Formal Power Series and Algebraic Combinatorics
Séries Formelles et Combinatoire Algébrique
San Diego, California 2006

A new construction of the Loday-Ronco algebra

Maxime Rey

ABSTRACT. We provide a new construction of the Loday-Ronco algebra by realizing it in terms of non-commutative polynomials in infinitely many variables. This construction relies on a bijection between words and labeled binary trees which can be regarded as a kind of degenerate Robinson-Schensted correspondence and leads to a new Knuth type correspondence involving binary trees.

RÉSUMÉ. Nous donnons une nouvelle construction de l'algèbre de Loday et Ronco en termes de polynômes non-commutatifs en une infinité de variables. Cette construction repose sur une bijection entre les mots et les arbres binaires étiquetés qui permet de définir une correspondence de type Robinson-Schensted dégénérée et aboutit à la construction d'une nouvelle correspondence de type Knuth mettant en jeu les arbres binaires.

1. Introduction

We give a new construction of the Loday-Ronco algebra of the plane binary trees, also known as the free dendriform dialgebra on one generator (see [**8**]). We first use, in Section 3, the argument given in [**9**] on dendriform trialgebras in order to prove that the algebra of non-commutative polynomials in infinitely variables can be endowed with the structure of a dendriform dialgebra. We then state that the sub-dialgebra generated by the sum of the letters is free as a dendriform dialgebra. To prove this statement, we introduce in Section 4 a bijection between words and labeled binary trees, which leads to a degenerate kind of Robinson-Schensted correspondence, reminiscent of the degenerate correspondence with ribbons and quasi-ribbon diagrams in [**5**], and dual to the Sylvester Schensted Algorithm of [**2**] as explained in Section 5. This leads, in Section 6, to a new Knuth type correspondence between integer matrices and some pairs of labeled binary trees. In Section 7 we define a family of elements indexed by binary trees that permits to prove that our dendriform dialgebra on one generator is free, using a bijection between binary trees and its elements.

2. Preliminaries and Notations

In this paper, $\mathbb{K}$ stands for a field of any characteristic. Let $A = \{a_1, a_2, ...\}$ be a totally ordered (infinite) alphabet and denote by A^* the free monoid on A. The map $max : A^* \to A$ maps a word w to its greatest letter, according to the total order of the alphabet A. We denote by $Std(w)$ the standardized word of $w \in A^*$ defined as follows.

DEFINITION 2.1. Let $w = w_1 \cdots w_n \in A^*$ and $Std(w) = w'_1 \cdots w'_n$. Then, $\forall i, j \in [1, n]$ with $i \neq j$:

- if $w_i > w_j$ then $w'_i > w'_j$,
- if $w_i = w_j$ with $i > j$, then $w'_i > w'_j$,

such that $Std(w)$ is a permutation.

For example $Std(abcadbcaa) = 157296834$. For a word $w \in A^*$ and a subset B of A, $w|_B$ stands for the subword of w obtained by erasing the letters which are not in B. The evaluation of a word w is the vector $ev(w) = (|w|_{a_1}, |w|_{a_2}, ...)$.

2000 *Mathematics Subject Classification.* Primary 05-06; Secondary 05E99.

Key words and phrases. Algebraic combinatorics, Robinson-Schensted correspondence, Knuth correspondence, words, dendriform dialgebras.

We will denote by BT the set of all plane binary trees and LBT stands for the set of labeled plane binary trees. We denote by *Shape* the map that for a labeled binary tree forgets its labels and returns a binary tree of the same shape.

Let w be an element of A^* without repetition. Its decreasing tree $\mathcal{T}(w)$ is an element of LBT obtained as follows: the root is labeled by the greatest letter, n of w, and if $w = unv$, where u and v are words without repetition, the left subtree is $\mathcal{T}(u)$ and the right subtree is $\mathcal{T}(v)$. Moreover, we associate the empty tree to the empty word.

Let w be an element of A^*, we will denote by $\mathcal{B}(w)$ its associated binary search tree. It is obtained by reading w from *right-to-left*, each letter being inserted into a binary search tree in the following way: if the tree is empty, one creates a node labeled by the letter; otherwise, this letter is recursively inserted into the left (resp. right) subtree if it is smaller or equal than (resp. greater than) the root. Exemples will be given further.

