# Algebraic and combinatorial structures on Baxter permutations

Samuele Giraudo

Université de Marne-la-Vallée

66th Séminaire Lotharingien de Combinatoire
March 8, 2011

# Contents

Hopf algebra of permutations and construction of subalgebras
- The Hopf algebra of permutations
- Equivalence relations and quotients of the free monoid
- More structure on quotient monoids
- From quotient monoids to Hopf subalgebras

Algebraic constructions on Baxter permutations
- The Baxter combinatorial family
- The Baxter monoid
- A Robinson-Schensted-like correspondence
- The Baxter lattice
- The Baxter Hopf algebra

# Contents

Hopf algebra of permutations and construction of subalgebras
The Hopf algebra of permutations
Equivalence relations and quotients of the free monoid
More structure on quotient monoids
From quotient monoids to Hopf subalgebras

# The vector space of permutations

Let $\mathbf{FQSym}_n$ be the $\mathbb{Q}$-vector space based on permutations of $\{1, \ldots, n\}$ and

$$\mathbf{FQSym} := \bigoplus_{n \geq 0} \mathbf{FQSym}_n,$$

be the vector space of permutations.

Its elements can be handled through the fundamental basis $\{\mathbf{F}_\sigma\}_{\sigma \in \mathfrak{S}}$.

A product and a coproduct can be added to this structure to form the Hopf algebra of Free quasi-symmetric functions, also known as the Malvenuto-Reutenauer Hopf algebra.

# A product in **FQSym**

**FQSym** is endowed by the shifted shuffle product:

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

# A product in **FQSym**

**FQSym** is endowed by the shifted shuffle product:

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

For example:

$$\mathbf{F}_{12} \cdot \mathbf{F}_{21} =$$

# A product in **FQSym**

**FQSym** is endowed by the shifted shuffle product:

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

For example:

$$\mathbf{F}_{12} \cdot \mathbf{F}_{21} = \mathbf{F}_{\bullet\bullet\bullet\bullet} + \mathbf{F}_{\bullet\bullet\bullet\bullet} + \mathbf{F}_{\bullet\bullet\bullet\bullet} + \mathbf{F}_{\bullet\bullet\bullet\bullet} + \mathbf{F}_{\bullet\bullet\bullet\bullet} + \mathbf{F}_{\bullet\bullet\bullet\bullet}.$$

# A product in **FQSym**

**FQSym** is endowed by the shifted shuffle product:

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

For example:

$$\mathbf{F}_{12} \cdot \mathbf{F}_{21} = \mathbf{F}_{12\bullet\bullet} + \mathbf{F}_{\bullet\bullet\bullet\bullet} + \mathbf{F}_{\bullet\bullet\bullet\bullet} + \mathbf{F}_{\bullet\bullet\bullet\bullet} + \mathbf{F}_{\bullet\bullet\bullet\bullet} + \mathbf{F}_{\bullet\bullet\bullet\bullet}.$$

# A product in **FQSym**

**FQSym** is endowed by the shifted shuffle product:

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

For example:

$$\mathbf{F}_{12} \cdot \mathbf{F}_{21} = \mathbf{F}_{12\bullet\bullet} + \mathbf{F}_{1\bullet 2\bullet} + \mathbf{F}_{\bullet\bullet\bullet\bullet} + \mathbf{F}_{\bullet\bullet\bullet\bullet} + \mathbf{F}_{\bullet\bullet\bullet\bullet} + \mathbf{F}_{\bullet\bullet\bullet\bullet} .$$

# A product in **FQSym**

**FQSym** is endowed by the shifted shuffle product:

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

For example:

$$\mathbf{F}_{12} \cdot \mathbf{F}_{21} = \mathbf{F}_{12\bullet\bullet} + \mathbf{F}_{1\bullet2\bullet} + \mathbf{F}_{1\bullet\bullet2} + \mathbf{F}_{\bullet\bullet\bullet\bullet} + \mathbf{F}_{\bullet\bullet\bullet\bullet} + \mathbf{F}_{\bullet\bullet\bullet\bullet}.$$

# A product in **FQSym**

**FQSym** is endowed by the shifted shuffle product:

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

For example:

$$\mathbf{F}_{12} \cdot \mathbf{F}_{21} = \mathbf{F}_{12\bullet\bullet} + \mathbf{F}_{1\bullet 2\bullet} + \mathbf{F}_{1\bullet\bullet 2} + \mathbf{F}_{\bullet 12\bullet} + \mathbf{F}_{\bullet\bullet\bullet\bullet} + \mathbf{F}_{\bullet\bullet\bullet\bullet}.$$

