A lattice of combinatorial Hopf algebras: Binary trees with multiplicities

Jean-Baptiste Priez†

LRI, Bât. 650, Université Paris-Sud 11, 91405 Orsay, France

Abstract. In a first part, we formalize the construction of *combinatorial Hopf algebras* from plactic-like monoids using *polynomial realizations*. Thank to this construction we reveal a lattice structure on those combinatorial Hopf algebras. As an application, we construct a new combinatorial Hopf algebra on binary trees with multiplicities and use it to prove a *hook length formula* for those trees.

Résumé. Dans une première partie, nous formalisons la construction d'*algèbres de Hopf combinatoires* à partir d'une *réalisation polynomiale* et de monoïdes de type *monoïde plaxique*. Grâce à cette construction, nous mettons à jour une structure de treillis sur ces algèbres de Hopf combinatoires. Comme application, nous construisons une nouvelle algèbre de Hopf sur des arbres binaires à multiplicités et on l'utilise pour démontrer une *formule des équerres* sur ces arbres.

Keywords: Combinatorial Hopf algebras, monoids, polynomial realization, hook length formula, generating series, binary trees

1 Introduction

In the past decade a large amount of work in algebraic combinatorics has been done around combinatorial Hopf algebras. Many have been constructed on various combinatorial objects such as *partitions* (symmetric functions [Mac95]), *compositions* (**NCSF** [GKL+94, MR95]), *permutations* (**FQSym** [MR95, DHT02]), *set-partitions* (**WQSym** [Hiv99]), *binary trees* (**PBT** or the LODAY-RONCO Hopf algebra [LR98, HNT05]), or *parking functions* (**PQSym** [NT04, NT07]). A powerful method to construct those algebras, called *polynomial realization*, is to construct the Hopf algebra as a sub algebra of a free algebra of polynomials (commutative or not) admitting certain symmetries. Beside the contruction of *Combinatorial Hopf algebra*, several recent papers investigate toward the formalization of combinatorial applications such as *hook formulas*, or seek some structure in this large zoo.

This extended abstract, reports on a work in progress which proposes to formalize the construction of Hopf algebras by polynomial realizations: starting with one of the three Hopf algebras **FQSym**, **WQSym** or **PQSym** realized in a *free algebra*, we impose some relations on the variables. Under some simple hypotheses, the result is again a Hopf algebra (Theorem 1). Two important examples are already known,

† `jean-baptiste.priez@lri.fr`

namely the POIRIER-REUTENAUER algebra of tableaux (**FSym** [PR95, DHT02]) obtained from the *plactic monoid* [LS81] and the *planar binary tree algebra* of LODAY-RONCO obtained from the *sylvester monoid* [LR98, HNT05].

We further observe that the construction transports the lattice structure on monoids to a lattice structure on those Hopf algebras (Theorem 2). This structure was used implicitely by GIRAUDO for constructing the Baxter Hopf algebra from the Baxter monoid as the infimum of the *sylvester monoid* and its image under SCHÜTZENBERGER involution. The supremum of those two monoids is known as the *hypoplactic monoid* which gives the algebra of *quasi symmetric functions* [Nov00].

As an application (Section 5) we take the supremum of the *sylvester monoid* and the *stalactic monoid* of [HNT08a]. The result is a monoid on binary search trees with multiplicities leading to a Hopf algebra on binary trees with multiplicities. Interestingly, there is a *hook length formula* for those trees (Theorem 3) and we prove it using the Hopf algebra as generating series.

Acknowledgements

I would like to thank Florent HIVERT and Nicolas M. THIÉRY for their patience and advice during the preparation of this manuscript. Several examples in this paper were computed using the open-source mathematical software `Sage` [S+12] and its extension: `Sage-Combinat` [SCc12]. The implementation of several Hopf algebras is available at `http://code.google.com/p/sage-hopf-algebra/` and will be available on `Sage-Combinat` soon and later in `Sage`. That code is due to Rémi MAURICE and I.

2 Background

In this section, we introduce some notations and three specific maps from words to words: *standardization*, *packing*, and *parkization*. These will be the main tool for polynomial realizations of Hopf algebras.

2.1 Lattice structure on Congruences

The free monoid $\mathfrak{A}^*$ on an alphabet $\mathfrak{A}$ is the set of words with concatenation as multiplication. We denote by 1 the empty word. Recall that a *monoid congruence* is an equivalence relation $\equiv$ which is left and right compatible with the product; in other words, for any monoid elements a, b, c, d, if $a \equiv b$ and $c \equiv d$ then $ac \equiv bd$. Starting with two congruences on can build two new congruences:

- the *union* $\sim \vee \approx$ of $\sim$ and $\approx$ is the transitive closure of the union $\sim$ and $\approx$; that is $u \equiv v$ if there exists $u = u_0, \dots, u_k = v$ such that for any i, $u_i \sim u_{i+1}$ or $u_i \approx u_{i+1}$. It is the smallest congruence containing both $\sim$ and $\approx$;
- the *intersection* $\sim \wedge \approx$ of $\sim$ and $\approx$ is defined as the relation $\equiv$ with $u \equiv v$ if $u \sim v$ and $u \approx v$.

2.2 Some φ-maps

Throughout this paper we construct Hopf algebras from the equivalence classes of words given by the fibers of some map φ from the free monoid to itself. Our main examples are *standardization* and *packing* functions which can be defined for any totally ordered alphabet $\mathfrak{A}$. We could easily extend these following properties to *parkization* [NT04, NT07] *if the alphabet* $\mathfrak{A}$ *is well-ordered* (any element has a successor).

In the following, *we suppose that* $\mathfrak{A}$ *is an totally ordered infinite alphabet*. Most of the time we use $\mathfrak{A} = \mathbb{N}^{>0}$ for simplicity. For w in $\mathfrak{A}^*$, we denote by $part(w)$ the ordered set partition of positions of w

letters obtained as follows: for each letter $l \in \mathfrak{A}$ appearing in w, there is a part containing the positions of each occurrence of l in w; the parts are ordered using the order on the alphabet $\mathfrak{A}$. For example: $part(13231) = [\{1,5\},\{3\},\{2,4\}]$ and $part(1112) = [\{1,2,3\},\{4\}]$.

