Partitioning maps and Hopf algebras

The 72th Séminaire Lotharingien de Combinatoire
Lyon

Florent HIVERT and Jean-Baptiste PRIEZ

Laboratoire de Recherche en Informatique
Faculté des sciences d'Orsay
Université Paris-Sud

Tuesday, March 25th 2014

Introduction

Partitioning maps and Hopf algebras

Generalized Parking Functions and (a_n)-parkization

Combinatorial Hopf Algebra (CHA)

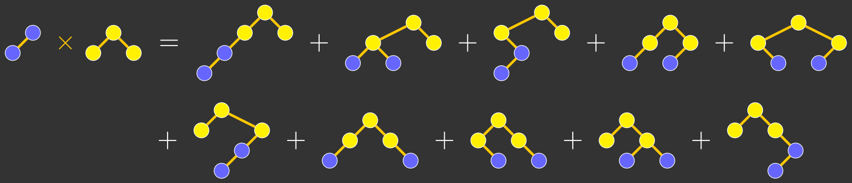
- vector space $\mathcal{H}$
- with *basis* indexed by a *combinatorial class*

Combinatorial Hopf Algebra (CHA)

- vector space $\mathcal{H}$
- with *basis* indexed by a *combinatorial class*
- $\times$: the product *assembles* elements

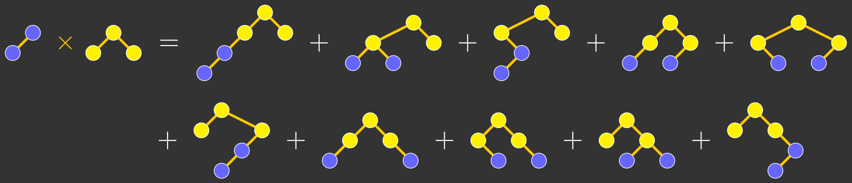
Combinatorial Hopf Algebra (CHA)

- vector space $\mathcal{H}$
- with *basis* indexed by a *combinatorial class*
- $\times$: the product *assembles* elements



Combinatorial Hopf Algebra (CHA)

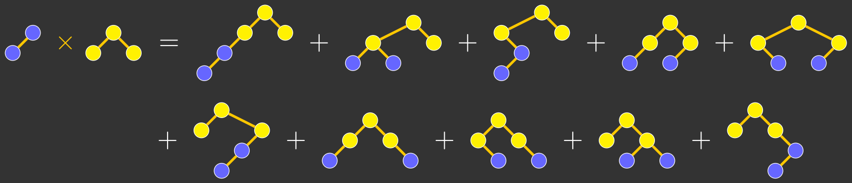
- vector space $\mathcal{H}$
- with *basis* indexed by a *combinatorial class*
- $\times$: the product *assembles* elements



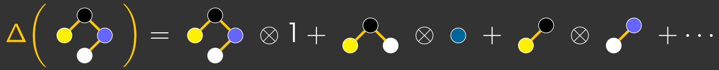
- Δ : the coproduct *disassembles* elements

Combinatorial Hopf Algebra (CHA)

- vector space $\mathcal{H}$
- with *basis* indexed by a *combinatorial class*
- $\times$: the product *assembles* elements



- Δ : the coproduct *disassembles* elements

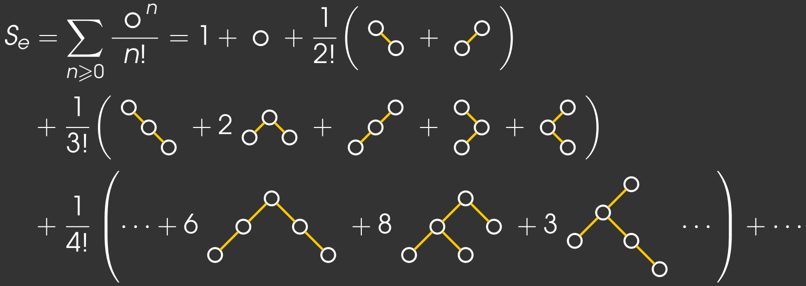


Improve the expressivity of formal series

Refine the size by the objects :

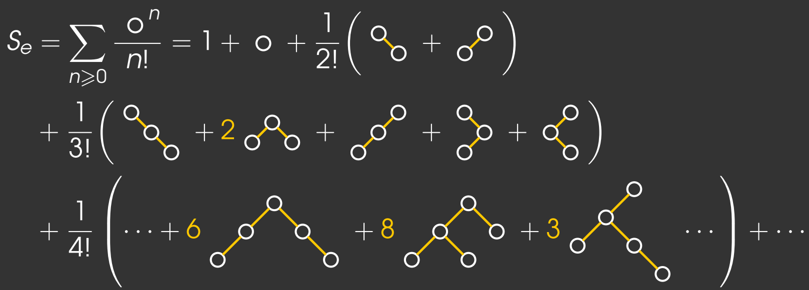
Improve the expressivity of formal series

Refine the size by the objects :



Improve the expressivity of formal series

Refine the size by the objects :



More information :

What is the coefficient of the tree T ?
$\rightarrow$ *The number of permutations giving T by binary search tree insertion*

Outline

Polynomial realization + "alphabet doubling trick" :

- Define lots of Hopf algebras : ***FQSym***, ***WQSym***, ***PQSym***, ...

$$m_\sigma(\mathfrak{A}) = \sum_{\substack{w \in \mathfrak{A}^* \\ \varphi(w)=\sigma}} w.$$

Outline

Polynomial realization + "alphabet doubling trick" :

- Define lots of Hopf algebras : ***FQSym***, ***WQSym***, ***PQSym***, ...

$$m_\sigma(\mathfrak{A}) = \sum_{\substack{w\in\mathfrak{A}^* \\ \varphi(w)=\sigma}} w.$$

- *Product* = *concatenation* :

$$m_\sigma(\mathfrak{A})m_\mu(\mathfrak{A}) = \left(\sum_{\varphi(u)=\sigma} u\right) \cdot \left(\sum_{\varphi(v)=\mu} v\right)$$

Outline

Polynomial realization + "alphabet doubling trick" :

- Define lots of Hopf algebras : ***FQSym***, ***WQSym***, ***PQSym***, . . .

$$m_\sigma(\mathfrak{A}) = \sum_{\substack{w\in\mathfrak{A}^* \\ \varphi(w)=\sigma}} w.$$

- *Product* = *concatenation* :

$$m_\sigma(\mathfrak{A})m_\mu(\mathfrak{A}) = \left(\sum_{\varphi(u)=\sigma} u\right)\cdot\left(\sum_{\varphi(v)=\mu} v\right) = \sum_{\substack{\varphi(u)=\sigma \\ \varphi(v)=\mu}} u\cdot v.$$

Outline

Polynomial realization + "alphabet doubling trick" :

- Define lots of Hopf algebras : ***FQSym***, ***WQSym***, ***PQSym***, ...

$$m_\sigma(\mathfrak{A}) = \sum_{\substack{w \in \mathfrak{A}^* \\ \varphi(w)=\sigma}} w.$$

- *Product* = *concatenation* :

$$m_\sigma(\mathfrak{A}) m_\mu(\mathfrak{A}) = \left(\sum_{\varphi(u)=\sigma} u\right) \cdot \left(\sum_{\varphi(v)=\mu} v\right) = \sum_{\substack{\varphi(u)=\sigma \\ \varphi(v)=\mu}} u \cdot v.$$

- *Coproduct* by *specialization* : "*alphabet doubling trick*" :

$$m_\sigma(\mathfrak{A}) \simeq m_\sigma(\mathfrak{A} \oplus \mathfrak{B}) \simeq m_\sigma(\mathfrak{B}).$$

Outline

Polynomial realization + "alphabet doubling trick" :

- Define lots of Hopf algebras : ***FQSym***, ***WQSym***, ***PQSym***, ...

$$m_\sigma(\mathfrak{A}) = \sum_{\substack{w\in\mathfrak{A}^* \\ \varphi(w)=\sigma}} w.$$

- *Product* = *concatenation* :

$$m_\sigma(\mathfrak{A})m_\mu(\mathfrak{A}) = \left(\sum_{\varphi(u)=\sigma} u\right) \cdot \left(\sum_{\varphi(v)=\mu} v\right) = \sum_{\substack{\varphi(u)=\sigma \\ \varphi(v)=\mu}} u \cdot v.$$

- *Coproduct* by *specialization* : "*alphabet doubling trick*" :

$$m_\sigma(\mathfrak{A}) \simeq m_\sigma(\mathfrak{A} \oplus \mathfrak{B}) \simeq m_\sigma(\mathfrak{B}) \quad \text{and}$$

$$\delta_{\mathfrak{A}\rightleftarrows\mathfrak{B}}(m_\sigma(\mathfrak{A} \oplus \mathfrak{B})).$$

Outline

Polynomial realization + "alphabet doubling trick" :

- Define lots of Hopf algebras : ***FQSym***, ***WQSym***, ***PQSym***, ...

