{E}_{J''}$ where the pairs of twin binary trees $J', \dots, J''$ are connected.

Moreover, since there is no relation in **FQSym** between the elements $\mathbf{E}^\sigma$ where σ is a connected permutation [DHT02], by Corollary 6.8, there is either 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. □

Actually, as we shall show in the following proposition, J is connected iff the Baxter permutation belonging to the $\equiv_{\mathrm{B}}$-equivalence class encoded by J is connected. That implies that **Baxter** is free on the elements $\mathbf{E}_J$ where the Baxter permutation belonging to the $\equiv_{\mathrm{B}}$-equivalence class encoded by J is connected.

Proposition 6.10. *If σ is a connected Baxter permutation, then the minimal permutation belonging to the $\equiv_{\mathrm{B}}$-equivalence class of σ is also connected.*

Proof. Let us prove by induction on n that for $\sigma \in \mathfrak{S}_n$ a connected Baxter permutation, the permutation $\sigma\!\downarrow$ is also connected. The property is easily checked by hand for $n \leq 4$. Let $n \geq 5$ and $\sigma \in \mathfrak{S}_n$ be a connected Baxter permutation. By Proposition 3.8 and by Lemma

3.9, it is possible to reach $\sigma\downarrow$ from σ by applying adjacency transformations of the form $\mathtt{x}u\mathtt{da}v\mathtt{y} \to \mathtt{x}u\mathtt{ad}v\mathtt{y}$ where $\mathtt{a} < \mathtt{x}, \mathtt{y} < \mathtt{d}$, as much as possible. Hence, we have a sequence $\sigma \to \pi \to \ldots \to \sigma\downarrow$ rewriting the connected Baxter permutation σ into $\sigma\downarrow$. Set $I := [1, n-1]$. By Proposition 3.3, we have the sequence $\sigma_{|I} \Rightarrow \pi_{|I} \Rightarrow \ldots \Rightarrow \sigma\downarrow_{|I}$ where $\Rightarrow$ stands for $\to$ or $=$. Note that $\sigma_{|I}$ is a Baxter permutation. We have now two cases:

If $\sigma_{|I}$ is connected, by induction hypothesis, $\sigma\downarrow_{|I}$ is also connected. The permutation $\sigma\downarrow$ is obtained from $\sigma\downarrow_{|I}$ by inserting n. If $\sigma\downarrow = \sigma\downarrow_{|I} .n$, since any permutation of a given $\equiv_{\mathrm{B}}$-equivalence class ends by the same letter, that would imply that σ also ends by the letter n and thus, that σ is not connected, contradicting our hypothesis. Otherwise, for any other place of insertion of n, the permutation $\sigma\downarrow$ is clearly still connected.

If $\sigma_{|I}$ is not connected, the permutations σ and $\sigma_{|I}$ are of the form $\sigma = u\mathtt{n}vw$ and $\sigma_{|I} = uvw$ where $\mathtt{n}$ is the maximal letter of σ and uv the shortest non-empty prefix of $\sigma_{|I}$ that is a permutation. Since σ is connected, $v \neq \epsilon$ and since $\sigma_{|I}$ is not connected, $w \neq \epsilon$. Now, if $u = \epsilon$, the first letter of σ is $\mathtt{n}$, and, since that any element of a $\equiv_{\mathrm{B}}$-equivalence class begins by the same letter, $\sigma\downarrow$ is also connected. Otherwise, since $\sigma_{|I}$ is not connected and uv is the shortest non-empty prefix of $\sigma_{|I}$ that is a permutation, for all $\mathtt{c} \in \mathrm{Alph}(w)$ and $\mathtt{b} \in \mathrm{Alph}(uv)$, $\mathtt{b} < \mathtt{c}$. That implies that it is not possible to disconnect σ by $\to$ rewritings. Indeed, any rewriting $\to$ transposing two letters in the factors u or vw does not disconnect σ. The only way to disconnect σ is by moving its letter $\mathtt{n}$ to the right. Assuming that it is possible, and denoting by $\mathtt{a}$ the first letter of v, we would have $\mathtt{a} < \mathtt{b} < \mathtt{c} < \mathtt{n}$ implying that σ would contain the pattern $2-41-3$ and σ would not be a Baxter permutation, which is contradictory with our assumptions. That implies that any permutation reachable from σ by $\to$ rewritings is connected, and in particular, $\sigma\downarrow$ is. □

Note that the previous proposition becomes false if we assume that the maximal permutation of its $\equiv_{\mathrm{B}}$-equivalence is connected instead of the minimal. Indeed, considering the $\equiv_{\mathrm{B}}$-equivalence class $\{2143, 2413\}$, the permutation 2413 is connected and maximal of its class but the Baxter permutation 2143 is not connected.