A biletter on A is a pair $(a, b) \in A \times A$ which we will write $\begin{bmatrix} a \\ b \end{bmatrix}$ for convenience. A biword $\begin{bmatrix} u \\ v \end{bmatrix}$ on A^* is a concatenation of biletters $\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u_1 \\ v_1 \end{bmatrix}\begin{bmatrix} u_2 \\ v_2 \end{bmatrix}\ldots\begin{bmatrix} u_n \\ v_n \end{bmatrix}$. We denote by $\begin{bmatrix} u' \\ v' \end{bmatrix}$ the nondecreasing rearrangement of $\begin{bmatrix} u \\ v \end{bmatrix}$ for the lexicographic order with priority on the top row, and by $\begin{bmatrix} u'' \\ v'' \end{bmatrix}$ the nondecreasing rearrangement for the lexicographic order with priority on the bottom row. Let $\langle\rangle$ denote the linear map from $\mathbb{K}[[A, B]]$ to $\mathbb{K}\langle\langle A\rangle\rangle \otimes \mathbb{K}\langle\langle B\rangle\rangle$, defined by $\langle\begin{pmatrix} u \\ v \end{pmatrix}\rangle = u'' \otimes v'$, with $u \in A$ and $v \in B$.

3. The free dendriform dialgebra embedded in words

Following a suggestion of [**9**], we define the following operations on words.

Definition 3.1. For all $u, v \in A^+$,

$$u \leftharpoonup v := \begin{cases} uv & \text{if } max(u) > max(v) \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

$$u \rightharpoondown v := \begin{cases} uv & \text{if } max(u) \leq max(v) \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

Clearly, the usual operation of concatenation $\cdot$ on A^* can be written this way:

$$\cdot \;=\; \leftharpoonup + \rightharpoondown \quad . \tag{3.3}$$

Proposition 3.1. *$(\bigoplus_{n\geq 0} \mathbb{K}[A], \leftharpoonup, \rightharpoondown)$ is a dendriform dialgebra, in the sense of* [**7**].

Proof – Since (3.3) holds by definition, we only have to check the following three relations:

$$\begin{cases} (u \leftharpoonup v) \leftharpoonup w &= u \leftharpoonup (v \cdot w)\,, & (i) \\ (u \rightharpoondown v) \leftharpoonup w &= u \rightharpoondown (v \leftharpoonup w)\,, & (ii) \\ (u \cdot v) \rightharpoondown w &= u \rightharpoondown (v \rightharpoondown w)\,, & (iii) \end{cases} \tag{3.4}$$

whith $u, v, w \in A^+$. Notice first that for all these relations, there are only two possible values for each side, which are 0 and uvw.

(i) We first prove that

$$(u \leftharpoonup v) \leftharpoonup w = uvw \Longleftrightarrow u \leftharpoonup (v \cdot w) = uvw\,. \tag{3.5}$$

By definition, we have $(u \leftharpoonup v) \leftharpoonup w = uvw$ if and only if $max(u) > max(v)$ and $max(uv) > max(w)$. Since if $max(u) > max(v)$ then $max(uv) = max(u)$, we have a necessary and sufficient condition

$$max(u) > max(v) \;\wedge\; max(u) > max(w)\,. \tag{3.6}$$

On the right-hand side, we have $u \leftharpoonup (v \cdot w) = uvw$ if and only if

$$max(u) > max(vw)\,. \tag{3.7}$$

Since (3.6) and (3.7) are clearly equivalent, we have proved assertion (3.5).

(ii) We have $(u \rightharpoondown v) \leftharpoondown w = uvw$ if and only if

$$max(u) \leq max(v) \ \wedge \ max(w) < max(uv) \,. \tag{3.8}$$

But since $max(u) \leq max(v)$, assertion (3.8) is equivalent to

$$max(u) \leq max(v) \ \wedge \ max(w) < max(v) \,. \tag{3.9}$$

Moreover, $u \rightharpoondown (v \leftharpoondown w) = uvw$ if and only if

$$max(w) < max(v) \ \wedge \ max(u) \leq max(vw) \,,$$

which can be rewritten as

$$max(w) < max(v) \ \wedge \ max(u) \leq max(v) \,, \tag{3.10}$$

due to $max(w) < max(v)$. It results that $(u \rightharpoondown v) \leftharpoondown w = uvw$ if and only if $u \rightharpoondown (v \leftharpoondown w) = uvw$, by equivalence of assertions (3.9) and (3.10).