$$m_\sigma(\mathfrak{A}) = \sum_{\substack{w \in \mathfrak{A}^* \\ \varphi(w)=\sigma}} w.$$

- *Product* = *concatenation* :

$$m_\sigma(\mathfrak{A}) m_\mu(\mathfrak{A}) = \left(\sum_{\varphi(u)=\sigma} u\right) \cdot \left(\sum_{\varphi(v)=\mu} v\right) = \sum_{\substack{\varphi(u)=\sigma \\ \varphi(v)=\mu}} u \cdot v.$$

- *Coproduct* by *specialization* : "*alphabet doubling trick*" :

$$m_\sigma(\mathfrak{A}) \simeq m_\sigma(\mathfrak{A} \oplus \mathfrak{B}) \simeq m_\sigma(\mathfrak{B}) \quad \text{and}$$

$$\delta_{\mathfrak{A} \rightleftarrows \mathfrak{B}}(m_\sigma(\mathfrak{A} \oplus \mathfrak{B})) = \sum_{\varphi(w)=\sigma} \delta_{\mathfrak{A} \rightleftarrows \mathfrak{B}}(w).$$

Outline

Polynomial realization + "alphabet doubling trick" :

- Define lots of Hopf algebras : ***FQSym***, ***WQSym***, ***PQSym***, . . .

$$m_\sigma(\mathfrak{A}) = \sum_{\substack{w \in \mathfrak{A}^* \\ \varphi(w)=\sigma}} w.$$

- *Product* = *concatenation* :

$$m_\sigma(\mathfrak{A}) m_\mu(\mathfrak{A}) = \left(\sum\nolimits_{\varphi(u)=\sigma} u\right) \cdot \left(\sum\nolimits_{\varphi(v)=\mu} v\right) = \sum_{\substack{\varphi(u)=\sigma \\ \varphi(v)=\mu}} u \cdot v.$$

- *Coproduct* by *specialization* : "*alphabet doubling trick*" :

$$m_\sigma(\mathfrak{A}) \simeq m_\sigma(\mathfrak{A} \oplus \mathfrak{B}) \simeq m_\sigma(\mathfrak{B}) \quad \text{and}$$

$$\delta_{\mathfrak{A} \rightleftarrows \mathfrak{B}}(m_\sigma(\mathfrak{A} \oplus \mathfrak{B})) = \sum_{\varphi(w)=\sigma} \delta_{\mathfrak{A} \rightleftarrows \mathfrak{B}}(w) = \sum_{\varphi(w)=\sigma} w_{|\mathfrak{A}} \otimes w_{|\mathfrak{B}} \,.$$

Outline

Polynomial realization + "alphabet doubling trick" :

- Define lots of Hopf algebras : ***FQSym***, ***WQSym***, ***PQSym***, . . .

$$m_\sigma(\mathfrak{A}) = \sum_{\substack{w \in \mathfrak{A}^* \\ \varphi(w)=\sigma}} w.$$

- *Product* = *concatenation* :

$$m_\sigma(\mathfrak{A}) m_\mu(\mathfrak{A}) = \left(\sum_{\varphi(u)=\sigma} u\right) \cdot \left(\sum_{\varphi(v)=\mu} v\right) = \sum_{\substack{\varphi(u)=\sigma \\ \varphi(v)=\mu}} u \cdot v.$$

- *Coproduct* by *specialization* : "*alphabet doubling trick*" :

$$m_\sigma(\mathfrak{A}) \simeq m_\sigma(\mathfrak{A} \oplus \mathfrak{B}) \simeq m_\sigma(\mathfrak{B}) \quad \text{and}$$

$$\delta_{\mathfrak{A} \rightleftarrows \mathfrak{B}}(m_\sigma(\mathfrak{A} \oplus \mathfrak{B})) = \sum_{\varphi(w)=\sigma} \delta_{\mathfrak{A} \rightleftarrows \mathfrak{B}}(w) = \sum m_\tau(\mathfrak{A}) \otimes m_\mu(\mathfrak{B})\,.$$

Outline

Polynomial realization + "alphabet doubling trick" :

- Define lots of Hopf algebras : ***FQSym***, ***WQSym***, ***PQSym***, . . .

$$m_\sigma(\mathfrak{A}) = \sum_{\substack{w\in\mathfrak{A}^*\\ \varphi(w)=\sigma}} w.$$

- *Product* = *concatenation* :

$$m_\sigma(\mathfrak{A})m_\mu(\mathfrak{A}) = \left(\sum_{\varphi(u)=\sigma} u\right) \cdot \left(\sum_{\varphi(v)=\mu} v\right) = \sum_{\substack{\varphi(u)=\sigma\\ \varphi(v)=\mu}} u \cdot v.$$

- *Coproduct* by *specialization* : "*alphabet doubling trick*" :

$$m_\sigma(\mathfrak{A}) \simeq m_\sigma(\mathfrak{A} \oplus \mathfrak{B}) \simeq m_\sigma(\mathfrak{B}) \quad \text{and}$$

$$\delta_{\mathfrak{A}\rightleftarrows\mathfrak{B}}(m_\sigma(\mathfrak{A} \oplus \mathfrak{B})) = \sum_{\varphi(w)=\sigma} \delta_{\mathfrak{A}\rightleftarrows\mathfrak{B}}(w) = \sum m_\tau(\mathfrak{A}) \otimes m_\mu(\mathfrak{B}).$$

Question : What are the constraints on φ and $\mathfrak{A}$?

Outline

Polynomial realization + "alphabet doubling trick" :

- Define lots of Hopf algebras : ***FQSym***, ***WQSym***, ***PQSym***, ...

$$m_\sigma(\mathfrak{A}) = \sum_{\substack{w \in \mathfrak{A}^* \\ \varphi(w)=\sigma}} w.$$

- *Product* = *concatenation* :

$$m_\sigma(\mathfrak{A}) m_\mu(\mathfrak{A}) = \left(\sum_{\varphi(u)=\sigma} u\right) \cdot \left(\sum_{\varphi(v)=\mu} v\right) = \sum_{\substack{\varphi(u)=\sigma \\ \varphi(v)=\mu}} u \cdot v.$$

- *Coproduct* by *specialization* : "*alphabet doubling trick*" :

$$m_\sigma(\mathfrak{A}) \simeq m_\sigma(\mathfrak{A} \oplus \mathfrak{B}) \simeq m_\sigma(\mathfrak{B}) \quad \text{and}$$

$$\delta_{\mathfrak{A} \rightleftarrows \mathfrak{B}}(m_\sigma(\mathfrak{A} \oplus \mathfrak{B})) = \sum_{\varphi(w)=\sigma} \delta_{\mathfrak{A} \rightleftarrows \mathfrak{B}}(w) = \sum m_\tau(\mathfrak{A}) \otimes m_\mu(\mathfrak{B}).$$

Question : What are the constraints on φ and $\mathfrak{A}$?

- *Partitioning maps*

Outline

Polynomial realization + "alphabet doubling trick" :

- Define lots of Hopf algebras : ***FQSym***, ***WQSym***, ***PQSym***, . . .

$$m_\sigma(\mathfrak{A}) = \sum_{\substack{w \in \mathfrak{A}^* \\ \varphi(w)=\sigma}} w.$$

- *Product* = *concatenation* :

$$m_\sigma(\mathfrak{A}) m_\mu(\mathfrak{A}) = \left(\sum_{\varphi(u)=\sigma} u\right) \cdot \left(\sum_{\varphi(v)=\mu} v\right) = \sum_{\substack{\varphi(u)=\sigma \\ \varphi(v)=\mu}} u \cdot v.$$

- *Coproduct* by *specialization* : "*alphabet doubling trick*" :

$$m_\sigma(\mathfrak{A}) \simeq m_\sigma(\mathfrak{A} \oplus \mathfrak{B}) \simeq m_\sigma(\mathfrak{B}) \quad \text{and}$$

$$\delta_{\mathfrak{A} \rightleftarrows \mathfrak{B}}(m_\sigma(\mathfrak{A} \oplus \mathfrak{B})) = \sum_{\varphi(w)=\sigma} \delta_{\mathfrak{A} \rightleftarrows \mathfrak{B}}(w) = \sum m_\tau(\mathfrak{A}) \otimes m_\mu(\mathfrak{B}).$$

Question : What are the constraints on φ and $\mathfrak{A}$?

- *Partitioning maps*
- *Type of alphabet*

Introduction

Partitioning maps and Hopf algebras

Generalized Parking Functions and (a_n)-parkization

Partitioning maps (Condition 1/3)

Let $\varphi : \mathfrak{A}^* \to \mathcal{C}$.

Definition (Condition 1)

φ is *graded* and *surjective*.

Partitioning maps (Condition 1/3)

Let $\varphi : \mathfrak{A}^* \to \mathcal{C}$.

Definition (Condition 1)

φ is *graded* and *surjective*.

Occ : $\mathfrak{A}^* \to \mathbb{S}$: identifying position of repeated letters
$\to$ *word* to *set partition*

$$Occ(\texttt{jgezoinobgzegio}) = \{$$

Partitioning maps (Condition 1/3)

Let $\varphi : \mathfrak{A}^* \to \mathcal{C}$.