Standardization std computes the lexicographically smallest word w which has same length and same set of inversions. This map is used in the realization of the Hopf algebra **FQSym** of permutations [DHT02, MR95]. The image $std(\mathfrak{A}^*)$ is identified with the set $\mathfrak{S}$ of all permutations.

Algorithm 1: Standardization std

Data: $w = (a_1, \ldots, a_k)$ a word of $\mathfrak{A}^k$
Result: $\sigma = (\sigma_1, \ldots, \sigma_k) \in \mathfrak{S}_k \subseteq \mathfrak{A}^k$
$osp \leftarrow part(w); i \leftarrow 1$
forall the $set \in osp$ **do**
  forall the $p \in set$ **do**
    $\sigma_p \leftarrow i$
    increment(i)
return σ

Some examples:

w	$std(w)$
$(7,2,14,3,7)$	$(3,1,5,2,4)$
$(23,14,5,92)$	$(3,2,1,4)$
$(4,2,1,3,5)$	$(4,2,1,3,5)$
$(1,5,1,1,5,5)$	$(1,4,2,3,5,6)$

Packing $pack$ computes the lexicographically smallest word w which has same ordered set partitions. This map is used in the realization of the Hopf algebra **WQSym** of ordered set partition or packed words [Hiv99]. We identify $tass(\mathfrak{A}^*)$ with the collection of ordered set partitions.

Algorithm 2: Packing $pack$

Data: $w = (a_1, \ldots, a_k)$ a word of $\mathfrak{A}^*$
Result: $c = (c_1, \ldots, c_k)$
$osp \leftarrow part(w); i \leftarrow 1$
forall the $set \in osp$ **do**
  forall the $p \in set$ **do** $c_p \leftarrow i$
  increment(i)
return c

Some examples:

w	$tass(w)$
$(3,13,3,2,13)$	$(2,3,2,1,3)$
$(2,2,2,5,8,2)$	$(1,1,1,2,3,1)$
$(4,2,1,3,5)$	$(4,2,1,3,5)$
$(2,3,1,1,2)$	$(2,3,1,1,2)$

Those maps are used to realize some Hopf algebras like **FQSym**, **WQSym**, or **PQSym**. For each such map φ we say that a word w is *canonical* if $\varphi(w) = w$. For example, 1423 is canonical for std and 1121 is canonical for $pack$. The set of canonical words for the *standardization* function is the set of permutations set; for the *packing* function it is the set of *packed words*. The set $\varphi(\mathfrak{A}^*)$ of canonical words is denoted by can_φ. We call these maps the *φ-maps*.

3 Polynomial realizations and Hopf algebras

In this section we describe how, from a φ-map, one can construct a Hopf algebra such as **FQSym**, **WQSym**, or **PQSym**, using two tricks: *polynomial realization* and *alphabet doubling*. *Polynomial realizations* are a powerful trick to construct algebras as sub-algebras of a free algebra by manipulating some polynoms having certain symmetries. Futhermore the *alphabet doubling trick* defines a graded algebra morphism on a free algebra which endows it with a compatible coproduct, that is a Hopf algebra structure.

3.1 φ-polynomial realization

The notion of polynomial realizations has been introduced and implicitly used in many articles of the "*phalanstère de Marne-la-Vallée*" (France). See *e.g.* [DHT02, NT06, HNT08a]. In the following, we call *alphabet* $\mathfrak{A}$ an infinite and totally ordered (when appropriate, we assume furthermore that the total order admits a successor function) set of symbols all of which are of weight 1. By an abuse of language, we call the free algebra the graded algebra infinite **but finite degree** sum of words.

Definition 1 (Polynomial realization): *Let $\mathcal{A} := \oplus_{n\geqslant 0}\mathcal{A}_n$ be a graded algebra. A* polynomial realization *r of $\mathcal{A}$ is a map which associates to each alphabet $\mathfrak{A}$ an injective graded algebra morphism $r_{\mathfrak{A}}$ from $\mathcal{A}$ to the free non-commutative algebra $\mathbb{K}\langle\mathfrak{A}\rangle$ such that, if $\mathfrak{A} \subset \mathfrak{B}$, then for all $x \in \mathcal{A}$ one has $r_{\mathfrak{A}}(x) = r_{\mathfrak{B}}(x)/_{\mathfrak{A}}$, where $r_{\mathfrak{B}}(x)/_{\mathfrak{A}}$ is the sub linear combination obtained from $r_{\mathfrak{B}}(x)$ by keeping only those words in $\mathfrak{A}^*$.*

When the realization is clear from the context we write $\mathcal{A}(\mathfrak{A}) := r_{\mathfrak{A}}(\mathcal{A})$ for short.

For a given φ, we consider the subspace $\mathcal{A}_\varphi$ admitting the basis $(m_u)_{u\in can_\varphi}$ defined on $\mathcal{A}_\varphi(\mathfrak{A})$:

$$r_{\mathfrak{A},\varphi}(m_u) = \sum_{w\in\mathfrak{A}^*;\varphi(w)=u} w\,. \tag{1}$$

The result does not depend on the alphabet. For $\varphi = std$, $pack$ or $park$ the linear span of $(m_u)_{u\in can_\varphi}$ is a sub-algebra of $\mathbb{K}\langle\mathfrak{A}\rangle$.