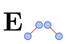
Corollary 6.11. *The algebra* **Baxter** *is free on the elements* $\mathbf{E}_J$ *where the Baxter permutation belonging to the* $\equiv_{\mathrm{B}}$*-equivalence class encoded by* J *is connected.*

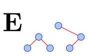
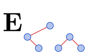
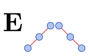
The generating series $B_C(z)$ of connected Baxter permutations, and hence, algebraic generators of **Baxter** satisfies

$$B_C(z) = 1 - \frac{1}{B(z)}. \tag{67}$$

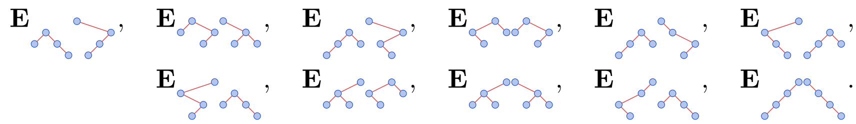
First dimensions of algebraic generators of **Baxter** are 1, 1, 1, 3, 11, 47, 221, 1113, 5903, 32607, 186143, 1092015. Here follows algebraic generators of **Baxter** of order 1 to 4:

(68) $\mathbf{E}$;

(69) $\mathbf{E}$;

(70) $\mathbf{E}$, $\mathbf{E}$, $\mathbf{E}$;

(71)



6.5. **Bidendriform bialgebra structure.** A Hopf algebra $(H, \cdot, \Delta)$ can be fitted into a bidendriform bialgebra structure [Foi05] 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 fulfil 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 [Foi05]. Indeed, for all $\sigma, \nu \in \mathfrak{S}$ set

$$\mathbf{F}_\sigma \prec \mathbf{F}_\nu := \sum_{\substack{\pi \,\in\, \sigma \sqcup\!\sqcup \nu \\ \pi_{|\pi|} = \sigma_{|\sigma|}}} \mathbf{F}_\pi, \tag{72}$$

$$\mathbf{F}_\sigma \succ \mathbf{F}_\nu := \sum_{\substack{\pi \,\in\, \sigma \sqcup\!\sqcup \nu \\ \pi_{|\pi|} = \nu_{|\nu|} + |\sigma|}} \mathbf{F}_\pi, \tag{73}$$

$$\Delta_\prec(\mathbf{F}_\sigma) := \sum_{\substack{\sigma = uv \\ \max(u) = \max(\sigma)}} \mathbf{F}_{\mathrm{std}(u)} \otimes \mathbf{F}_{\mathrm{std}(v)}, \tag{74}$$

$$\Delta_\succ(\mathbf{F}_\sigma) := \sum_{\substack{\sigma = uv \\ \max(v) = \max(\sigma)}} \mathbf{F}_{\mathrm{std}(u)} \otimes \mathbf{F}_{\mathrm{std}(v)}, \tag{75}$$

where $\max(u)$ is the maximal letter of the word u.

Proposition 6.12. *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 (25) spans a bidendriform sub-bialgebra of* **FQSym***, and 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 of **FQSym** spanned by the elements $\{\mathbf{P}_{\widehat{\sigma}}\}_{\widehat{\sigma} \in \mathfrak{S}/_\equiv}$. Indeed, the results of Foissy [Foi05] imply the rest of the proposition.