Definition (Condition 1)

φ is *graded* and *surjective*.

$Occ : \mathfrak{A}^* \to \mathbb{S}$: identifying position of repeated letters
$\to$ *word* to *set partition*

$$Occ(\mathtt{jgezoinobgzegio}) = \{\{1\}$$

Partitioning maps (Condition 1/3)

Let $\varphi : \mathfrak{A}^* \to \mathcal{C}$.

Definition (Condition 1)

φ is *graded* and *surjective*.

Occ $: \mathfrak{A}^* \to \mathbb{S}$: identifying position of repeated letters
$\to$ *word* to *set partition*

$$Occ(\texttt{jgezoinobgzegio}) = \{\{1\}, \{2, 10, 13\}$$

Partitioning maps (Condition 1/3)

Let $\varphi : \mathfrak{A}^* \to \mathcal{C}$.

Definition (Condition 1)

φ is *graded* and *surjective*.

$Occ : \mathfrak{A}^* \to \mathbb{S}$: identifying position of repeated letters
$\to$ *word* to *set partition*

$$Occ(\texttt{jgezoinobgzegio}) = \{\{1\}, \{2, 10, 13\}, \{3, 12\}$$

Partitioning maps (Condition 1/3)

Let $\varphi : \mathfrak{A}^* \to \mathcal{C}$.

Definition (Condition 1)

φ is *graded* and *surjective*.

$Occ : \mathfrak{A}^* \to \mathbb{S}$: identifying position of repeated letters
$\to$ *word* to *set partition*

$$Occ(\texttt{jgeZoinobgZegio}) = \{\{1\}, \{2, 10, 13\}, \{3, 12\}, \{4, 11\}$$

Partitioning maps (Condition 1/3)

Let $\varphi : \mathfrak{A}^* \to \mathcal{C}$.

Definition (Condition 1)

φ is *graded* and *surjective*.

$\underline{Occ : \mathfrak{A}^* \to \mathbb{S}}$: identifying position of repeated letters
$\to$ *word* to *set partition*

$$Occ(\texttt{jgezoinobgzegio}) = \{\{1\}, \{2, 10, 13\}, \{3, 12\}, \{4, 11\}, \{5, 8, 15\}$$

Partitioning maps (Condition 1/3)

Let $\varphi : \mathfrak{A}^* \to \mathcal{C}$.

Definition (Condition 1)

φ is *graded* and *surjective*.

$Occ : \mathfrak{A}^* \to \mathbb{S}$: identifying position of repeated letters
$\to$ *word* to *set partition*

$$Occ(\mathtt{jgezoinobgzegio}) = \{\{1\}, \{2, 10, 13\}, \{3, 12\}, \{4, 11\}, \{5, 8, 15\}, \{6, 14\}$$

Partitioning maps (Condition 1/3)

Let $\varphi : \mathfrak{A}^* \to \mathcal{C}$.

Definition (Condition 1)

φ is *graded* and *surjective*.

$Occ : \mathfrak{A}^* \to \mathbb{S}$: identifying position of repeated letters
$\to$ *word* to *set partition*

$$Occ(\texttt{jgezoinobgzegio}) = \{\{1\}, \{2, 10, 13\}, \{3, 12\}, \{4, 11\}, \{5, 8, 15\}, \{6, 14\}, \{7\}$$

Partitioning maps (Condition 1/3)

Let $\varphi : \mathfrak{A}^* \to \mathcal{C}$.

Definition (Condition 1)

φ is *graded* and *surjective*.

$Occ : \mathfrak{A}^* \to \mathbb{S}$: identifying position of repeated letters
$\to$ *word* to *set partition*

$Occ(\texttt{jgezoinobgzegio}) = \{\{1\}, \{2, 10, 13\}, \{3, 12\}, \{4, 11\}, \{5, 8, 15\}, \{6, 14\}, \{7\}, \{9\}$

Partitioning maps (Condition 1/3)

Let $\varphi : \mathfrak{A}^* \to \mathcal{C}$.

Definition (Condition 1)

φ is *graded* and *surjective*.

$Occ : \mathfrak{A}^* \to \mathbb{S}$: identifying position of repeated letters
$\to$ *word* to *set partition*

$Occ(\texttt{jgezoinobgzegio}) = \{\{1\}, \{2, 10, 13\}, \{3, 12\}, \{4, 11\}, \{5, 8, 15\}, \{6, 14\}, \{7\}, \{9\}\}$

Partitioning maps (Condition 1/3)

Let $\varphi : \mathfrak{A}^* \to \mathcal{C}$.

Definition (Condition 1)

φ is *graded* and *surjective*.

$Occ : \mathfrak{A}^* \to \mathbb{S}$: identifying position of repeated letters
$\to$ *word* to *set partition*

$Occ(\texttt{jgezoinobgzegio}) = \{\{1\}, \{2, 10, 13\}, \{3, 12\}, \{4, 11\}, \{5, 8, 15\}, \{6, 14\}, \{7\}, \{9\}\}$

- *graded* : word w of size k $\to$ π set partition of size k,

Partitioning maps (Condition 1/3)

Let $\varphi : \mathfrak{A}^* \to \mathcal{C}$.

Definition (Condition 1)

φ is *graded* and *surjective*.

$Occ : \mathfrak{A}^* \to \mathbb{S}$: identifying position of repeated letters
$\to$ *word* to *set partition*

$Occ(\texttt{jgezoinobgzegio}) = \{\{1\}, \{2, 10, 13\}, \{3, 12\}, \{4, 11\}, \{5, 8, 15\}, \{6, 14\}, \{7\}, \{9\}\}$

- *graded* : word w of size k $\quad\to\quad$ π set partition of size k,
- *surjective* : if $\mathfrak{A}$ infinite.

Partitioning maps (Condition 1/3) - 2

Lemma

$S_\varphi(\mathfrak{A}) := \left\langle m_\sigma \mid m_\sigma := \sum_{\varphi(w)=\sigma} w \right\rangle$ has bases indexed by $\mathcal{C}$.

Partitioning maps (Condition 1/3) - 2

Lemma

$S_\varphi(\mathfrak{A}) := \left\langle m_\sigma \mid m_\sigma := \sum_{\varphi(w)=\sigma} w \right\rangle$ has bases indexed by $\mathcal{C}$.

$$m_{\{\{1,3\},\{2\}\}} = \sum_{Occ(w)=\{\{1,3\},\{2\}\}} w$$

Partitioning maps (Condition 1/3) - 2

Lemma

$S_\varphi(\mathfrak{A}) := \left\langle m_\sigma \mid m_\sigma := \sum_{\varphi(w)=\sigma} w \right\rangle$ has bases indexed by $\mathcal{C}$.

$$m_{\{\{1,3\},\{2\}\}} = \sum_{Occ(w)=\{\{1,3\},\{2\}\}} w = \sum_{\substack{\mathfrak{a},\mathfrak{b}\in\mathfrak{A} \\ \mathfrak{a}\neq\mathfrak{b}}} \mathfrak{a}\mathfrak{b}\mathfrak{a}$$

Partitioning maps (Condition 1/3) - 2

Lemma

$S_\varphi(\mathfrak{A}) := \left\langle m_\sigma \mid m_\sigma := \sum_{\varphi(w)=\sigma} w \right\rangle$ has bases indexed by $\mathcal{C}$.

$$m_{\{\{1,3\},\{2\}\}} = \sum_{Occ(w)=\{\{1,3\},\{2\}\}} w = \sum_{\substack{\mathfrak{a},\mathfrak{b}\in\mathfrak{A} \\ \mathfrak{a}\neq\mathfrak{b}}} \mathfrak{aba}$$

$$= \mathtt{aba} + \mathtt{aca} + \cdots + \mathtt{bab} + \mathtt{bcb} + \mathtt{bdb} + \cdots$$

Partitioning maps (Condition 1/3) - 2

Lemma

$S_\varphi(\mathfrak{A}) := \left\langle m_\sigma \mid m_\sigma := \sum_{\varphi(w)=\sigma} w \right\rangle$ has bases indexed by $\mathcal{C}$.

$$m_{\{\{1,3\},\{2\}\}} = \sum_{Occ(w)=\{\{1,3\},\{2\}\}} w = \sum_{\substack{a,b\in\mathfrak{A}\\ a\neq b}} ab a$$

$$= \mathtt{aba} + \mathtt{aca} + \cdots + \mathtt{bab} + \mathtt{bcb} + \mathtt{bdb} + \cdots$$

$\rightarrow (m_\pi)_{\pi\in\mathbb{S}}$: basis indexed by *set partition*.