Example 1 (Realization of FQSym): If $\varphi = std$ then can_φ is in fact the set of permutations and $\mathfrak{A}_\varphi$ is the permutations Hopf algebra **FQSym** [DHT02, MR95]. It is realized by the std-polynomial realization in $\mathbb{K}\langle\mathfrak{A}\rangle$: let $\mathbb{G}_\sigma(\mathfrak{A}) := r_{\mathfrak{A},std}(\mathbb{G}_\sigma)$ such that, for example

$$\mathbb{G}_{132}(\mathbb{N}^*) = 121 + 131 + 132 + 141 + 142 + 143 + \cdots + 242 + 243 + \cdots\,.$$

The realization is an algebra morphism: $\mathbb{G}_\sigma(\mathfrak{A}) \cdot \mathbb{G}_\mu(\mathfrak{A}) = r_{\mathfrak{A},std}(\mathbb{G}_\sigma \times \mathbb{G}_\mu)$ where "$\cdot$" is the classical concatenation product on words in the free algebra. For example,

$$\mathbb{G}_{213} \times \mathbb{G}_1 \quad = \quad \mathbb{G}_{2134} + \mathbb{G}_{2143} + \mathbb{G}_{3142} + \mathbb{G}_{3241}$$

which is equivalent to

$$\begin{aligned} r_{std,\mathbb{N}^*}(\mathbb{G}_{213} \times \mathbb{G}_1) &= \mathbb{G}_{213}(\mathbb{N}^*) \cdot \mathbb{G}_{\mathbf{1}}(\mathbb{N}^*) = (212 + 213 + 214 + \cdots) \cdot (\mathbf{1} + \mathbf{2} + \mathbf{3} + \mathbf{4} + \cdots) \\ &= 212\mathbf{1} + 212\mathbf{2} + 212\mathbf{3} + \cdots 213\mathbf{1} + 213\mathbf{2} + 213\mathbf{3} + \cdots + 324\mathbf{1} + \cdots \end{aligned}$$

Proposition 1: *If $span((m_u)_{u\in can_\varphi})$ is stable under the product $\times$ then it is given by:*

$$m_u \times m_v = \sum_{\substack{w:=u'v'\in can_\varphi \\ \varphi(u')=u;\varphi(v')=v}} m_w\,. \tag{2}$$

Remark 1: Let $\mathfrak{A}, \mathfrak{B}$ be two totally ordered alphabets such that any element in $\mathfrak{A}$ is strictly smaller than any element of $\mathfrak{B}$. By definition we have the following isomorphisms, where $\sqcup$ denotes the disjoint union:

$$\mathcal{A} \simeq \mathcal{A}(\mathfrak{A}) \simeq \mathcal{A}(\mathfrak{B}) \simeq \mathcal{A}(\mathfrak{A} \sqcup \mathfrak{B})\,. \tag{3}$$

3.2 Alphabet doubling trick

The *alphabet doubling trick* [DHT02, Hiv07] is a way to define coproducts. We consider the algebra $\mathbb{K}\langle\mathfrak{A} \sqcup \mathfrak{B}\rangle$ generated by two (infinite and totally ordered) alphabets $\mathfrak{A}$ and $\mathfrak{B}$ such that the letters of $\mathfrak{A}$ are strictly smaller than the letters of $\mathfrak{B}$. The relation $\rightleftarrows$ make the letters of $\mathfrak{A}$ commute with those of $\mathfrak{B}$. One identifies $\mathbb{K}\langle\mathfrak{A}\sqcup\mathfrak{B}\rangle/_{\rightleftarrows}$ with the algebra $\mathbb{K}\langle\mathfrak{A}\rangle\otimes\mathbb{K}\langle\mathfrak{B}\rangle$. We follow here the abuse of language allowing infinite but finite degree sum. We denote by $r_{\mathfrak{A}\sqcup\mathfrak{B}}(x)/_{\rightleftarrows}$ the image of $r_{\mathfrak{A}\sqcup\mathfrak{B}}(x)$ given by the canonical map from $\mathbb{K}\langle\mathfrak{A}\sqcup\mathfrak{B}\rangle$ to $\mathbb{K}\langle\mathfrak{A}\sqcup\mathfrak{B}\rangle/_{\rightleftarrows}$. The map $x\mapsto r_{\mathfrak{A}\sqcup\mathfrak{B}}(x)/_{\rightleftarrows}$ is always an algebra morphism from $\mathcal{A}$ to $\mathbb{K}\langle\mathfrak{A}\rangle\otimes\mathbb{K}\langle\mathfrak{B}\rangle$. Whenever its image is included in $\mathcal{A}(\mathfrak{A})\otimes\mathcal{A}(\mathfrak{B})$ this defines a coproduct on $\mathcal{A}$.

Definition 2 (Hopf polynomial realization): *A* Hopf polynomial realization r *of* $\mathcal{H}$ *is a polynomial realization such that for all* x*:*

$$r_{\mathfrak{A}\sqcup\mathfrak{B}}(x)/_{\rightleftarrows}=(r_{\mathfrak{A}}\otimes r_{\mathfrak{B}})(\Delta(x))\ . \tag{4}$$

Example 2 (Coproduct in FQSym): We denote by $\mathbb{G}_\sigma(\mathfrak{A}\sqcup\mathfrak{B})$ the std-polynomial realization of the **FQSym** element indexed by σ in the algebra $\mathbb{K}\langle\mathfrak{A}\sqcup\mathfrak{B}\rangle/_{\rightleftarrows}$. Also we denote by $1,2,3,\cdots$ the symbols of $\mathfrak{A}$ and in bold red $\mathbf{1},\mathbf{2},\mathbf{3},\cdots$ the symbols of $\mathfrak{B}$ ordered with $1<2<3<\cdots<\mathbf{1}<\mathbf{2}<\mathbf{3}<\cdots$. Then,

$$\begin{aligned}\mathbb{G}_{132}(\mathfrak{A}\sqcup\mathfrak{B}) &= 121+131+132+\cdots+1\mathbf{1}1+1\mathbf{12}+\cdots+1\mathbf{21}+1\mathbf{31}+\cdots+\mathbf{121}+\cdots\\ &= 121+131+\cdots+11\cdot\mathbf{1}+11\cdot\mathbf{2}+\cdots+1\cdot\mathbf{21}+1\cdot\mathbf{31}+\cdots+\mathbf{123}+\mathbf{132}+\cdots\\ &= \Delta(\mathbb{G}_{132})=1\otimes\mathbb{G}_{132}+\mathbb{G}_1\otimes\mathbb{G}_{21}+\mathbb{G}_{12}\otimes\mathbb{G}_1+\mathbb{G}_{132}\otimes 1\ .\end{aligned}$$

3.3 Good Hopf algebras

We call a *Hopf algebra* $\mathcal{H}_\varphi$ *good* if it is defined by a Hopf polynomial realization r_φ. We call a *function* φ *good* if it produces a *good Hopf algebra* $\mathcal{H}_\varphi$. Currently, we know three main good Hopf algebras: **FQSym**, **WQSym** and **PQSym** are respectivly associated to the *standardization*, *packing* and *parkization* functions.