(iii) We have $u \rightharpoondown (v \rightharpoondown w) = uvw$ if and only if

$$max(u) \leq max(vw) \ \wedge \ max(v) \leq max(w) \,,$$

which can immediately be rewritten as

$$max(u) \leq max(w) \ \wedge \ max(v) \leq max(w) \,, \tag{3.11}$$

due to $max(v) \leq max(w)$. Moreover, we have $(u \cdot v) \rightharpoondown w = uvw$ if and only if $max(uv) \leq max(w)$, which is equivalent to (3.11). Hence it results $(u \cdot v) \rightharpoondown w = uvw$ if and only if $u \rightharpoondown (v \rightharpoondown w) = uvw$. ∎

Consider now the sub-dialgebra $\mathfrak{D}$ of $(\bigoplus_{n \geq 0} \mathbb{K}[A], \leftharpoondown, \rightharpoondown)$ generated by

$$P_\bullet := \sum_{a \in A} a \,.$$

There are two basis elements in the homogeneous component of degree 2 of $\mathfrak{D}$:

$$P_\bullet \leftharpoondown P_\bullet = \sum_{a<b} ba \,,$$

$$P_\bullet \rightharpoondown P_\bullet = \sum_{a \leq b} ab \,.$$

There are only five independent basis elements in the homogeneous component of degree 3 of $\mathfrak{D}$:

$$P_\bullet \leftharpoondown (P_\bullet \leftharpoondown P_\bullet) = \sum_{a<b<c} cba \,, \tag{3.12}$$

$$P_\bullet \rightharpoondown (P_\bullet \leftharpoondown P_\bullet) = \sum_{a<b;\ a' \leq b} a'ba \,, \tag{3.13}$$

$$P_\bullet \leftharpoondown (P_\bullet \rightharpoondown P_\bullet) = \sum_{a \leq b<c} cab \,, \tag{3.14}$$

$$(P_\bullet \leftharpoondown P_\bullet) \rightharpoondown P_\bullet = \sum_{a<b \leq c} bac \,, \tag{3.15}$$

$$(P_\bullet \rightharpoondown P_\bullet) \rightharpoondown P_\bullet = \sum_{a \leq b \leq c} abc \,, \tag{3.16}$$

since following equalities hold:

$$(P_\bullet \leftharpoondown P_\bullet) \leftharpoondown P_\bullet = \sum_{a<b;\ a'<b} baa' = (3.12) + (3.14) \,,$$

$$(P_\bullet \to P_\bullet) \leftarrow P_\bullet = \sum_{a \leq b;\ a' < b} aba' = (3.13)\,,$$

$$P_\bullet \to (P_\bullet \to P_\bullet) = \sum_{a \leq b;\ a' \leq b} a'ab = (3.15) + (3.16)\,,$$

which are exactly relations (3.4). We can now state our main result.

THEOREM 3.2. *$\mathfrak{D}$ is the free dendriform dialgebra on one generator.*

The remainder of this article will provide appropriate tools to prove this statement and exhibit some interesting remarks about them.

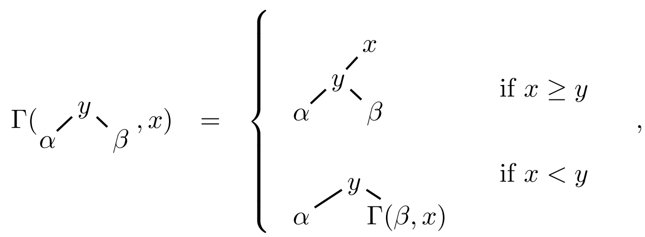
4. Algorithm Ψ

We first describe an algorithm Ψ that associates with a word a labelled plane binary tree. Then it will be possible to associate a plane binary tree with a word by considering only the shape of the labelled tree produced by algorithm Ψ. To this purpose we introduce a map

$$\Gamma : LBT \times A \longrightarrow LBT\,,$$

which can be recursively defined as follows:

(4.1)



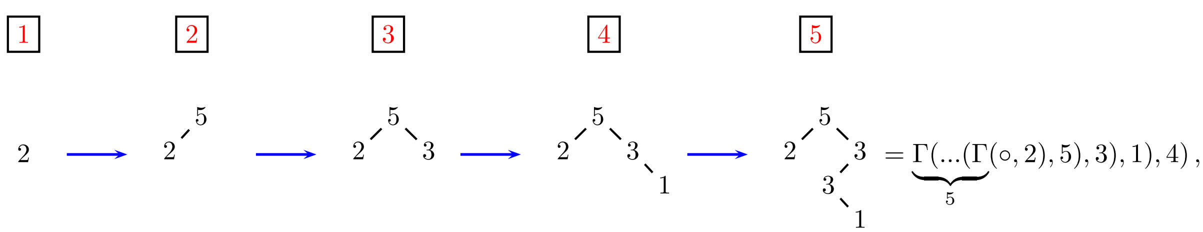
$$\Gamma(\circ, x) \quad = \quad x\,,$$

where ∘ stands for the empty tree and $\alpha, \beta \in LBT$.

DEFINITION 4.1. Consider the following function Ψ:

$$\Psi : \quad A^* \longrightarrow LBT\,, \qquad w = w_1 w_2 ... w_n \longmapsto \begin{cases} \Gamma(\Psi(w_1...w_{n-1}), w_n) & \text{if } n \geq 2\,, \\ \Gamma(\circ, w_1) & \text{if not}\,. \end{cases} \tag{4.2}$$

For example, using the alphabet $\mathbb{N}_{>0}$ with the natural order on integers, we apply Ψ to the word 25313 as follows:



starting with $\Gamma(\circ, 2)$.

PROPOSITION 4.1. *Let $w \in A^*$. Then, $Shape(\Psi(w)) = Shape(\Psi(Std(w)))$.*

Proof – We proceed by induction. The initial case, considering a word of size 1, is obvious since there is only one tree of size 1. Assuming that this property is satisfied for words of length $n-1$, we check that it is true for words of length n due to Definition 2.1 and the inductive definition (4.1). ∎

We recall that $\mathcal{T}$ stands for the decreasing tree algorithm which associates to a permutation its decreasing tree.

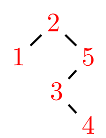
PROPOSITION 4.2. $\forall \sigma \in \mathfrak{S}, \Psi(\sigma) = \mathcal{T}(\sigma)$.

Proof – We proceed by induction. This property is obvious for the permutation of size 1. Assume that this property is satisfied for permutations of all sizes smaller than n and $\sigma = \sigma_1 \cdot max(\sigma) \cdot \sigma_2$ is a permutation of size n. By the inductive Definition 4.1 it is clear that the root of $\Psi(\sigma)$ is labeled by $max(\sigma)$, and that the left subtree of the root of $\Psi(\sigma)$ will be $\mathcal{T}(\sigma_1)$ by the inductive hypothesis, and similarly the right subtree of the root of $\Psi(\sigma)$ will be $\mathcal{T}(\sigma_2)$. ∎

Hence algorithm Ψ is clearly a generalization on words of the well-known decreasing tree algorithm for permutations. From Propositions 4.2 and 4.1 the following result is immediate.

PROPOSITION 4.3. *The algorithm Ψ is injective.*

Let $w \in A^*$, we now consider the labeled binary tree having the same shape as $\Psi(w)$ and for which the label of each node is the step of its insertion in the tree $\Psi(w)$. We will denote it by $\psi(w)$. From the previous calculation of $\Psi(25313)$ we get the following tree:



.

We notice that $\psi(25313)$ is the binary search tree of $Std(25313)^{-1} = 41352$. We develop a new Schensted-like correspondence and a new Schensted-Knuth-like correspondence from this consideration.

5. The Co-Sylvester Schensted Algorithm ($CSSA$)

The Sylvester Schensted Algorithm (SSA) has been introduced in [**2**]. From Algorithms Ψ and ψ we give a dual correspondence of SSA.