Partitioning maps (Condition 2/3)

Definition (Condition 2)

If $\varphi(uv) = \varphi(u'v')$ *then* $\begin{cases} \varphi(u) = \varphi(u')\,, \\ \varphi(v) = \varphi(v')\,. \end{cases}$

(for $|u| = |u'|$ et $|v| = |v'|$)

Partitioning maps (Condition 2/3)

Definition (Condition 2)

$$\textit{If} \quad \varphi(uv) = \varphi(u'v') \quad \textit{then} \quad \begin{cases} \varphi(u) = \varphi(u')\,, \\ \varphi(v) = \varphi(v')\,. \end{cases}$$

(for $|u| = |u'|$ et $|v| = |v'|$)

$$\mathit{Occ}(\texttt{jgezoi} \cdot \texttt{nobgzegio}) = \mathit{Occ}(\texttt{okfipg} \cdot \texttt{npbkifkgp})$$
$$= \{\{1\}, \{2, 10, 13\}, \{3, 12\}, \{4, 11\}, \{5, 8, 15\}, \{6, 14\}, \{7\}, \{9\}\}$$

Partitioning maps (Condition 2/3)

Definition (Condition 2)

$$\text{If} \quad \varphi(uv) = \varphi(u'v') \quad \text{then} \quad \begin{cases} \varphi(u) = \varphi(u'), \\ \varphi(v) = \varphi(v'). \end{cases}$$

(for $|u| = |u'|$ et $|v| = |v'|$)

$$\begin{aligned} Occ(\texttt{jgezoi} \cdot \texttt{nobgzegio}) &= Occ(\texttt{okfipg} \cdot \texttt{npbkifkgp}) \\ &= \{\{1\}, \{2, 10, 13\}, \{3, 12\}, \{4, 11\}, \{5, 8, 15\}, \{6, 14\}, \{7\}, \{9\}\} \end{aligned}$$

$$Occ(\texttt{jgezoi}) = Occ(\texttt{okfipg})?$$

Partitioning maps (Condition 2/3)

Definition (Condition 2)

$$\textit{If} \quad \varphi(uv) = \varphi(u'v') \quad \textit{then} \quad \begin{cases} \varphi(u) = \varphi(u'), \\ \varphi(v) = \varphi(v'). \end{cases}$$

(for $|u| = |u'|$ et $|v| = |v'|$)

$$\begin{aligned} Occ(\texttt{jgezoi} \cdot \texttt{nobgzegio}) &= Occ(\texttt{okfipg} \cdot \texttt{npbkifkgp}) \\ &= \{\{1\}, \{2, 10, 13\}, \{3, 12\}, \{4, 11\}, \{5, 8, 15\}, \{6, 14\}, \{7\}, \{9\}\} \end{aligned}$$

$$\underbrace{Occ(\texttt{jgezoi}) = Occ(\texttt{okfipg})}_{\{\{1\},\{2,\cancel{10,13}\},\{3,\cancel{12}\},\{4,\cancel{11}\},\{5,\cancel{8,15}\},\{6,\cancel{14}\},\cancel{\{7\},\{9\}}\}}$$

Partitioning maps (Condition 2/3)

Definition (Condition 2)

If $\varphi(uv) = \varphi(u'v')$ *then* $\begin{cases} \varphi(u) = \varphi(u')\,, \\ \varphi(v) = \varphi(v')\,. \end{cases}$

(for $|u| = |u'|$ et $|v| = |v'|$)

$$\mathit{Occ}(\mathtt{jgezoi} \cdot \mathtt{nobgzegio}) = \mathit{Occ}(\mathtt{okfipg} \cdot \mathtt{npbkifkgp})$$
$$= \{\{1\}, \{2, 10, 13\}, \{3, 12\}, \{4, 11\}, \{5, 8, 15\}, \{6, 14\}, \{7\}, \{9\}\}$$

$$\underbrace{\mathit{Occ}(\mathtt{jgezoi}) = \mathit{Occ}(\mathtt{okfipg})}_{\{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\}\}}$$

Partitioning maps (Condition 2/3)

Definition (Condition 2)

$$\text{If} \quad \varphi(uv) = \varphi(u'v') \quad \text{then} \quad \begin{cases} \varphi(u) = \varphi(u'), \\ \varphi(v) = \varphi(v'). \end{cases}$$

(for $|u| = |u'|$ et $|v| = |v'|$)

$$\begin{aligned} Occ(\texttt{jgezoi} \cdot \texttt{nobgzegio}) &= Occ(\texttt{okfipg} \cdot \texttt{npbkifkgp}) \\ &= \{\{1\}, \{2, 10, 13\}, \{3, 12\}, \{4, 11\}, \{5, 8, 15\}, \{6, 14\}, \{7\}, \{9\}\} \end{aligned}$$

$$\underbrace{Occ(\texttt{jgezoi}) = Occ(\texttt{okfipg})}_{\{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\}\}} \quad \text{and}$$

Partitioning maps (Condition 2/3)

Definition (Condition 2)

If $\varphi(uv) = \varphi(u'v')$ *then* $\begin{cases} \varphi(u) = \varphi(u')\,, \\ \varphi(v) = \varphi(v')\,. \end{cases}$

(for $|u| = |u'|$ et $|v| = |v'|$)

$$\mathit{Occ}(\mathtt{jgezoi} \cdot \mathtt{nobgzegio}) = \mathit{Occ}(\mathtt{okfipg} \cdot \mathtt{npbkifkgp})$$
$$= \{\{1\}, \{2, 10, 13\}, \{3, 12\}, \{4, 11\}, \{5, 8, 15\}, \{6, 14\}, \{7\}, \{9\}\}$$

$$\underbrace{\mathit{Occ}(\mathtt{jgezoi}) = \mathit{Occ}(\mathtt{okfipg})}_{\{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\}\}} \quad \text{and} \quad \underbrace{\mathit{Occ}(\mathtt{nobgzegio}) = \mathit{Occ}(\mathtt{npbkifkgp})}_{\{\{10,13\},\{12\},\{11\},\{8,15\},\{14\},\{7\},\{9\}\}[-6]}$$

Partitioning maps (Condition 2/3)

Definition (Condition 2)

$$\textit{If} \quad \varphi(uv) = \varphi(u'v') \quad \textit{then} \quad \begin{cases} \varphi(u) = \varphi(u')\,, \\ \varphi(v) = \varphi(v')\,. \end{cases}$$

(for $|u| = |u'|$ et $|v| = |v'|$)

$$\begin{aligned} \mathit{Occ}(\texttt{jgezoi} \cdot \texttt{nobgzegio}) &= \mathit{Occ}(\texttt{okfipg} \cdot \texttt{npbkifkgp}) \\ &= \{\{1\}, \{2, 10, 13\}, \{3, 12\}, \{4, 11\}, \{5, 8, 15\}, \{6, 14\}, \{7\}, \{9\}\} \end{aligned}$$

$$\underbrace{\mathit{Occ}(\texttt{jgezoi}) = \mathit{Occ}(\texttt{okfipg})}_{\{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\}\}} \quad \text{and} \quad \underbrace{\mathit{Occ}(\texttt{nobgzegio}) = \mathit{Occ}(\texttt{npbkifkgp})}_{\{\{1\},\{2,9\},\{3\},\{4,7\},\{5\},\{6\},\{8\}\}}$$

Partitioning maps (Condition 2/3)

Definition (Condition 2)

$$\text{If} \quad \varphi(uv) = \varphi(u'v') \quad \text{then} \quad \begin{cases} \varphi(u) = \varphi(u'), \\ \varphi(v) = \varphi(v'). \end{cases}$$

(for $|u| = |u'|$ et $|v| = |v'|$)

$$\mathit{Occ}(\mathtt{jgezoi} \cdot \mathtt{nobgzegio}) = \mathit{Occ}(\mathtt{okfipg} \cdot \mathtt{npbkifkgp})$$
$$= \{\{1\}, \{2, 10, 13\}, \{3, 12\}, \{4, 11\}, \{5, 8, 15\}, \{6, 14\}, \{7\}, \{9\}\}$$

$$\underbrace{\mathit{Occ}(\mathtt{jgezoi}) = \mathit{Occ}(\mathtt{okfipg})}_{1/2/3/4/5/6} \quad \text{and} \quad \underbrace{\mathit{Occ}(\mathtt{nobgzegio}) = \mathit{Occ}(\mathtt{npbkifkgp})}_{1/29/3/47/5/6/8}.$$

Partitioning maps (Condition 2/3) - 2

Lemma

$S_\varphi(\mathfrak{A})$ is a sub-algebra of $\mathbb{K}\langle\mathfrak{A}\rangle$.

$$m_\sigma m_\mu = \sum_{\substack{\exists uv \in \varphi^{-1}(\tau) \\ \varphi(u)=\sigma;\ \ \varphi(v)=\mu}} m_\tau \quad \text{for} \quad \tau \in \mathcal{C}.$$

Partitioning maps (Condition 2/3) - 2

Lemma

$S_\varphi(\mathfrak{A})$ is a sub-algebra of $\mathbb{K}\langle\mathfrak{A}\rangle$.

$$m_\sigma m_\mu = \sum_{\substack{\exists uv \in \varphi^{-1}(\tau) \\ \varphi(u)=\sigma;\ \ \varphi(v)=\mu}} m_\tau \quad \text{for} \quad \tau \in \mathcal{C}.$$

$$m_{13/2} \cdot m_{1/2} = \left(\sum_{\substack{a,b \in \mathfrak{A} \\ a \neq b}} aba \right) \cdot \left(\sum_{\substack{c,d \in \mathfrak{A} \\ c \neq d}} cd \right)$$