4 Good monoids

In the previous section (Section 3), we realized some Hopf algebras in free algebras. In this section, we give sufficient conditions on a congruence $\equiv$ to build a combinatorial quotient of a *good Hopf algebra*. We call a *monoid good* if it statisfies these conditions. We give a sufficient compatibility between φ and $\equiv$ to ensure the product is carried to the quotient. The second condition ensures that the *alphabet doubling trick* map. It is used to project the coproduct in the quotient. Under these conditions, a monoids is guaranted to produce a Hopf algebra quotient (Theorem 1). Furthermore, these conditions on monoid are preserved under taking infimum and supremum (Theorem 2).

4.1 Definition

The notion of *Good monoids* has been introduced by HIVERT-NZEUTCHAP [HN07] to build quotients (sub-algebras) of **FQSym**. We could also mention PhD thesis.

A *good monoid* is a monoid which is similar to the *plactic monoid* [LS81, Knu73]. We consider a free monoid $\mathfrak{A}^*$ with concatenation product "$\cdot$", a congruence $\equiv$ on $\mathfrak{A}^*$ and a map $\varphi:\mathfrak{A}^*\to\mathfrak{A}^*$. We define the *evaluation* $ev(w)$ of a word w as its number of occurrences of each letter of w. For example, the words `ejajv` and `jjaev` have the same evaluation: both have one `a`, one `e`, one `v` and two `j`. The free monoid $\mathfrak{A}^*/\equiv$ is a φ*-good monoid* if it has the following properties:

Definition 3 (φ-congruence): *The congruence* $\equiv$ *is a* φ-congruence *if for all* $u,v\in\mathfrak{A}^*$*,* $u\equiv v$ *if and only if* $\varphi(u)\equiv\varphi(v)$ *and* $ev(u)=ev(v)$*.*

This first compatibility is sufficient to build a quotient algebra of $\mathcal{A}_\varphi$.

Definition 4 (Compatibility with restriction to alphabet intervals): *The congruence $\equiv$ is compatible with the* restriction to alphabet intervals *if, for all $u, v \in \mathfrak{A}^*$ such that $u \equiv v$ one has $u_{|I} \equiv v_{|I}$ for any I interval of $\mathfrak{A}$, where $w_{|\mathfrak{A}}$ is word restricted to the alphabet $\mathfrak{A}$.*

This second compatibility in association with the first ensures that *alphabet doubling trick* defines a quotient coproduct. Both compatibilities give us an extended definition of a HIVERT-NZEUTCHAP's *good monoid* which one is defined only with φ the standardization map:

Definition 5 (φ-good monoid): *A quotient $\mathfrak{A}^*/\equiv$ of the free monoid is a φ-*good monoid *if $\equiv$ is a φ-congruence and is compatible with restriction to alphabet intervals. We call such a congruence a φ-*good congruence.

In the following examples, we denote words of $\mathfrak{A}^*$ by u, v, w and the letters by a, b, c.

Example 3 (sylvester and stalactic monoids): The sylvester congruence: $\equiv_{sylv}$, defined by

$$u \cdot ac \cdot w \cdot b \cdot v \equiv_{sylv} u \cdot ca \cdot w \cdot b \cdot v \text{ whenever } a \leqslant b < c \ , \tag{5}$$

is std-compatible and compatible with the restriction to alphabet intervals. Thanks to the binary search tree insertion algorithm the equivalence classes are in natural bijection with binary search trees. The quotient monoid is a monoid on binary search trees called the sylvester monoid in [HNT05].

The stalactic congruence [HNT08a]: $\equiv_{stal}$, defined by

$$u \cdot ba \cdot v \cdot b \cdot w \equiv_{stal} u \cdot ab \cdot v \cdot b \cdot w \ , \tag{6}$$

is compatible with *packing* but not with *standardization*. The quotient monoid is the *stalactic monoid*. It is clear that any stalactic class contains a word of the form $a_1^{m_1} a_2^{m_2} \dots a_k^{m_k}$, where the a_i are distinct. We call these words canonical.

$$515431511145312455 \equiv_{stal} 3^2 1^5 2^1 4^3 5^6$$

4.2 Hopf algebra quotient

These differents *good monoids* tools was used to (re-)define several Hopf algebra quotients: **FSym** the Free Symmetric functions Hopf algebra [DHT02], **PBT** [LR98, HNT05] or Baxter Hopf algebra [Gir11, Gir12]; the Hopf algebra associated with the stalactic monoid [HNT08a]; or **CQSym** [NT04, NT07] (a **PQSym** quotient).

Lemma 1 (Algebra quotient): *Let $\mathcal{H}_\varphi$ be a* good Hopf algebra *and $\equiv$ be a φ-good congruence such that its free monoid quotient is a φ-*good monoid*. Then, the quotient $\mathcal{H}_\varphi/_\equiv$ is an algebra quotient whose bases are indexed by $can_\varphi/_\equiv$, identifying basis elements m_u and m_v whenever $u \equiv v$.*

Example 4 (PBT and Hopf algebra stalactic): We go back to Example 3. The sylvester quotient of **FQSym** is the Hopf algebra **PBT** [LR98, HNT05].