DEFINITION 5.1. We note $CSSA$ the *Co-Sylvester Schensted Algorithm* which sends a word $w \in A^*$ to the pair

$$(\Psi(w), \psi(w)) \,.$$

Algorithm $CSSB$ sends this pair to the word obtained by reading the labels of $\Psi(w)$ in the order of the corresponding labels in $\psi(w)$.

We know that

LEMMA 5.2 ([**2**]). *Let w be a word and $\sigma = Std(w)$. Then*

$$Shape(\mathcal{B}(w)) = Shape(\mathcal{B}(\sigma)) = Shape(\mathcal{T}(\sigma^{-1}))\,.$$

We generalize this result in terms of biwords.

LEMMA 5.3. *For any biword* $\begin{bmatrix} u \\ v \end{bmatrix}$ *, $Shape(\Psi(v')) = Shape(\mathcal{B}(u''))$.*

Proof – Using notations of Section 2, the standardization being compatible under transposition of two letters it follows that $Std(v)' = Std(v')$ and $Std(u)'' = Std(u'')$. It is well-known that the inverse of a permutation $\begin{bmatrix} Id \\ \sigma \end{bmatrix}$ is the biword $\begin{bmatrix} \sigma^{-1} \\ \sigma^{\uparrow} = Id \end{bmatrix}$ where $w^{\uparrow}$ denotes the nondecreasing rearrangement of a word w. Then, since $Std(v)' = (Std(u)'')^{-1}$, we have

$$Std(v') = (Std(u''))^{-1}\,.$$

Hence, using Lemma 5.2 and Proposition 4.1 we obtain these equivalences:

$$Shape(\Psi(v')) = Shape(\mathcal{T}(Std(v'))) = Shape(\mathcal{B}((Std(v'))^{-1})) = Shape(\mathcal{B}(u''))\,.$$

■

This allows us to state an analogous of the Schützenberger Theorem on tableaux for binary trees.

Theorem 5.4. *$\forall \sigma \in \mathfrak{S}$, $\psi(\sigma) = \mathcal{B}(\sigma^{-1})$.*

Proof – By definition of ψ, $\psi(\sigma)$ have the same shape as $\Psi(\sigma) = \mathcal{T}(\sigma)$. Moreover, by induction it is also clear that $\psi(\sigma)$ is a binary search tree. Indeed, assuming the inductive hypothesis for trees with $k-1$ nodes, the k-th node is, by definition of the algorithm ψ, labeled by k whereas the $k-1$ remaining nodes are labeled by elements of $[1,(k-1)]$. Moreover, by recursive definition (4.1) two cases arise:

- The k-th node is inserted at the root and then it is still a binary search tree since the $k-1$ remaining nodes are in its left subtree.
- The k-th node is inserted somewhere in the right subtree, and so it is still a binary search tree since it is greater than the root, and by inductive hypothesis.

Nevertheless, in general there is not a single permutation σ' such that $\mathcal{B}(\sigma') = T$, with T a given binary search tree (see [**2**] for the exact description of such sets of permutations). But it is clear, by definition of ψ, that at each step k of the insertion algorithm $CSSA$, $\Psi(\sigma_1\sigma_2...\sigma_k)$ and $\psi(\sigma_1\sigma_2...\sigma_k)$ have the same shape. Moreover setting

$$\begin{bmatrix} u(k)' \\ v(k)' \end{bmatrix} = \begin{bmatrix} 1 & 2 & \cdots & k \\ \sigma_1 & \sigma_2 & \cdots & \sigma_k \end{bmatrix} \qquad \text{and} \qquad \begin{bmatrix} u(k)'' \\ v(k)'' \end{bmatrix} = \begin{bmatrix} \sigma^{-1}|_{[1,k]} \\ 1\,2\cdots k \end{bmatrix}\ ,$$

from Lemma 5.3 we have that at each step of the insertion algorithm, $\Psi(\sigma_1\sigma_2\cdots\sigma_k)$ and $\mathcal{B}(\sigma^{-1}|_{[1,k]})$ have the same shape. Hence, at the last step, $\psi(\sigma) = \mathcal{B}(\sigma^{-1})$. ■

Proposition 5.1. *$\forall\, w \in A^*, \psi(w) = \psi(Std(w))$.*

Proof – From Definition 2.1 and Proposition 4.1, it is immediate since at each step k, $\Psi(w_1\cdots w_k)$ and $\Psi(Std(w_1\cdots w_k))$ have the same shape. ■

Hence from Theorem 5.4 and Proposition 5.1 we immediately obtain the following result on words.