First, by Theorem 6.1, for all $\widehat{\sigma} \in \mathfrak{S}/_\equiv$, the coproduct $\Delta(\mathbf{P}_{\widehat{\sigma}})$, expanded on elements $\mathbf{F}_\nu \otimes \mathbf{F}_\pi$ can be expressed on elements $\mathbf{P}_{\widehat{\nu}} \otimes \mathbf{P}_{\widehat{\pi}}$. Let us show that is still the case for the coproduct $y := \Delta_\prec(\mathbf{P}_{\widehat{\sigma}})$. Assume that an element $\mathbf{F}_\nu \otimes \mathbf{F}_\pi$ arises in y. Then, by definition of $\Delta_\prec$, there is $\sigma \in \widehat{\sigma}$ such that $\sigma = uv$, $\nu = \mathrm{std}(u)$, $\pi = \mathrm{std}(v)$ and the maximal letter of uv is in the factor u. Let us shows that the element $\mathbf{F}_{\nu'} \otimes \mathbf{F}_\pi$ where $\nu' \equiv \nu$ also arises in y. Since $\equiv$ is compatible with the destandardization process, there is a word u' such that $\mathrm{eval}(u') = \mathrm{eval}(u)$ and $\mathrm{std}(u') = \nu'$. Thus, $u \equiv u'$ and since $\equiv$ is a congruence, $u'v \equiv uv$, showing that $\mathbf{F}_{\nu'} \otimes \mathbf{F}_\pi$ also arises in y. In the same way, if $\pi' \equiv \pi$, the element $\mathbf{F}_\nu \otimes \mathbf{F}_{\pi'}$ arises also in y. Hence, $\Delta_\prec$ is well-defined and in an analogous way, $\Delta_\succ$ also is.

Finally, by Theorem 6.1, for all $\widehat{\sigma}, \widehat{\nu} \in \mathfrak{S}/_{\equiv}$, the product $x := \mathbf{P}_{\widehat{\sigma}} \cdot \mathbf{P}_{\widehat{\nu}}$, expanded on elements $\mathbf{F}_\pi$ can be expressed on elements $\mathbf{P}_{\widehat{\pi}}$. That is still the case for the product $y := \mathbf{P}_{\widehat{\sigma}} \prec \mathbf{P}_{\widehat{\nu}}$ because by definition of $\prec$, all the elements $\mathbf{F}_\pi$ arising in x such that $\pi_{|\pi|} = \sigma_{|\sigma|}$ arise also in y. Since the equivalence classes of $\equiv$ only contain words ending by the same letter, y can be expressed on elements $\mathbf{P}_{\widehat{\pi}}$. In the same way, $\succ$ is well-defined. □

The equivalence relation $\equiv_{\mathrm{B}}$ satisfies the premises of Proposition 6.12 so that **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.

6.6. The dual Hopf algebra Baxter*.

6.6.1. *Description of* **Baxter***. Let $\{\mathbf{P}^\star_J\}_{J \in \mathcal{TBT}}$ be the dual basis of the basis $\{\mathbf{P}_J\}_{J \in \mathcal{TBT}}$. The Hopf algebra **Baxter***, dual of **Baxter**, is a quotient Hopf algebra of **FQSym***. More precisely,

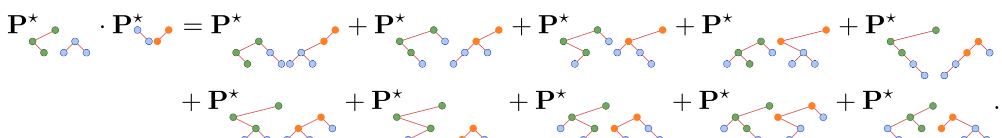
$$\mathbf{Baxter}^\star = \mathbf{FQSym}^\star / I \tag{76}$$

where I is the Hopf ideal of **FQSym*** spanned by the relations $\mathbf{F}^\star_\sigma = \mathbf{F}^\star_\nu$ whenever $\sigma \equiv_{\mathrm{B}} \nu$.