Partitioning maps (Condition 2/3) - 2

Lemma

$S_\varphi(\mathfrak{A})$ is a sub-algebra of $\mathbb{K}\langle\mathfrak{A}\rangle$.

$$m_\sigma m_\mu = \sum_{\substack{\exists uv \in \varphi^{-1}(\tau) \\ \varphi(u)=\sigma;\ \ \varphi(v)=\mu}} m_\tau \quad \text{for} \quad \tau \in \mathcal{C}.$$

$$m_{13/2} \cdot m_{1/2} = \left(\sum_{\substack{a,b \in \mathfrak{A} \\ a \neq b}} aba \right) \cdot \left(\sum_{\substack{c,d \in \mathfrak{A} \\ c \neq d}} cd \right) = \sum_{\substack{a,b,c,d \in \mathfrak{A} \\ a \neq b; c \neq d}} abacd$$

Partitioning maps (Condition 2/3) - 2

Lemma

$S_\varphi(\mathfrak{A})$ is a sub-algebra of $\mathbb{K}\langle\mathfrak{A}\rangle$.

$$m_\sigma m_\mu = \sum_{\substack{\exists uv \in \varphi^{-1}(\tau) \\ \varphi(u)=\sigma;\ \ \varphi(v)=\mu}} m_\tau \quad \text{for} \quad \tau \in \mathcal{C}.$$

$$m_{13/2} \cdot m_{1/2} = \left(\sum_{\substack{a,b \in \mathfrak{A} \\ a \neq b}} aba \right) \cdot \left(\sum_{\substack{c,d \in \mathfrak{A} \\ c \neq d}} cd \right) = \sum_{\substack{a,b,c,d \in \mathfrak{A} \\ a \neq b; c \neq d}} abacd$$

$$= \sum_{\substack{a,b,c,d \in \mathfrak{A} \\ a \neq b \neq c \neq d}} abacd + \sum_{\substack{a,b,c,d \in \mathfrak{A} \\ a \neq b; a=c; b=d}} abacd + \sum_{\substack{a,b,c,d \in \mathfrak{A} \\ a \neq b; a=d; b=c}} abacd + \cdots$$

Partitioning maps (Condition 2/3) - 2

Lemma

$S_\varphi(\mathfrak{A})$ is a sub-algebra of $\mathbb{K}\langle\mathfrak{A}\rangle$.

$$m_\sigma m_\mu = \sum_{\substack{\exists uv \in \varphi^{-1}(\tau) \\ \varphi(u)=\sigma;\ \varphi(v)=\mu}} m_\tau \quad \text{for} \quad \tau \in \mathcal{C}.$$

$$m_{13/2} \cdot m_{1/2} = \Big(\sum_{\substack{a,b\in\mathfrak{A} \\ a\neq b}} aba\Big) \cdot \Big(\sum_{\substack{c,d\in\mathfrak{A} \\ c\neq d}} cd\Big) = \sum_{\substack{a,b,c,d\in\mathfrak{A} \\ a\neq b; c\neq d}} abacd$$

$$= \sum_{\substack{a,b,c,d\in\mathfrak{A} \\ a\neq b\neq c\neq d}} abacd + \sum_{\substack{a,b,c,d\in\mathfrak{A} \\ a\neq b; a=c; b=d}} abacd + \sum_{\substack{a,b,c,d\in\mathfrak{A} \\ a\neq b; a=d; b=c}} abacd + \cdots$$

$$= m_{13/2/4/5} + m_{134/25} + m_{135/24} + m_{134/2/5} + m_{13/24/5} + \cdots$$

Partitioning maps (Condition 2/3) - 2

Lemma

$S_\varphi(\mathfrak{A})$ is a sub-algebra of $\mathbb{K}\langle\mathfrak{A}\rangle$.

$$m_\sigma m_\mu = \sum_{\substack{\exists uv \in \varphi^{-1}(\tau) \\ \varphi(u)=\sigma;\ \ \varphi(v)=\mu}} m_\tau \quad \text{for} \quad \tau \in \mathcal{C}.$$

$$m_{13/2} \cdot m_{1/2} = \Big(\sum_{\substack{\mathfrak{a},\mathfrak{b}\in\mathfrak{A} \\ \mathfrak{a}\neq\mathfrak{b}}} \mathfrak{aba} \Big) \cdot \Big(\sum_{\substack{\mathfrak{c},\mathfrak{d}\in\mathfrak{A} \\ \mathfrak{c}\neq\mathfrak{d}}} \mathfrak{cd} \Big) = \sum_{\substack{\mathfrak{a},\mathfrak{b},\mathfrak{c},\mathfrak{d}\in\mathfrak{A} \\ \mathfrak{a}\neq\mathfrak{b};\mathfrak{c}\neq\mathfrak{d}}} \mathfrak{abacd}$$

$$= \sum_{\substack{\mathfrak{a},\mathfrak{b},\mathfrak{c},\mathfrak{d}\in\mathfrak{A} \\ \mathfrak{a}\neq\mathfrak{b}\neq\mathfrak{c}\neq\mathfrak{d}}} \mathfrak{abacd} + \sum_{\substack{\mathfrak{a},\mathfrak{b},\mathfrak{c},\mathfrak{d}\in\mathfrak{A} \\ \mathfrak{a}\neq\mathfrak{b};\mathfrak{a}=\mathfrak{c};\mathfrak{b}=\mathfrak{d}}} \mathfrak{abacd} + \sum_{\substack{\mathfrak{a},\mathfrak{b},\mathfrak{c},\mathfrak{d}\in\mathfrak{A} \\ \mathfrak{a}\neq\mathfrak{b};\mathfrak{a}=\mathfrak{d};\mathfrak{b}=\mathfrak{c}}} \mathfrak{abacd} + \cdots$$

$$= m_{13/2/4/5} + m_{134/25} + m_{135/24} + m_{134/2/5} + m_{13/24/5} + \cdots$$

$\rightarrow \simeq$ ***WSym*** as algebra.

Type of alphabet **Alph**

For example :

$$\text{red integers } (\mathbb{N}, \leqslant) \quad \text{and} \quad \text{blue integers } (\mathbb{N}, \leqslant);$$

such that $\forall x, y \in \mathbb{N}$,

$$\zeta(x) \leqslant \zeta(y) \quad \text{whenever} \quad x \leqslant y;$$

and $\mathfrak{A} := \mathbb{N} \oplus \mathbb{N}$ endowed with the *order structure* $\leqslant$:

$$x \leqslant y \iff \begin{cases} x \leqslant y & \text{with } x, y \in \mathbb{N}, \\ x \leqslant y & \text{with } x, y \in \mathbb{N}, \text{ or} \\ x \in \mathbb{N} & \text{and} \quad y \in \mathbb{N}. \end{cases}$$

Type of alphabet **Alph**

For example :

$$\text{red integers } (\mathbb{N}, \leqslant) \quad \text{and} \quad \text{blue integers } (\mathbb{N}, \leqslant);$$

such that $\forall x, y \in \mathbb{N}$,

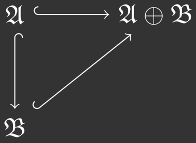
$$\zeta(x) \leqslant \zeta(y) \quad \text{whenever} \quad x \leqslant y;$$

and $\mathfrak{A} := \mathbb{N} \oplus \mathbb{N}$ endowed with the *order structure* $\leqslant$:

$$x \leqslant y \iff \begin{cases} x \leqslant y & \text{with } x, y \in \mathbb{N}, \\ x \leqslant y & \text{with } x, y \in \mathbb{N}, \text{ or} \\ x \in \mathbb{N} \quad \text{and} & y \in \mathbb{N}. \end{cases}$$

Ingredients :

Formally : *category* of *sets* with *injective maps* and $\oplus$.



Partitioning maps (3/3)

Let $\mathfrak{A} := \mathfrak{B} \oplus \mathfrak{C}$ be an alphabet.

Definition (Condition 3)

For any $b, b' \in \mathfrak{B}^m$ and any $c, c' \in \mathfrak{C}^n$,

$$\textit{if} \begin{cases} \varphi(b) = \varphi(b') \\ \varphi(c) = \varphi(c') \end{cases} \quad \textit{then}\ \varphi((bc)\sigma) = \varphi((b'c')\sigma)$$

for $\sigma \in 1 \cdots m \sqcup\!\sqcup 1 \cdots n$.

Partitioning maps (3/3)

Let $\mathfrak{A} := \mathfrak{B} \oplus \mathfrak{C}$ be an alphabet.

Definition (Condition 3)

For any $b, b' \in \mathfrak{B}^m$ and any $c, c' \in \mathfrak{C}^n$,

$$\textit{if} \begin{cases} \varphi(b) = \varphi(b') \\ \varphi(c) = \varphi(c') \end{cases} \textit{then } \varphi((bc)\sigma) = \varphi((b'c')\sigma)$$

for $\sigma \in 1 \cdots m \sqcup\!\!\sqcup 1 \cdots n$.