Corollary 5.5. *$\forall\, w \in A^*$, $\psi(w) = \mathcal{B}(Std(w)^{-1})$.*

From Theorem 5.4 and Proprosition 4.2 it is straightforward that $CSSA$ is the dual Schensted-like correspondence of SSA of [**2**] in the following sense.

Proposition 5.2. *$\forall \sigma \in \mathfrak{S}$,*

$$CSSA(\sigma) = (\Psi(\sigma), \mathcal{B}(\sigma^{-1})) \Longleftrightarrow SSA(\sigma^{-1}) = (\mathcal{B}(\sigma^{-1}), \Psi(\sigma))\,.$$

It is interesting to notice that these two correspondences look quite similar to the two Robinson-Schensted type correspondences on ribbons and quasi-ribbons, introduced in [**5**].

6. The Sylvester Schensted-Knuth correspondence

We first recall that there is an easy bijection between integer matrices and commutative biwords on A^* which consists to repeat m_{ij} times the biletter $\begin{pmatrix} \mathrm{i} \\ \mathrm{j} \end{pmatrix}$ for a matrix $M = (m_{ij})_{(i,j)\in[1,n]\times[1,m]}$ of dimensions $n \times m$. For example commutative biwords $\begin{pmatrix} 1111222333 \\ 1113123133 \end{pmatrix}$ and $\begin{pmatrix} 1112321233 \\ 1111123333 \end{pmatrix}$ (which are equal) have the same corresponding matrix which is :

$$\begin{pmatrix} 3 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 2 \end{pmatrix}.$$

Definition 6.1. Let $\begin{pmatrix} u \\ v \end{pmatrix}$ be a commutative biword and $\begin{bmatrix} u' \\ v' \end{bmatrix}$, $\begin{bmatrix} u'' \\ v'' \end{bmatrix}$ be the two biwords associated with $\begin{bmatrix} u \\ v \end{bmatrix}$ as explained in Section 2 . We note κ_S the *Sylvester Schensted-Knuth correspondence* defined as follows:

$$\kappa_S \begin{pmatrix} u \\ v \end{pmatrix} := (\Psi(v'), \mathcal{B}(u'')) .$$

This definition holds since for any biletter permutation of $\begin{bmatrix} u \\ v \end{bmatrix}$, v' and u'' remain the same. Moreover by Lemma 5.3 the following Proposition is straightforward.

Proposition 6.1. *For any commutative biword* $\begin{pmatrix} u \\ v \end{pmatrix}$, $Shape(\Psi(v')) = Shape(\mathcal{B}(u''))$.

We notice that $CSSA$ is recovered by encoding a word w by $\begin{pmatrix} 1 & 2 & \cdots & m \\ w_1 & w_2 & \cdots & w_m \end{pmatrix}$.

In order to prove that κ_S is a bijection, we proceed as Lascoux, Leclerc and Thibon did in [**6**] for the usual Knuth correspondence [**4**].

Theorem 6.2. *The algorithm* κ_S *is a bijection.*

Proof – Using again arguments of the proof of Lemma 5.3 we obtain $\mathcal{B}(Std(u'')) = \mathcal{B}(Std(v')^{-1})$. Then, applying Theorem 5.4 we have that $\mathcal{B}(Std(v')^{-1}) = \psi(Std(v'))$ and from Corollary 5.5 we get:

$$\mathcal{B}(Std(u'')) = \psi(v') .$$

This means that $\mathcal{B}(u'')$ is the unique binary search tree of evaluation $ev(u'')$ such that $\mathcal{B}(Std(u'')) = \psi(v')$. ■