The stalactic monoid gives a quotient of **WQSym**. Let π be the projection of **WQSym** in **WQSym**$/_{\equiv_{stal}}$ and $u := 112$ and $v := 11$ two (packed) words. We denote by π the projection of $\mathbb{M}_u$ by $\mathbb{Q}_s$, with s the planar diagram associated to the stalactic class of u.

$$\begin{aligned} \pi(\mathbb{M}_{112} \times \mathbb{M}_{11}) &= \pi(\mathbb{M}_{11211} + \mathbb{M}_{11222} + \mathbb{M}_{11233} + \mathbb{M}_{11322} + \mathbb{M}_{22311}) \\ = \mathbb{Q}_{1^2 2} \times \mathbb{Q}_{1^2} &= \mathbb{Q}_{21^4} + \mathbb{Q}_{1^2 2^3} + \mathbb{Q}_{1^2 2 3^2} + \mathbb{Q}_{1^2 3 2^2} + \mathbb{Q}_{2^2 3 1^2} \end{aligned}$$

Lemma 2 (Coalgebra quotient): *The quotient $\mathcal{H}/_{\equiv}$ is a coalgebra quotient.*

SKETCH OF THE PROOF: The relation $\equiv$ is compatible with the restriction to alphabet intervals, hence the alphabet doubling trick ensures that coproduct projects to the quotient. ∎

Example 5:

$$\begin{aligned} \pi(\Delta(\mathbb{M}_{332122})) &= \pi(1 \otimes \mathbb{M}_{332122} + \mathbb{M}_1 \otimes \mathbb{M}_{22111} + \mathbb{M}_{2122} \otimes \mathbb{M}_{11} + \mathbb{M}_{332122} \otimes 1) \\ = \Delta(\mathbb{Q}_{3^21 2^3}) &= 1 \otimes \mathbb{Q}_{3^2 1 2^3} + \mathbb{Q}_1 \otimes \mathbb{Q}_{2^2 1^3} + \mathbb{Q}_{12^3} \otimes \mathbb{Q}_{1^2} + \mathbb{Q}_{3^2 1 2^3} \otimes 1 \end{aligned}$$

Theorem 1 (Good monoid and good Hopf algebra): *Let $\mathcal{H}_\varphi$ be a good Hopf algebra and $\equiv$ be a φ-good congruence. The quotient $\mathcal{H}/_{\equiv}$ is a Hopf algebra quotient.*

Corollary 1: *The dual Hopf algebra $(\mathcal{H}/_{\equiv})^{\#}$ is a sub-algebra of the dual Hopf algebra $\mathcal{H}^{\#}$, with basis given by:*

$$\overline{M}^{\#}_{U \in can_\varphi/_{\equiv}} = \textstyle\sum_{u \in U} m_u^{\#} \ . \tag{7}$$

4.3 Operations

Previously we introduced some *good functions* φ: std, $pack$ (and $park$). It is interesting to investigate the connections between them:

Definition 6 (refinement): *Let φ and π be two functions. We say that π refines φ, written $\varphi \prec \pi$ if $\varphi(\pi(u)) = \varphi(u)$ for all $u \in \mathfrak{A}^*$.*

It is clear that *refinement* is an order.

Proposition 2 (std, $tass$, $park$ and refinement): *For these three functions:* standardization std, packing $pack$ *and* parking $park$ *we have the relation: $std \prec pack \prec park$.*

Proposition 3 (Good functions and refinement): *Let φ and π be two* good functions *such that $\varphi \prec \pi$. Then any φ-*good monoid *is a π-*good monoid*.*

Propositions 2 and 3 give us, for example, that any std-good monoid is $pack$-good. Furthermore operations on two *good congruences* give *good congruences*.

Theorem 2 ($\vee$, $\wedge$ and good congruences): *The union and intersection of two φ-*good congruences $\sim$ *and $\approx$ are φ-*good congruences*.*

As an intriguing consequence the lattice structure on monoids is transported to Hopf algebras. Several examples of this are know.

Example 6: The intersection ($\equiv_{sylv} \ \wedge \ \equiv_{\#sylv}$) of the sylvester relation (5) and its image under the SCHÜTZENBERGER involution gives std-good monoid: the *Baxter monoid* [Gir11, **?**].

The union ($\equiv_{sylv} \vee \equiv_{\#sylv}$) of those relations gives the *hypoplactic monoid* [Nov00].

In the sequel, we study in detail another example.

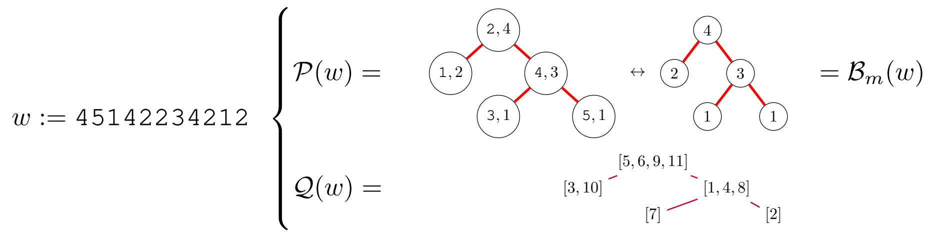


Figure 1: We start by considering the packed word `45142234212`, and insert it in a BSTM by the algorithm $\mathcal{P}$; that give us $\mathcal{P}($`45142234212`$)$ above in the middle. On the right, there is the BTm ($\mathcal{B}_m(w)$) associated with the BSTM ($\mathcal{P}(w)$) of **WQSym**$/_{\equiv_t}$. At the top of the figure there is the P-symbol given by $\mathcal{P}$ or $\mathcal{B}_m$ and below the Q-symbol is given by $\mathcal{Q}$.

5 The union of the sylvester and the stalactic congruences

As an application of the preceding construction, we consider the union ($\equiv_{sylv} \vee \equiv_{stal}$) of the sylvester congruence (5) and the stalactic congruence (6); we call it the *taïga relation* $\equiv_t$,

$$\begin{aligned} &u \cdot ac \cdot v \cdot b \cdot w \equiv_t u \cdot ca \cdot v \cdot b \cdot w \quad \text{for } a \leqslant b < c\ , \\ &u \cdot ba \cdot v \cdot b \cdot w \equiv_t u \cdot ab \cdot v \cdot b \cdot w \end{aligned} \tag{8}$$

From Proposition 3 we know that the sylvester congruence (5) is a $pack\text{-}good\ congruence$ and from Theorem 2 we deduce that the *taïga monoid* is a $pack\text{-}good\ monoid$.