Exercise : show that *Occ* satisfies Condition 3.

Partitioning maps (3/3)

Let $\mathfrak{A} := \mathfrak{B} \oplus \mathfrak{C}$ be an alphabet.

Definition (Condition 3)

For any $b, b' \in \mathfrak{B}^m$ and any $c, c' \in \mathfrak{C}^n$,

$$\textit{if } \begin{cases} \varphi(b) = \varphi(b') \\ \varphi(c) = \varphi(c') \end{cases} \quad \textit{then } \varphi((bc)\sigma) = \varphi((b'c')\sigma)$$

for $\sigma \in 1 \cdots m \sqcup\!\!\sqcup 1 \cdots n$.

Exercise : show that *Occ* satisfies Condition 3.
Hint : define $\pi \cdot \sigma$ such that $\varphi(w\sigma) = \varphi(w)\sigma$.

Partitioning maps (Condition 3/3) - 2

Theorem

$S_\varphi(\mathfrak{A})$ is endowed with a *Hopf algebra structure*.

Alphabet doubling trick : for $\mathfrak{A} := \mathfrak{B} \oplus \mathfrak{C}$,

$$\Delta(m_\sigma) \simeq \delta_{\mathfrak{B} \rightleftarrows \mathfrak{C}} \left(\sum_{\varphi(w)=\sigma} w \right) = \sum_{\varphi(w)=\sigma} w_{|\mathfrak{B}} \otimes w_{|\mathfrak{C}} \,.$$

Partitioning maps (Condition 3/3) - 2

Theorem

$S_\varphi(\mathfrak{A})$ is endowed with a *Hopf algebra structure*.

Alphabet doubling trick : for $\mathfrak{A} := \mathfrak{B} \oplus \mathfrak{C}$,

$$\Delta(m_\sigma) \simeq \delta_{\mathfrak{B}\rightleftarrows\mathfrak{C}}\Big(\sum_{\varphi(w)=\sigma} w\Big) = \sum_{\varphi(w)=\sigma} w_{|\mathfrak{B}} \otimes w_{|\mathfrak{C}}\,.$$

$$\Delta(m_{13/2}) \simeq \delta_{\mathfrak{B}\rightleftarrows\mathfrak{C}}\Big(\sum_{Occ(w)=13/2} w\Big)$$

Partitioning maps (Condition 3/3) - 2

Theorem

$S_\varphi(\mathfrak{A})$ is endowed with a *Hopf algebra structure*.

Alphabet doubling trick : for $\mathfrak{A} := \mathfrak{B} \oplus \mathfrak{C}$,

$$\Delta(m_\sigma) \simeq \delta_{\mathfrak{B}\rightleftarrows\mathfrak{C}}\left(\sum_{\varphi(w)=\sigma} w \right) = \sum_{\varphi(w)=\sigma} w_{|\mathfrak{B}} \otimes w_{|\mathfrak{C}} \,.$$

$$\Delta(m_{13/2}) \simeq \delta_{\mathfrak{B}\rightleftarrows\mathfrak{C}}\left(\sum_{Occ(w)=13/2} w \right) = \delta_{\mathfrak{B}\rightleftarrows\mathfrak{C}}\left(\sum_{a\neq b} aba + aba + aba + aba \right)$$

Partitioning maps (Condition 3/3) - 2

Theorem

$S_\varphi(\mathfrak{A})$ is endowed with a *Hopf algebra structure*.

Alphabet doubling trick : for $\mathfrak{A} := \mathfrak{B} \oplus \mathfrak{C}$,

$$\Delta(m_\sigma) \simeq \delta_{\mathfrak{B}\rightleftarrows\mathfrak{C}}\left(\sum_{\varphi(w)=\sigma} w\right) = \sum_{\varphi(w)=\sigma} w_{|\mathfrak{B}} \otimes w_{|\mathfrak{C}} .$$

$$\Delta(m_{13/2}) \simeq \delta_{\mathfrak{B}\rightleftarrows\mathfrak{C}}\left(\sum_{Occ(w)=13/2} w\right) = \delta_{\mathfrak{B}\rightleftarrows\mathfrak{C}}\left(\sum_{\mathfrak{a}\neq\mathfrak{b}} \mathfrak{aba} + \mathfrak{aba} + \mathfrak{aba} + \mathfrak{aba}\right)$$

$$\simeq m_{13/2} \otimes 1 + m_{12} \otimes m_1 + m_1 \otimes m_{12} + 1 \otimes m_{13/2}$$

Partitioning maps (Condition 3/3) - 2

Theorem

$S_\varphi(\mathfrak{A})$ is endowed with a *Hopf algebra structure*.

Alphabet doubling trick : for $\mathfrak{A} := \mathfrak{B} \oplus \mathfrak{C}$,

$$\Delta(m_\sigma) \simeq \delta_{\mathfrak{B}\rightleftarrows\mathfrak{C}}\left(\sum_{\varphi(w)=\sigma} w \right) = \sum_{\varphi(w)=\sigma} w_{|\mathfrak{B}} \otimes w_{|\mathfrak{C}} \,.$$

$$\Delta(m_{13/2}) \simeq \delta_{\mathfrak{B}\rightleftarrows\mathfrak{C}}\left(\sum_{Occ(w)=13/2} w \right) = \delta_{\mathfrak{B}\rightleftarrows\mathfrak{C}}\left(\sum_{a\neq b} aba + aba + aba + aba \right)$$

$$\simeq m_{13/2} \otimes 1 + m_{12} \otimes m_1 + m_1 \otimes m_{12} + 1 \otimes m_{13/2}$$

$\rightarrow \simeq$ ***WSym*** as Hopf algebra.

Some examples

φ-map	structure on $\mathfrak{A}$	Hopf algebra
Occ	none	**WSym**
Std	totally ordered	**FQSym**
BST	totally ordered	**PBT**
Pack	totally ordered	**WQSym**
Park	totally ordered + successor	**PQSym**
...	...	...

(a_n)-Parking Functions

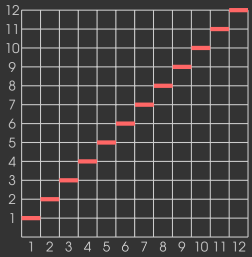
Let $(a_n)_{n\in\mathbb{N}}$ be an *increasing positive sequence* with $a_1 = 1$.

(a_n)-Parking Functions

Let $(a_n)_{n \in \mathbb{N}}$ be an *increasing positive sequence* with $a_1 = 1$.

For example : for $n > 0$,

- $a_n := n$,

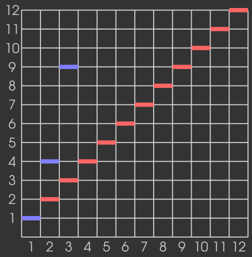


(a_n)-Parking Functions

Let $(a_n)_{n \in \mathbb{N}}$ be an *increasing positive sequence* with $a_1 = 1$.

For example : for $n > 0$,

- $a_n := n$,
- $b_n := n^2$,

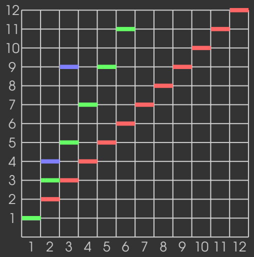


(a_n)-Parking Functions

Let $(a_n)_{n\in\mathbb{N}}$ be an *increasing positive sequence* with $a_1 = 1$.

For example : for $n > 0$,

- $a_n := n$,
- $b_n := n^2$,
- $c_n := c_{n-1} + 2,$
  $= 2(n-1) + 1,$

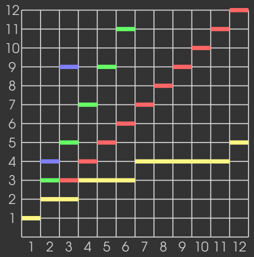


(a_n)-Parking Functions

Let $(a_n)_{n\in\mathbb{N}}$ be an *increasing positive sequence* with $a_1 = 1$.

For example : for $n > 0$,

- $a_n := n$,
- $b_n := n^2$,
- $c_n := c_{n-1} + 2$,
  $= 2(n-1) + 1$,
- $d_n := \lfloor \log((n+1)^2) \rfloor$.

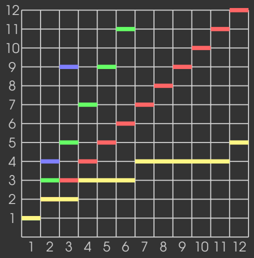


(a_n)-Parking Functions

Let $(a_n)_{n\in\mathbb{N}}$ be an *increasing positive sequence* with $a_1 = 1$.

For example : for $n > 0$,

- $a_n := n$,
- $b_n := n^2$,
- $c_n := c_{n-1} + 2,$ $\quad = 2(n-1)+1$,
- $d_n := \lfloor \log((n+1)^2) \rfloor$.