Let $\phi : \mathbf{FQSym}^\star \to \mathbf{Baxter}^\star$ be the canonical projection, mapping $\mathbf{F}^\star_\sigma$ on $\mathbf{P}^\star_J$ where $J := \mathbb{P}(\sigma)$. By definition, the product of **Baxter*** is

$$\mathbf{P}^\star_{J_0} \cdot \mathbf{P}^\star_{J_1} = \phi\left(\mathbf{F}^\star_\sigma \cdot \mathbf{F}^\star_\nu\right) \tag{77}$$

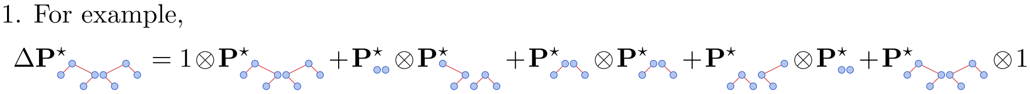
where σ and ν are any permutations such that $\mathbb{P}(\sigma) = J_0$ and $\mathbb{P}(\nu) = J_1$. By Proposition 6.3 and then by duality, the basis $\{\mathbf{P}^\star_J\}_{J \in \mathcal{TBT}}$ is a set-algebra basis of **Baxter***. Moreover, due to the fact that **Baxter*** is a quotient of **FQSym***, the number of terms occurring in a product $\mathbf{P}^\star_{J_0} \cdot \mathbf{P}^\star_{J_1}$ depends only of the number m (resp. n) of nodes of each binary tree of J_0 (resp. J_1) and is $\binom{m+n}{m}$. For example,

(78) 

In the same way, the coproduct of **Baxter*** is

$$\Delta(\mathbf{P}_J) = (\phi \otimes \phi)\left(\Delta\left(\mathbf{F}^\star_\sigma\right)\right) \tag{79}$$

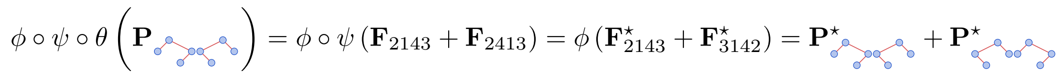
where σ is any permutation such that $\mathbb{P}(\sigma) = J$. By Proposition 6.3 and then by duality, the basis $\{\mathbf{P}^\star_J\}_{J \in \mathcal{TBT}}$ is a set-coalgebra basis of **Baxter***. Moreover, the number of terms occurring in a coproduct $\Delta\left(\mathbf{P}_J\right)$ depends only of the number n of nodes of each binary trees of J and is $n+1$. For example,

(80) 

6.6.2. *Non-triviality of the isomorphism between* **Baxter** *and* **Baxter***. Considering the map $\theta' : \mathbf{PBT} \hookrightarrow \mathbf{FQSym}$ that is the injection from **PBT** to **FQSym**, $\psi : \mathbf{FQSym} \leftrightarrow \mathbf{FQSym}^\star$ the isomorphism from **FQSym** to **FQSym*** defined by $\psi(\mathbf{F}_\sigma) = \mathbf{F}^\star_{\sigma^{-1}}$, and $\phi : \mathbf{FQSym}^\star \twoheadrightarrow \mathbf{PBT}^\star$ the surjection from **FQSym*** to **PBT***, it is well-known from [HNT05] that the map $\phi' \circ \psi \circ \theta'$ induces an isomorphism between **PBT** and **PBT***. Hence, since by Proposition 6.12, the Hopf algebras **Baxter** and **Baxter*** are isomorphic, it is natural to test if an analogous map is still an isomorphism between **Baxter** and **Baxter***. However, denoting by $\theta : \mathbf{Baxter} \hookrightarrow \mathbf{FQSym}$ the injection from **Baxter** to **FQSym** and $\phi : \mathbf{FQSym}^\star \twoheadrightarrow \mathbf{Baxter}^\star$ the surjection from **FQSym***

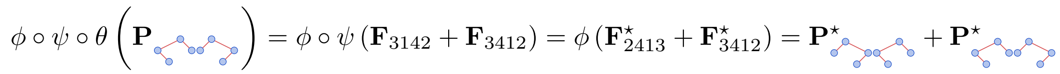
to **Baxter*** defined above, the map $\phi \circ \psi \circ \theta$: **Baxter** $\to$ **Baxter*** is not an isomorphism. Indeed:

(81)



$$\phi \circ \psi \circ \theta \left(\mathbf{P}_{\cdot}\right) = \phi \circ \psi \left(\mathbf{F}_{2143} + \mathbf{F}_{2413}\right) = \phi\left(\mathbf{F}^\star_{2143} + \mathbf{F}^\star_{3142}\right) = \mathbf{P}^\star_{\cdot} + \mathbf{P}^\star_{\cdot},$$