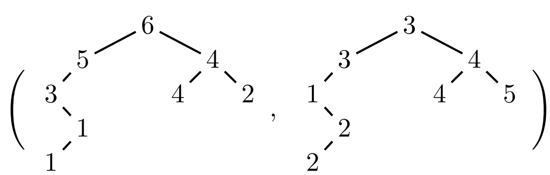
An easy remark is the following:

$$\kappa_S \begin{pmatrix} u \\ v \end{pmatrix} = (\Psi(v'), \mathcal{B}(u'')) \Longleftrightarrow \kappa_S \begin{pmatrix} v \\ u \end{pmatrix} = (\Psi(u''), \mathcal{B}(v')) ,$$

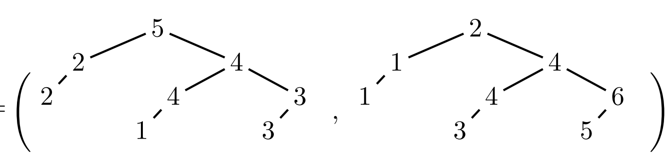
which generalizes results of Section 5. Nevertheless the symmetry of the usual Knuth correspondence is broken for P and Q symbols. At last, we give a full example of our construction.

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 21335424 \\ 13652414 \end{bmatrix}, \quad \begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} 12233445 \\ 31156442 \end{bmatrix} \quad \begin{bmatrix} u'' \\ v'' \end{bmatrix} = \begin{bmatrix} 22514433 \\ 11234456 \end{bmatrix},$$

$(\Psi(v'), \mathcal{B}(u''))$ =

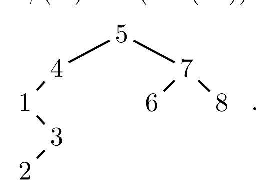


$(\Psi(u''), \mathcal{B}(v'))$ =



Since $Std(22514433) = 23816745$, we can check that $\psi(v') = \mathcal{B}(Std(u''))$:

$\psi(31156442) = \mathcal{B}(23816745) =$



This bijection leads to a Cauchy formula type for binary trees.

Proposition 6.2. *Let A and B be two non-commutative alphabets, such that A and B are commuting one with the other and words on alphabet B are quotiented by the sylvester congruence (see* [**2**]*). Then,*

$$\left\langle \overrightarrow{\prod_{a\in A,\, b\in B}} \frac{1}{1-ab} \right\rangle = \sum_{T\in BT} P_T(A)\otimes Q_T(B)\,,$$

where $(P_T)_{T\in BT}$ comes from Definition 7.1 and $(Q_T)_{T\in BT}$ is its dual basis introduced in [**2**].

Proof – We recall that the sylvester monoid introduced in [**2**] is the monoid such that two words having the same shape through algorithm $\mathcal{B}$ are equal. This proof needs the Free Cauchy identity mentionned in [**3**] and to be fully introduced in [**1**]:

$$\left\langle \overrightarrow{\prod_{a\in A,\, b\in B}} \frac{1}{1-ab} \right\rangle = \sum_{Std(v)=Std(u)^{-1}} u\otimes v\,. \tag{6.1}$$

Since v is an element of the sylvester monoid and by definition of $(Q_T)_{T\in BT}$ (see [**2**]), by Theorem 6.2 and by Proposition 6.1, right-hand side of Equation (6.1) can be rewritten as

$$\sum_{T\in BT}\left(\sum_{Shape(\Psi(w))=T} w\right)\otimes Q_T\,.$$

Using Definition 7.1 we obtain the desired equality. ■

7. Back to dendriform structure

Definition 7.1. Let $T\in BT$. We define

$$P_T := \sum_{w;\ Shape(\Psi(w))=T} w\,. \tag{7.1}$$

As a special case we recover:

$$P_\bullet = \sum_{a\in A} a\,. \tag{7.2}$$

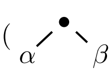
We now provide an algorithm that associates a plane binary tree to an element of the dendriform algebra $\mathfrak{D}$.

Definition 7.2. We consider

$$\Phi : BT \longrightarrow (\mathbb{K}[A], \leftharpoonup, \rightharpoondown)\,,$$

whose recursive definition is the following:

(7.3)

$\Phi($  $) = (\Phi(\alpha) \rightharpoondown (P_\bullet \leftharpoonup \Phi(\beta)))\,,$ (*i*)

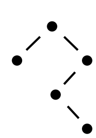
$\Phi($  $) = (\Phi(\alpha) \rightharpoondown P_\bullet)\,,$ (*ii*)

$\Phi($  $) = (P_\bullet \leftharpoonup \Phi(\beta))\,,$ (*iii*)

$\Phi(\bullet) = P_\bullet\,,$ (*iv*)

where $\alpha, \beta \in BT$.