Definition (PITMAN-STANLEY, 99)

Let f be a word on $\mathbb{N}_{>0}$ of size k.
Let $f^{\uparrow}$ be the *non-decreasing arrangment* of f.

The word f is an *(a_n)-parking function* ($f \in \mathbf{PF}_{(a_n)}$) if

$$\forall i \in [k], \quad f_i^{\uparrow} \leqslant a_i .$$

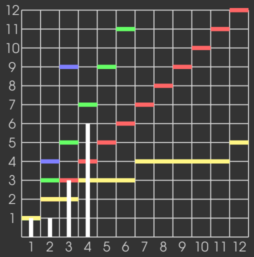
(a_n)-Parking Functions

Let $(a_n)_{n\in\mathbb{N}}$ be an *increasing positive sequence* with $a_1 = 1$.

For example : for $n > 0$,

- $a_n := n$,
- $b_n := n^2$,
- $c_n := c_{n-1} + 2$, $= 2(n-1) + 1$,
- $d_n := \lfloor \log((n+1)^2) \rfloor$.

$f :=$ 3161 :



Definition (Pitman-Stanley, 99)

Let f be a word on $\mathbb{N}_{>0}$ of size k.
Let $f^{\uparrow}$ be the *non-decreasing arrangment* of f.

The word f is an *(a_n)-parking function* ($f \in \mathbf{PF}_{(a_n)}$) if

$$\forall i \in [k], \quad f_i^{\uparrow} \leqslant a_i .$$

(a_n)-Parking Functions

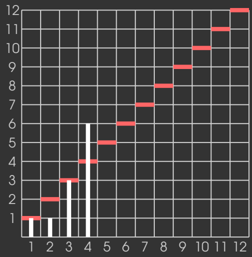
Let $(a_n)_{n\in\mathbb{N}}$ be an *increasing positive sequence* with $a_1 = 1$.

For example : for $n > 0$,

- $a_n := n$,
- $b_n := n^2$,
- $c_n := c_{n-1} + 2,$
  $= 2(n-1) + 1,$
- $d_n := \lfloor \log((n+1)^2) \rfloor$.

$f :=$ 3161 :

- $f \notin \mathbf{PF}_{(a_n)}$,



Definition (Pitman-Stanley, 99)

Let f be a word on $\mathbb{N}_{>0}$ of size k.
Let $f^{\uparrow}$ be the *non-decreasing arrangment* of f.

The word f is an *(a_n)-parking function* ($f \in \mathbf{PF}_{(a_n)}$) if

$$\forall i \in [k], \quad f_i^{\uparrow} \leqslant a_i \,.$$

(a_n)-Parking Functions

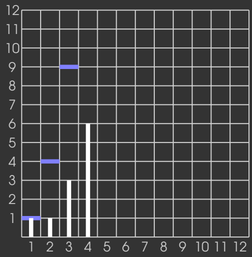
Let $(a_n)_{n\in\mathbb{N}}$ be an *increasing positive sequence* with $a_1 = 1$.

For example : for $n > 0$,

- $a_n := n$,
- $b_n := n^2$,
- $c_n := c_{n-1} + 2,$
  $= 2(n-1) + 1,$
- $d_n := \lfloor \log((n+1)^2) \rfloor$.

$f :=$ `3161` :

- $f \notin$ **PF**$_{(a_n)}$,
- $f \in$ **PF**$_{(b_n)}$,



Definition (Pitman-Stanley, 99)

Let f be a word on $\mathbb{N}_{>0}$ of size k.
Let $f^{\uparrow}$ be the *non-decreasing arrangment* of f.

The word f is an *(a_n)-parking function* ($f \in$ **PF**$_{(a_n)}$) if

$$\forall i \in [k], \quad f_i^{\uparrow} \leqslant a_i .$$

(a_n)-Parking Functions

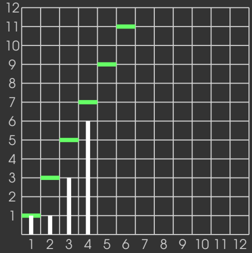
Let $(a_n)_{n\in\mathbb{N}}$ be an *increasing positive sequence* with $a_1 = 1$.

For example : for $n > 0$,

- $a_n := n$,
- $b_n := n^2$,
- $c_n := c_{n-1} + 2$,
  $= 2(n-1) + 1$,
- $d_n := \lfloor \log((n+1)^2) \rfloor$.

$f :=$ `3161` :

- $f \notin \mathbf{PF}_{(a_n)}$,
- $f \in \mathbf{PF}_{(b_n)}$,
- $f \in \mathbf{PF}_{(c_n)}$,



Definition (Pitman-Stanley, 99)

Let f be a word on $\mathbb{N}_{>0}$ of size k.
Let $f^{\uparrow}$ be the *non-decreasing arrangment* of f.

The word f is an *(a_n)-parking function* ($f \in \mathbf{PF}_{(a_n)}$) if

$$\forall i \in [k], \quad f_i^{\uparrow} \leqslant a_i .$$

(a_n)-Parking Functions

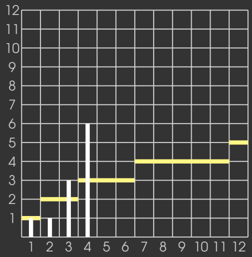
Let $(a_n)_{n\in\mathbb{N}}$ be an *increasing positive sequence* with $a_1 = 1$.

For example : for $n > 0$,

- $a_n := n$,
- $b_n := n^2$,
- $c_n := c_{n-1} + 2$,
  $= 2(n-1) + 1$,
- $d_n := \lfloor \log((n+1)^2) \rfloor$.

$f :=$ `3161` :

- $f \notin \mathbf{PF}_{(a_n)}$,
- $f \in \mathbf{PF}_{(b_n)}$,
- $f \in \mathbf{PF}_{(c_n)}$,
- $f \notin \mathbf{PF}_{(d_n)}$.



Definition (Pitman-Stanley, 99)

Let f be a word on $\mathbb{N}_{>0}$ of size k.
Let $f^{\uparrow}$ be the *non-decreasing arrangment* of f.

The word f is an *(a_n)-parking function* ($f \in \mathbf{PF}_{(a_n)}$) if

$$\forall i \in [k], \quad f^{\uparrow}_i \leqslant a_i \,.$$

(a_n)-Parkization

Algorithm park :

Input: $f \in \mathbb{N}^*_{>0}$ a word,
(a_n) an increasing sequence with $a_1 = 1$.

Result: $f \in \mathbf{PF}_{(a_n)}$

while $\exists k$ *such that* $f^{\uparrow}_k > a_k$ **do**
  $k \leftarrow \min(k \mid f^{\uparrow}_k > a_k)$;
  decrement f all letters $\ell > a_k$ of f;
return f

(a_n)-Parkization

Algorithm park :

Input: $f \in \mathbb{N}^*_{>0}$ a word,
(a_n) an increasing sequence with $a_1 = 1$.
Result: $f \in \mathbf{PF}_{(a_n)}$

while $\exists k$ *such that* $f^{\uparrow}_k > a_k$ **do**
  $k \leftarrow \min(k \mid f^{\uparrow}_k > a_k)$;
  decrement f all letters $\ell > a_k$ of f;
return f

Let $f := 4 \cdot 7 \cdot 9 \cdot 8 \cdot 23 \cdot 1 \cdot 72 \cdot 22 \cdot 7 \cdot 8 \cdot 11$ be a word.
Let (c_n) be a sequel defined by $\begin{cases} c_1 = 1\,, \\ c_n = c_{n-1} + 2\,. \end{cases}$

$$park(f, (c_n)) \quad =$$

(a_n)-Parkization

Algorithm park :

Input: $f \in \mathbb{N}^*_{>0}$ a word,
(a_n) an increasing sequence with $a_1 = 1$.
Result: $f \in \mathbf{PF}_{(a_n)}$

while $\exists k$ *such that* $f^{\uparrow}_k > a_k$ **do**
    $k \leftarrow \min(k \mid f^{\uparrow}_k > a_k)$;
    decrement f all letters $\ell > a_k$ of f;
return f

Let $f := 4 \cdot 7 \cdot 9 \cdot 8 \cdot 23 \cdot 1 \cdot 72 \cdot 22 \cdot 7 \cdot 8 \cdot 11$ be a word.