(82)



$$\phi \circ \psi \circ \theta \left(\mathbf{P}_{\cdot}\right) = \phi \circ \psi \left(\mathbf{F}_{3142} + \mathbf{F}_{3412}\right) = \phi\left(\mathbf{F}^\star_{2413} + \mathbf{F}^\star_{3412}\right) = \mathbf{P}^\star_{\cdot} + \mathbf{P}^\star_{\cdot},$$

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

6.6.3. *A pair of graded graphs in duality.* Following Fomin [Fom94], 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. $\mathbf{P}^\star_{J'}$) arises 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 13 (resp. Figure 14) shows the graded graph $G_{\mathbf{P}}$ (resp. $G_{\mathbf{P}^\star}$) restricted to vertices of order smaller than 5.

6.7. **Primitive and totally primitive elements.**

6.7.1. *Primitive elements.* Since the family $\{\mathbf{E}_J\}_{J \in C}$, where C is the set of connected pairs of twin binary trees are indecomposable elements of **Baxter**, its dual family $\{\mathbf{E}^\star_J\}_{J \in C}$ forms a basis of the Lie algebra of the primitive elements of **Baxter***. By Proposition 6.12, this Lie algebra is free.

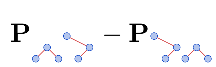
6.7.2. *Totally primitive elements.* Following [Foi05], the generating series $B_T(z)$ of the totally primitive elements of **Baxter** is

$$B_T(z) = \frac{B(z) - 1}{B(z)^2}. \tag{83}$$

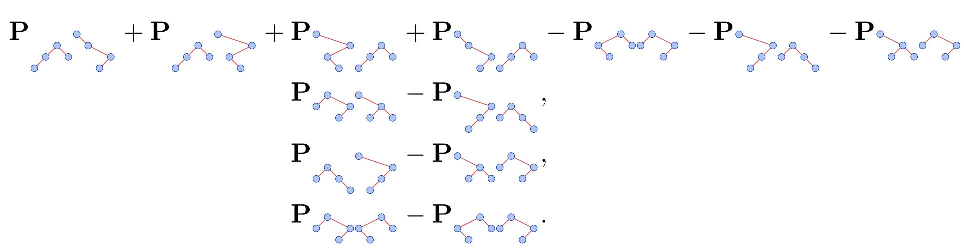
First dimensions of totally primitive elements of **Baxter** are 0, 1, 0, 1, 4, 19, 96, 511, 2832, 16215, 95374, 573837. Here follows a basis of the totally primitive elements of **Baxter** of order 1, 3 and 4:

(84)

(85)



(86)



References

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

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

[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.

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

[BS00] E. Babson and E. Steingrímsson. Generalized permutation patterns and a classification of the Mahonian statistic. *Séminaire Lotharingien de Combinatoire*, B44b, 2000.

[CS98] Ivan Chajda and Václav Snásel. Congruences in 0rdered Sets. *Mathematica Bohemica*, 123:95–100, 1998.

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

[DHNT08] G. Duchamp, F. Hivert, J.-C. Novelli, and J.-Y. Thibon. Noncommutative symmetric functions VII: free quasi-symmetric functions revisited. *arXiv:0809.4479v2 [math.CO]*, 2008.

[DHT02] G. Duchamp, F. Hivert, and J.-Y. Thibon. Noncommutative Symmetric Functions VI: Free Quasi-Symmetric Functions and Related Algebras. *International Journal of Algebra and Computation*, 12:671–717, 2002.

[Foi05] L. Foissy. Bidendriform bialgebras, trees, and free quasi-symmetric functions. *arXiv:math/0505207v1 [math.RA]*, 2005.

[Fom94] S. Fomin. Dualiy of Graded Graphs. *Journal of Algebraic Combinatorics*, 3:357–404, 1994.

[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.

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

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

[HNT05] F. Hivert, J.-C. Novelli, and J.-Y. Thibon. The Algebra of Binary Search Trees. *Theoretical Computer Science*, 339, Issue 1:129–165, 2005.