5.1 Algorithm and taïga monoid

The taïga congruence can be calculated using an insertion algorithm similar to the binary search tree insertion (see Algorithm 3 for a definition). This insertion algorithm uses a search tree structure:

Definition 7 (Binary search tree with multiplicity): *A (planar)* binary search tree with multiplicity *(BSTM) is a binary tree T where each node is labelled by a letter l and a non-negative integer k, called the multiplicity, so that T is a binary search tree if we drop the multiplicities and such that each letter appears at most once in T. We denote by (l, k) a node label and for any node n, by $l(n)$ its letter and by $m(n)$ its multiplicity.*

We denote by $\mathcal{P}(w)$ the result of the insertion using Algorithm 3 of w from *the right to the left* in the empty tree (*cf.* the left part of the figure 1).

Proposition 4: *The taïga classes are the fibers of $\mathcal{P}$. That is for u and v two words: $u \equiv_t v$ if and only if $\mathcal{P}(u) = \mathcal{P}(v)$.*

The Q-symbol of w is the tree $\mathcal{Q}(w)$ of same shape as $\mathcal{P}(w)$ which records the positions of each inserted letter. This gives us a ROBINSON-SCHENSTED *like correspondance* [LS81] (*cf.* Figure 1). As a corollary of Theorem 2 we get the taïga monoid is a $tass\text{-}good\ monoid$.

5.2 Quotient of ***WQSym***: ***PBTm***

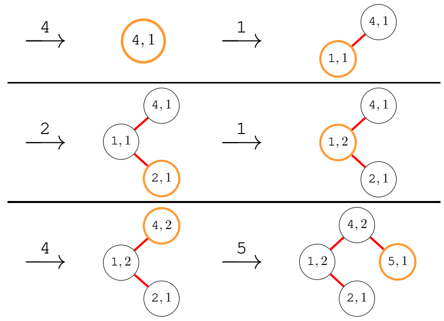
As in [HNT05], we consider a binary trees with multiplicities without letters.

Algorithm 3: insertion in a BSTM

```
Data: t a BSTM with L_t and R_t its left and right subtrees, and l a
      letter of 𝔄
Result: t with l inserted
if t is empty tree then
    t ← node labelled by (l, 1)
else
    if l(t) = l then
        increment m(t)
    else
        if l(t) < l then insert recursively l in L_t

        else insert recursively l in R_t
return t
```

Insertion of word `541214` from *the right to the left* in the empty tree :



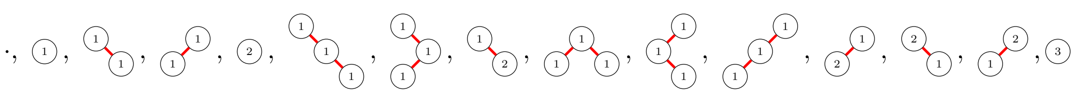
Definition 8 (BTM): *A* binary tree with multiplicities *(BTM) is a (planar) binary tree labelled by non-negative integers on its nodes. The size of a BTM T denoted by $|T|$ is the sum of the multiplicities.*

Let T_w be a BSTM associated to a packed word w, and T be the BTM obtained by removing its letters. One can recover uniquely T_w from T: indeed each letter of T_w is deduced by a left infix reading of T. We identify the set of words in $pack(\mathfrak{A}^*)/_{\equiv_t}$ of size k (for $k \geqslant 0$) with the set of BTM of size k. We denote $\mathcal{B}_m$ the algorithm which computes the BTM associated to the BSTM computed by $\mathcal{P}$ (*cf.* Figure 1).

Let us denote by $S(t)$ the generating series of these trees counted by size. The generating serie statisfies the following functional equation $S(t) = 1 + S(t)^2(1-t)^{-1}$ (see `A002212` of OEIS):

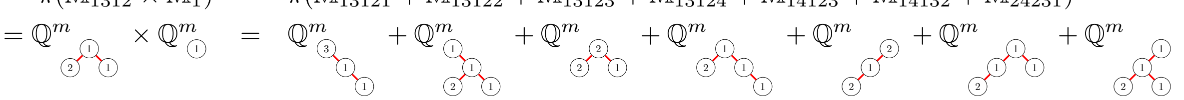
$$S(t) \quad = \frac{1-t-\sqrt{5t^2-6t+1}}{2t} = \quad 1 + t + 3t^2 + 10t^3 + 36t^4 + 137t^5 + 543t^6 + 2219t^7 + \dots \tag{9}$$

This structure is in bijection with *binary unary tree* structure. Here is the list of trees of size 0,1,2 and 3:

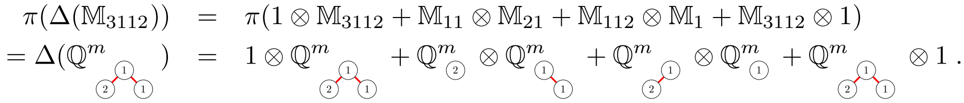


With Lemma 1 and Theorem 1 we know that the quotient of **WQSym**$(\mathfrak{A})$ by the taïga relations has a natural basis indexed by $tass(\mathfrak{A}^*)/_{\equiv_t}$ identified by BTM. We call **PBTm** (planar binary tree with multiplicities) that quotient. More precisely, we consider the basis $(\mathbb{M}_u)_u$ of **WQSym** obtained by the Hopf polynomial realization r_{tass}. We denote by $(\mathbb{Q}^m_t)_t$ the canonical projection by the map π of $(\mathbb{M}_u)_u$ in **PBTm** such that $\pi(\mathbb{M}_u) := \mathbb{Q}^m_t$ if $t = \mathcal{B}_m(u)$. The product and coproduct are given by some explicit algorithms. For brevity, we only give here some examples:

$$\pi(\mathbb{M}_{1312} \times \mathbb{M}_1) \quad = \quad \pi(\mathbb{M}_{13121} + \mathbb{M}_{13122} + \mathbb{M}_{13123} + \mathbb{M}_{13124} + \mathbb{M}_{14123} + \mathbb{M}_{14132} + \mathbb{M}_{24231})$$