For example:

$\Phi($  $) = (P_\bullet \rightharpoondown (P_\bullet \leftharpoonup ((P_\bullet \leftharpoonup P_\bullet) \rightharpoondown P_\bullet)))\,.$

LEMMA 7.3. *Let $T \in BT$. Then,* $P_T = \Phi(T)$ *.*

Proof – We proceed by induction. The initial case is immediate by (*iv*) of Definition 7.2. Assume that this property is satisfied for trees of size $n-1$. We consider a tree T of size n, with $n \geq 2$. Three kinds of trees are possible, according to wether their roots have only a non-empty right subtree or only a non-empty left subtree or finally have both left and right subtrees are non-empty. These cases correspond to $(i), (ii)$ and (iii) of (7.2).

(*i*) By Definition 7.1 this means that for all words appearing in the sum P_T, the $|\alpha| + 1$ letter is greater or equal than its $|\alpha|$ first letters and greater to its $|\beta|$ last letters. This is the exact meaning of the right-hand side of (*i*) using (3.1), (3.2), remembering relation (*ii*) (associativity) of (3.4) and assuming inductive hypothesis.

(*ii*) By Definition 7.1 this means that for all words appearing in the sum P_T, their last letter is greater or equal than all others letters appearing in it. Then, by (3.2) and by induction hypothesis we have proved this case.

(*iii*) By Definition 7.1 this means that for all words appearing in the sum P_T, their first letter is greater than all others letters appearing in it. Hence by (3.1) and by induction hypothesis we have proved this case.

■

We now are able to provide the proof of Theorem 3.2:

Proof of Theorem 3.2 – Since the Loday-Ronco algebra of plane binary trees is the free dendriform algebra on one generator, we only have to check that the Hilbert serie of this subalgebra $\mathfrak{D}$ generated by $P_\bullet$ is counted by Catalan numbers. To this purpose we consider the familly of $(P_T)_{T\in BT}$. By Lemma 7.3 they are clearly elements of $\mathfrak{D}$. Moreover the intersection of any pairs of them is always empty, by construction. Then $(P_T)_{T\in BT}$ are linearly independent. ■

References

[1] G. Duchamp, F. Hivert, J.-C. Novelli, J.-Y. Thibon, Noncommutative symmetric functions VII (under construction).
[2] F. Hivert, J.-C. Novelli, J.-Y. Thibon, The algebra of binary search trees, Theoret. Computer Sci. 339 (2005), 129–165.
[3] F. Hivert, J.-C. Novelli, J.-Y. Thibon, Un analogue du monoïde plaxique pour les arbres binaires de recherche, C.R. Acad. Sci. Paris Sr I Math. 332 (2002) 577-580.
[4] D. Knuth, "Permutations, matrices, and generalized Young tableaux", Pacific J. Math. 34 (1970), 709-727.
[5] D. Krob and J.-Y. Thibon, Noncommutative symmetric functions IV: Quantum linear groups and Hecke algebras at q = 0, J. Alg. Comb. 6 (1997), 339–376.
[6] A. Lascoux, B. Leclerc, and J.-Y. Thibon, The Plactic Monoid, in: Algebraic Combinatorics on Words, M. Lothaire, Univ. Press Cambridge (2001).
[7] J.-L. Loday, Dialgebras and related operads, Springer Lecture Notes in Mathematics 1763 (2001), 7-66.
[8] J.-L. Loday, M. O. Ronco, Hopf algebra of the planar binary trees, Adv. Math. 139 (1998), no. 2, 293–309.
[9] J.-C Novelli and J.-Y. Thibon, Construction of dendriform trialgebras, preprint math.CO/0510218.

INSTITUT GASPARD MONGE, UNIVERSITÉ DE MARNE-LA-VALLÉE, 5 BOULEVARD DESCARTES, CHAMPS-SUR-MARNE, 77454 MARNE-LA-VALLÉE CEDEX 2, FRANCE

E-mail address: rey@univ-mlv.fr