[Knu06] D. Knuth. *The Art of Computer Programming, Generating All Trees, History of Combinatorial Generation*, volume 4. 2006.

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

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

[LR10] S. Law and N. Reading. The Hopf algebra of diagonal rectangulations. *arXiv:1011.2118v1 [math.CO]*, 2010.

[LS81] A. Lascoux and M.-P. Schützenberger. Le monoïde plaxique. *Noncommutative Structures in Algebra and Geometric Combinatorics*, pages 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 Mathematics*, 217, Issues 1-3:315–336, 1998.

[NRT09] J.-C. Novelli, C. Reutenauer, and J.-Y. Thibon. Generalized descent patterns in permutations and associated Hopf algebras. *arXiv:0911.5615v1 [math.CO]*, 2009.

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

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

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

[S+10] W. A. Stein et al. *Sage Mathematics Software (Version 4.5.3)*. The Sage Development Team, 2010. 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

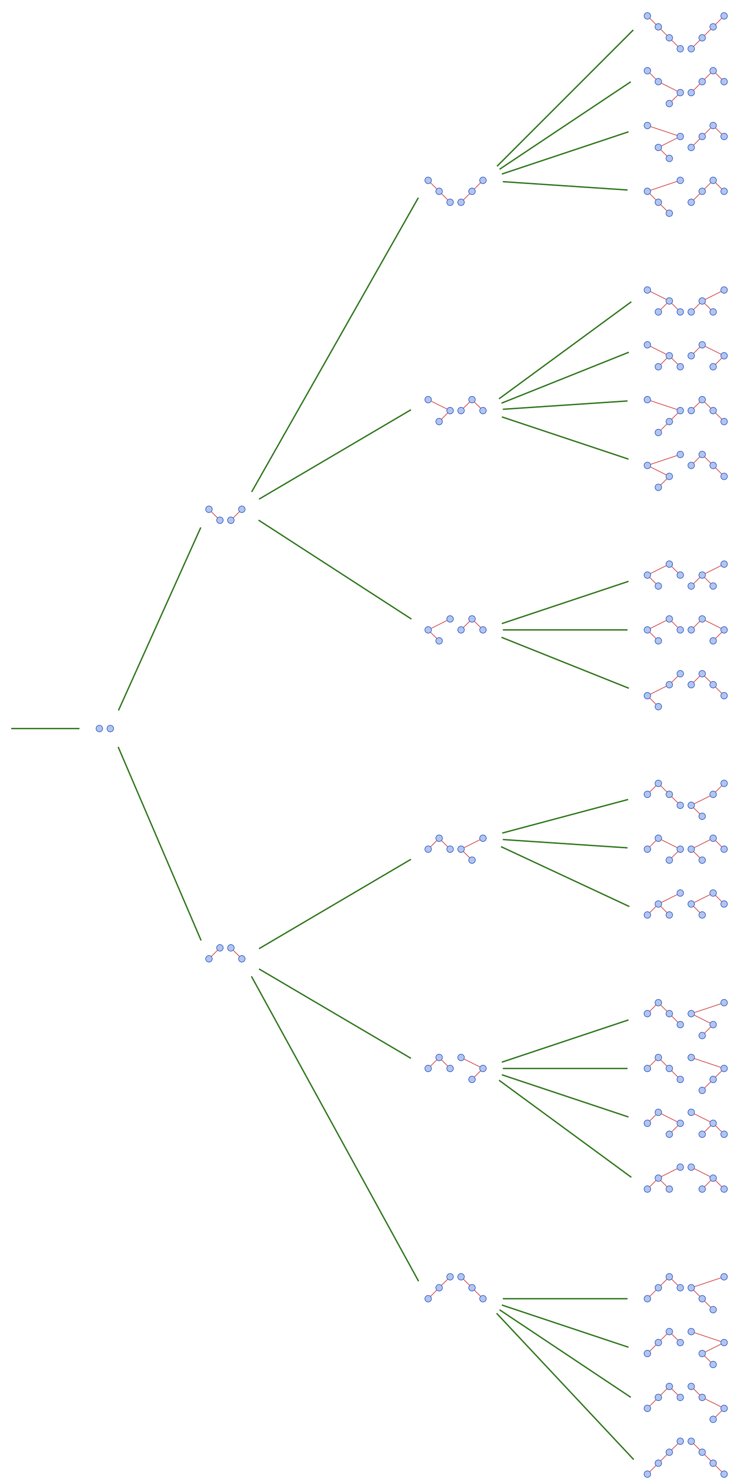


FIGURE 13. The graded graph $G_{\mathbf{P}}$ restricted to vertices of order smaller than 5.