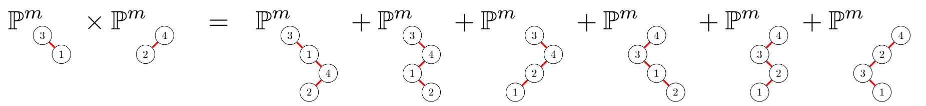
$$\pi(\Delta(\mathbb{M}_{3112})) \quad = \quad \pi(1 \otimes \mathbb{M}_{3112} + \mathbb{M}_{11} \otimes \mathbb{M}_{21} + \mathbb{M}_{112} \otimes \mathbb{M}_1 + \mathbb{M}_{3112} \otimes 1)$$



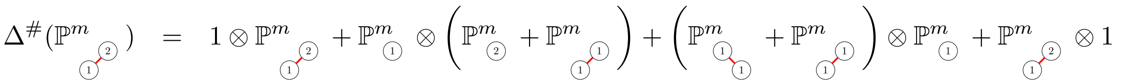
We consider $\mathbf{PBTm}^{\#}$ the dual of $\mathbf{PBTm}$. This is a sub-algebra of $\mathbf{WQSym}^{\#}$. We denote by $(\mathbb{P}^m_t)_t := (\mathbb{Q}^m_t)^{\#}$ its dual basis: $\langle \mathbb{Q}^m_t, \mathbb{P}^m_{t'}\rangle = \delta_{t,t'}$. The product is given by:

$$\mathbb{P}^m_{t'} \times \mathbb{P}^m_{t''} \quad = \quad \textstyle\sum_t \langle \Delta(\mathbb{Q}^m_t), \mathbb{P}^m_{t'} \times \mathbb{P}^m_{t''}\rangle \mathbb{P}^m_t \ . \tag{10}$$

Here is an example,



If we consider only shape tree, the product is exactly the product of $(\mathbb{P}_t)_t$ basis in **PBT** [HNT05]. Hence this product is a shifted shuffle on trees. The coproduct is given by: $\Delta^{\#}(\mathbb{P}^m_t) = \sum_{t',t''} \langle \mathbb{Q}^m_{t'} \times \mathbb{Q}^m_{t''}, \mathbb{P}^m_t\rangle \, \mathbb{P}^m_{t'} \otimes \mathbb{P}^m_{t''}$. Here is an example:



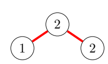
6 The hook length formula

Its well known from [Knu73] (§5.14 ex. 20) that the number of decreasing labelling of a binary tree is given by a simple product formula. [HNT05] remarks that this is also the number of permutations given upon a tree by the binary search tree insertion. In this section we generalize this formula for trees with multiplicities.

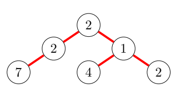
Proposition 5: *The cardinal $f(T)$ of the taïga class associated to T (i.e. the set of packed words giving the tree T by the insertion algorithm $\mathcal{B}_m$) is given by*

$$f(T) \quad = \quad |T|! \left(\textstyle\prod_{t\in T} |t| \, (m(t)-1)!\right)^{-1} \ . \tag{11}$$

where t ranges throwgh all the subtrees of T and $|T|$ denotes the size of T (the sum of the multiplicities).

Example 7: The taïga class of $T :=$  contains 12 packed words w:

$$23132, 33122, 31232, 32312, 13232, 33212, 23312, 32132, 21332, 31322, 12332, 13322 \ .$$

The class of  contains $23,337,600 = \frac{18!}{(18\cdot 9\cdot 7\cdot 7\cdot 4\cdot 2)(1!1!6!0!3!1!)}$ packed words.

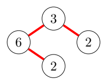
This formula is easily proven by induction. However, we prefer to give a generating series proof as in [HNT08b]. Let $\mathcal{A}$ be an associative algebra, and consider the functional equation for power series $x \in \mathcal{A}[[z]]$:

$$x \quad = \quad a + \textstyle\sum_{k\geqslant 1} B_k(x,x) \ , \tag{12}$$

where $a \in \mathcal{A}$ and for any $k > 0$, $B_k(x,y)$ is a bilinear map with values in $\mathcal{A}[[z]]$. We suppose such that the valuation of $B_k(x,y)$ is strictly greater than the sum of the valuations of x and y (plus k). Then

Equation 12 has a unique solution:

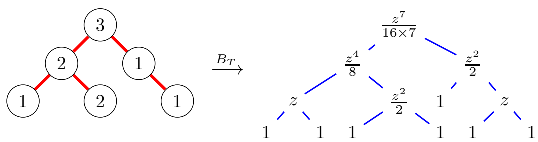
$$\begin{aligned} x &= a + \sum_{k\geqslant 1}\Big(B_k(a,a) + B_k(a, \sum_{k'\geqslant 1} B_{k'}(a,a)) + B_k(\sum_{k'\geqslant 1} B_{k'}(a,a)), a) + \ldots\Big) \quad (13)\\ &= \sum_{T\in \mathbf{BTM}} B_T(a)\ , \end{aligned}$$

where for a tree T, $B_T(a)$ is the result of evaluating the expression formed by labelling by a the leaves of the complete tree associated to T and by B_k its internal node labelled by k.
For example: B  $(a) = B_3(B_6(a, B_2(a,a)), B_2(a,a))$.

So if we try to solve the fixed point problem:

$$x = 1 + \int_0^z e^s x(s)^2 ds = 1 + \sum_{k\geqslant 1} \int_0^z \frac{s^{k-1}}{(k-1)!} x(s)^2 ds \quad = \quad 1 + \sum_{k\geqslant 1} B_k(x,x)\ , \qquad (14)$$

where $B_k(x,y) = \int_0^z \frac{s^{k-1}}{(k-1)!} x(s)y(s)ds$. Then for a binary tree of non-negative integer T, $B_T(1)$ is the monomial obtained by putting 1 on each leaf and integrating at each node n the product of the evaluations of its subtrees and $s^k/k!$ with $m(n) = k+1$.