Let (c_n) be a sequel defined by $\begin{cases} c_1 = 1\,, \\ c_n = c_{n-1} + 2\,. \end{cases}$

$$park(f, (c_n)) = 4 \cdot 7 \cdot 9 \cdot 8 \cdot 23 \cdot 1 \cdot 72 \cdot 22 \cdot 7 \cdot 8 \cdot 11$$

i	1	2	3	4	5	6	7	8	9	10	11
$f^{\uparrow}_i$	1	4	7	7	8	8	9	11	22	23	72
c_i	1	3	5	7	9	11	13	15	17	19	21

(a_n)-Parkization

Algorithm park :

Input: $f \in \mathbb{N}^*_{>0}$ a word,
(a_n) an increasing sequence with $a_1 = 1$.
Result: $f \in \mathbf{PF}_{(a_n)}$

1 **while** $\exists k$ *such that* $f^{\uparrow}_k > a_k$ **do**
2 $k \leftarrow \min(k \mid f^{\uparrow}_k > a_k)$;
3 decrement f all letters $\ell > a_k$ of f;
4 **return** f

Let $f := 4 \cdot 7 \cdot 9 \cdot 8 \cdot 23 \cdot 1 \cdot 72 \cdot 22 \cdot 7 \cdot 8 \cdot 11$ be a word.
Let (c_n) be a sequel defined by $\begin{cases} c_1 = 1\,, \\ c_n = c_{n-1} + 2\,. \end{cases}$

$$park(f, (c_n)) \quad = 4 \cdot 7 \cdot 9 \cdot 8 \cdot 23 \cdot 1 \cdot 72 \cdot 22 \cdot 7 \cdot 8 \cdot 11$$

i	1	2	3	4	5	6	7	8	9	10	11
$f^{\uparrow}_i$	1	4	7	7	8	8	9	11	22	23	72
c_i	1	3	5	7	9	11	13	15	17	19	21

(a_n)-Parkization

Algorithm park :

Input: $f \in \mathbb{N}^*_{>0}$ a word,
(a_n) an increasing sequence with $a_1 = 1$.
Result: $f \in \mathbf{PF}_{(a_n)}$

```
while ∃k such that f_k^↑ > a_k do
    k ← min(k | f_k^↑ > a_k);
    decrement f all letters ℓ > a_k of f;
return f
```

Let $f := 4 \cdot 7 \cdot 9 \cdot 8 \cdot 23 \cdot 1 \cdot 72 \cdot 22 \cdot 7 \cdot 8 \cdot 11$ be a word.

Let (c_n) be a sequel defined by $\begin{cases} c_1 = 1\,, \\ c_n = c_{n-1} + 2\,. \end{cases}$

$$park(f, (c_n)) \quad = 3 \cdot 6 \cdot 8 \cdot 7 \cdot 22 \cdot 1 \cdot 71 \cdot 21 \cdot 6 \cdot 7 \cdot 10$$

i	1	2	3	4	5	6	7	8	9	10	11
$f_i^{\uparrow}$	1	3	6	6	7	7	8	10	21	22	71
c_i	1	3	5	7	9	11	13	15	17	19	21

(a_n)-Parkization

Algorithm park :

Input: $f \in \mathbb{N}^*_{>0}$ a word,
(a_n) an increasing sequence with $a_1 = 1$.
Result: $f \in \mathbf{PF}_{(a_n)}$

while $\exists k$ *such that* $f_k^{\uparrow} > a_k$ **do**
    $k \leftarrow \min(k \mid f_k^{\uparrow} > a_k)$;
    decrement f all letters $\ell > a_k$ of f;
return f

Let $f := 4 \cdot 7 \cdot 9 \cdot 8 \cdot 23 \cdot 1 \cdot 72 \cdot 22 \cdot 7 \cdot 8 \cdot 11$ be a word.
Let (c_n) be a sequel defined by $\begin{cases} c_1 = 1\,, \\ c_n = c_{n-1} + 2\,. \end{cases}$

$$park(f, (c_n)) = 3 \cdot 5 \cdot 7 \cdot 6 \cdot 21 \cdot 1 \cdot 70 \cdot 20 \cdot 5 \cdot 6 \cdot 9$$

i	1	2	3	4	5	6	7	8	9	10	11
$f_i^{\uparrow}$	1	3	5	5	6	6	7	9	20	21	70
c_i	1	3	5	7	9	11	13	15	17	19	21

(a_n)-Parkization

Algorithm park :

Input: $f \in \mathbb{N}^*_{>0}$ a word,
(a_n) an increasing sequence with $a_1 = 1$.

Result: $f \in \mathbf{PF}_{(a_n)}$

while $\exists k$ *such that* $f^{\uparrow}_k > a_k$ **do**
    $k \leftarrow \min(k \mid f^{\uparrow}_k > a_k)$;
    decrement f all letters $\ell > a_k$ of f;
return f

Let $f := 4 \cdot 7 \cdot 9 \cdot 8 \cdot 23 \cdot 1 \cdot 72 \cdot 22 \cdot 7 \cdot 8 \cdot 11$ be a word.

Let (c_n) be a sequel defined by $\begin{cases} c_1 = 1\,, \\ c_n = c_{n-1} + 2\,. \end{cases}$

$$park(f, (c_n)) \quad = 3 \cdot 5 \cdot 7 \cdot 6 \cdot 18 \cdot 1 \cdot 67 \cdot 17 \cdot 5 \cdot 6 \cdot 9$$

i	1	2	3	4	5	6	7	8	9	10	11
$f^{\uparrow}_i$	1	3	5	5	6	6	7	9	17	18	67
c_i	1	3	5	7	9	11	13	15	17	19	21

(a_n)-Parkization

Algorithm park :

Input: $f \in \mathbb{N}^*_{>0}$ a word,
(a_n) an increasing sequence with $a_1 = 1$.

Result: $f \in \mathbf{PF}_{(a_n)}$

while $\exists k$ *such that* $f^{\uparrow}_k > a_k$ **do**
    $k \leftarrow \min(k \mid f^{\uparrow}_k > a_k)$;
    decrement f all letters $\ell > a_k$ of f;
return f

Let $f := 4 \cdot 7 \cdot 9 \cdot 8 \cdot 23 \cdot 1 \cdot 72 \cdot 22 \cdot 7 \cdot 8 \cdot 11$ be a word.

Let (c_n) be a sequel defined by $\begin{cases} c_1 = 1\,, \\ c_n = c_{n-1} + 2\,. \end{cases}$

$$park(f, (c_n)) = 3 \cdot 5 \cdot 7 \cdot 6 \cdot 18 \cdot 1 \cdot 21 \cdot 17 \cdot 5 \cdot 6 \cdot 9$$

i	1	2	3	4	5	6	7	8	9	10	11
$f^{\uparrow}_i$	1	3	5	5	6	6	7	9	17	18	21
c_i	1	3	5	7	9	11	13	15	17	19	21

(a_n)-Parkization (2)

Theorem

The *(a_n)-parkization* is a *partitioning map*

if and only if

$$(a_n) := \begin{cases} a_1 = 1\,, \\ a_k = a_{k-1} + b \end{cases} \quad \text{for } k > 1 \text{ with } b \in \mathbb{N}.$$

Special case

For $a_k = k$, one obtains ***PQSym***.

Concluding pitch

Formalism :

3 conditions on words,

Concluding pitch

Formalism :

3 conditions on words,
\+ *1* operation on alphabets,

Concluding pitch

Formalism :

3 conditions on words,
+ *1* operation on alphabets,

---

combinatorial Hopf algebra ! ! !

(*versus* 6 complicated mathematical axioms.)

Concluding pitch

Formalism :

3 conditions on words,
+ 1 operation on alphabets,

---

combinatorial Hopf algebra ! ! !

(*versus* 6 complicated mathematical axioms.)

Application :

- New family of Hopf algebras indexed by $\mathbf{PF}_{(a_n)}$.

Perspectives

Plactic like monoids :

- $\mathtt{b} \cdot \mathtt{w} \cdot \mathtt{a} \equiv \mathtt{a} \cdot \mathtt{w} \cdot \mathtt{b}$ if $\mathtt{a}, \mathtt{b}$ have the same breakpoint :

$$S(t) = 1 + t + 3t^2 + 14t^3 + 87t^4 + 674t^5 + 6260t^6 + \dots$$

Perspectives

Plactic like monoids :

- $\mathtt{b} \cdot \mathtt{w} \cdot \mathtt{a} \equiv \mathtt{a} \cdot \mathtt{w} \cdot \mathtt{b}$ if $\mathtt{a}, \mathtt{b}$ have the same breakpoint :

$$S(t) = 1 + t + 3t^2 + 14t^3 + 87t^4 + 674t^5 + 6260t^6 + \dots$$

Formalize some others specializations ? :

- Difference, product of alphabets, . . .

Perspectives

Plactic like monoids :

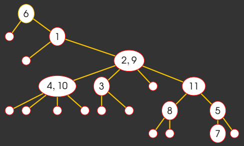
- $\texttt{b} \cdot \texttt{w} \cdot \texttt{a} \equiv \texttt{a} \cdot \texttt{w} \cdot \texttt{b}$ if $\texttt{a}, \texttt{b}$ have the same breakpoint :

$$S(t) = 1 + t + 3t^2 + 14t^3 + 87t^4 + 674t^5 + 6260t^6 + \dots$$

Formalize some others specializations ? :

- Difference, product of alphabets, ...

Links with Bérénice Oger work ? :

- $3 \cdot 5 \cdot 7 \cdot 6 \cdot 18 \cdot 1 \cdot 21 \cdot 17 \cdot 5 \cdot 6 \cdot 9 \in \mathbf{PF}^{11}_{(a_n)} \leftrightarrow$  $\in$ boxed trees.