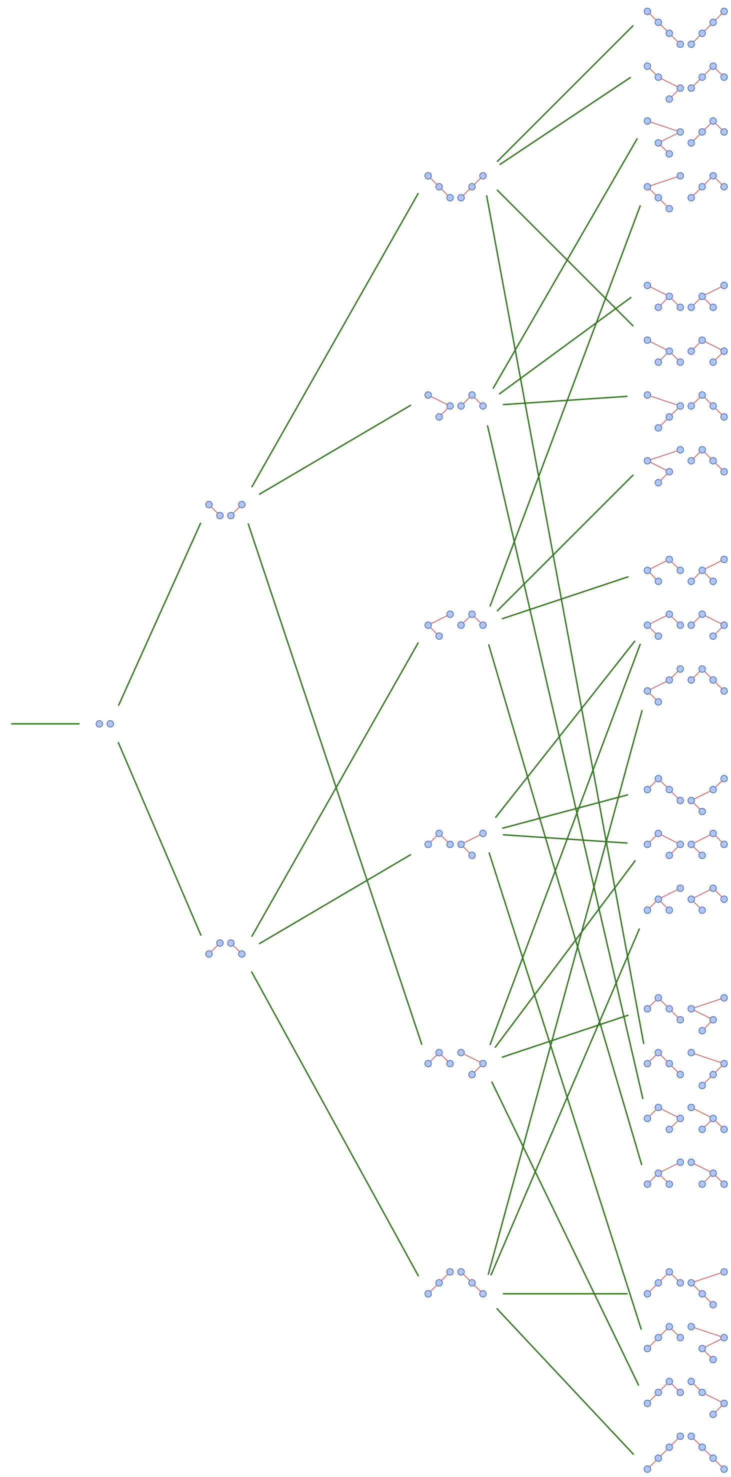


FIGURE 14. The graded graph $G_{\mathbf{P}^{\star}}$ restricted to vertices of order smaller than 5.