For example:  .

One can observe that $B_T(1) = f(T)\frac{z^n}{n!}$, where $n = |T|$.

To prove the *hook length formula*, following the same technique as in [HNT08b], we want to lift in $\mathbf{WQSym}^{\#}$ the fixed point computation of Equation 14. From the multiplication rule [Hiv99] of the dual basis $\mathbb{S}_u$ ($\mathbb{M}_u^{\#} := \mathbb{S}_u$), one easily sees that the linear map $\phi : \mathbb{S}_u \mapsto \frac{z^n}{n!}$ with n the length of u is a morphism of algebras from $\mathbf{WQSym}^{\#}$ to $\mathbb{K}[[z]]$. For u, v two packed words of respective size $n-1$ and m, set $B_k(\mathbb{S}_u, \mathbb{S}_v) := \sum_{w\in (u\overline{\sqcup\!\sqcup} 1^{k-1} \overline{\sqcup\!\sqcup} v)\cdot n} \mathbb{S}_w$. The crucial observation which allows to express the *hook length formula* in a generating series way is the following theorem:

Theorem 3: *For any binary tree with multiplicities $B_T(1) = \sum_{\mathcal{B}_m(u)=T} \mathbb{S}_u$.*

In particular, $B_T(1)$ coincide with $\mathbb{P}_T^m$, the natural basis of $\textbf{PBTm}^{\#}$.

Corollary 2: *The number of packed words u such that $\mathcal{B}_m(u) = T$ is computed by $f(T)$.*

7 Conclusion, work in progress and perspectives

In this paper, we unraveled some new combinatorics on binary trees with multiplicities from the union of the *sylvester* and *stalactic* monoids. Using the machinery of *realizations*, we built a Hopf algebra on those trees, allowing us to give a generating series proof of a new *hook length formula*. Following [HNT08b], it is very likely that we will also be able to prove a *q-hook length formula*. On the other hand, the usual case of the LODAY-RONCO algebra has a lot of nice properties. For example, the product and coproduct can be expressed by the means of an order on the trees called the *Tamari Lattice* [LR98]. It would be good to know if such a lattice exists for trees with multiplicities. This should also relate to N. READING work on lattice congruences [Rea05]. Also it could be interesting to study some other combinations in the lattice of *good monoids*. For example, the union of the *plactic monoid* and the *stalactic monoid* should give a Hopf algebra of *tableaux* with multiplicties. Finally, in our construction, it seems that std, $tass$ and $park$ play some canonical role from which everything else is built. Are there some more examples? Is there a definition for such a φ-map? Could we except to always have a *hook formula* as soon as we have a *good monoid*?

References

[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(05):671–717, 2002.

[Gir11] S. Giraudo. Algebraic and combinatorial structures on baxter permutations. In *Formal Power Series and Algebraic Combinatorics*, volume 23, pages 387–398, 2011.

[Gir12] S. Giraudo. Algebraic and combinatorial structures on pairs of twin binary trees. *Journal of Algebra*, 2012.

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

[Hiv99] F. Hivert. Combinatoire des fonctions quasi-symétriques. *Thèse de Doctorat, Marne-La-Vallée*, 1999.

[Hiv07] F. Hivert. An introduction to combinatorial hopf algebras. *Physics and Theoretical Computer Science: From Numbers and Languages to (Quantum) Cryptography Security*, 7:253, 2007.

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

[HNT05] F. Hivert, J.C. Novelli, and J.Y. Thibon. The algebra of binary search trees. *Theoretical Computer Science*, 339(1):129–165, 2005.

[HNT08a] F. Hivert, J.C. Novelli, and J.Y. Thibon. Commutative combinatorial hopf algebras. *Journal of Algebraic Combinatorics*, 28(1):65–95, 2008.

[HNT08b] F. Hivert, J.C. Novelli, and J.Y. Thibon. Trees, functional equations, and combinatorial hopf algebras. *European Journal of Combinatorics*, 29(7):1682–1695, 2008.

[Knu73] D.E. Knuth. *The art of computer programming, Vol. 3*, volume 109. Addison-Wesley, Reading, MA, 1973.

[LR98] J.L. Loday and M.O. Ronco. Hopf algebra of the planar binary trees. *Advances in Mathematics*, 139(2):293–309, 1998.

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

[Mac95] I.G. Macdonald. *Symmetric functions and Hall polynomials*. New York, 1995.

[MR95] C. Malvenuto and C. Reutenauer. Duality between quasi-symmetrical functions and the solomon descent algebra. *Journal of Algebra*, 177(3):967–982, 1995.

[Nov00] J.C. Novelli. On the hypoplactic monoid. *Discrete Mathematics*, 217(1):315–336, 2000.

[NT04] J.C. Novelli and J.Y. Thibon. A hopf algebra of parking functions. In *FPSAC PROCEEDINGS 2004 ACTES SFCA 2004 Vancouver CANADA*, page 215, 2004.

[NT06] JC Novelli and JY Thibon. Polynomial realizations of some trialgebras, fpsac'06. *Also preprint ArXiv: math. CO/0605061*, 2006.

[NT07] J.C. Novelli and J.Y. Thibon. Parking functions and descent algebras. *Annals of Combinatorics*, 11(1):59–68, 2007.

[PR95] S. Poirier and C. Reutenauer. Algebres de hopf de tableaux. In *Annales des Sciences mathématiques du Québec*, volume 19, pages 79–90. Université du Québec à Montréal, Département de mathématiques et informatique, 1995.

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

[S+12] W. A. Stein et al. *Sage Mathematics Software (Version 5.3)*. The Sage Development Team, 2012. `http://www.sagemath.org`.

[SCc12] The Sage-Combinat community. Sage-Combinat: enhancing Sage as a toolbox for computer exploration in algebraic combinatorics, 2012. `http://combinat.sagemath.org`.