# A product in **FQSym**

**FQSym** is endowed by the shifted shuffle product:

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

For example:

$$\mathbf{F}_{12} \cdot \mathbf{F}_{21} = \mathbf{F}_{12\bullet\bullet} + \mathbf{F}_{1\bullet2\bullet} + \mathbf{F}_{1\bullet\bullet2} + \mathbf{F}_{\bullet12\bullet} + \mathbf{F}_{\bullet1\bullet2} + \mathbf{F}_{\bullet\bullet\bullet\bullet}.$$

# A product in **FQSym**

**FQSym** is endowed by the shifted shuffle product:

$$\mathbf{F}_\sigma \cdot \mathbf{F}_\nu := \sum_{\pi \in \sigma \overline{⧢} \nu} \mathbf{F}_\pi .$$

For example:

$$\mathbf{F}_{12} \cdot \mathbf{F}_{21} = \mathbf{F}_{12\bullet\bullet} + \mathbf{F}_{1\bullet 2\bullet} + \mathbf{F}_{1\bullet\bullet 2} + \mathbf{F}_{\bullet 12\bullet} + \mathbf{F}_{\bullet 1\bullet 2} + \mathbf{F}_{\bullet\bullet 12} .$$

# A product in **FQSym**

**FQSym** is endowed by the shifted shuffle product:

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

For example:

$$\mathbf{F}_{12} \cdot \mathbf{F}_{21} = \mathbf{F}_{1243} + \mathbf{F}_{1\bullet 2\bullet} + \mathbf{F}_{1\bullet\bullet 2} + \mathbf{F}_{\bullet 12\bullet} + \mathbf{F}_{\bullet 1\bullet 2} + \mathbf{F}_{\bullet\bullet 12}.$$

# A product in **FQSym**

**FQSym** is endowed by the shifted shuffle product:

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

For example:

$$\mathbf{F}_{12} \cdot \mathbf{F}_{21} = \mathbf{F}_{1243} + \mathbf{F}_{1423} + \mathbf{F}_{1\bullet\bullet 2} + \mathbf{F}_{\bullet 12\bullet} + \mathbf{F}_{\bullet 1\bullet 2} + \mathbf{F}_{\bullet\bullet 12}.$$

# A product in **FQSym**

**FQSym** is endowed by the shifted shuffle product:

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

For example:

$$\mathbf{F}_{12} \cdot \mathbf{F}_{21} = \mathbf{F}_{1243} + \mathbf{F}_{1423} + \mathbf{F}_{1432} + \mathbf{F}_{\bullet 12 \bullet} + \mathbf{F}_{\bullet 1 \bullet 2} + \mathbf{F}_{\bullet \bullet 12}.$$

# A product in **FQSym**

**FQSym** is endowed by the shifted shuffle product:

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

For example:

$$\mathbf{F}_{12} \cdot \mathbf{F}_{21} = \mathbf{F}_{1243} + \mathbf{F}_{1423} + \mathbf{F}_{1432} + \mathbf{F}_{4123} + \mathbf{F}_{\bullet 1 \bullet 2} + \mathbf{F}_{\bullet\bullet 12}.$$

# A product in **FQSym**

**FQSym** is endowed by the shifted shuffle product:

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

For example:

$$\mathbf{F}_{12} \cdot \mathbf{F}_{21} = \mathbf{F}_{1243} + \mathbf{F}_{1423} + \mathbf{F}_{1432} + \mathbf{F}_{4123} + \mathbf{F}_{4132} + \mathbf{F}_{\bullet\bullet 12}.$$

# A product in **FQSym**

**FQSym** is endowed by the shifted shuffle product:

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

For example:

$$\mathbf{F}_{12} \cdot \mathbf{F}_{21} = \mathbf{F}_{1243} + \mathbf{F}_{1423} + \mathbf{F}_{1432} + \mathbf{F}_{4123} + \mathbf{F}_{4132} + \mathbf{F}_{4312}.$$

# A product in **FQSym**

**FQSym** is endowed by the shifted shuffle product:

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

For example:

$$\mathbf{F}_{12} \cdot \mathbf{F}_{21} = \mathbf{F}_{1243} + \mathbf{F}_{1423} + \mathbf{F}_{1432} + \mathbf{F}_{4123} + \mathbf{F}_{4132} + \mathbf{F}_{4312}.$$

- This product is associative and non-commutative;
- It admits $\mathbf{F}_\epsilon$ as unit;
- It is graded: $\cdot : \mathbf{FQSym}_n \otimes \mathbf{FQSym}_m \to \mathbf{FQSym}_{n+m}$.

Hence, $(\mathbf{FQSym}, \cdot)$ is a graded unital associative algebra.

# Right permutohedron order

Let $\sigma, \nu \in \mathfrak{S}_n$. $\sigma$ is covered by $\nu$ if $\sigma = u\mathtt{a}\mathtt{b}v$ and $\nu = u\mathtt{b}\mathtt{a}v$ where $\mathtt{a} < \mathtt{b}$. This covering relation spans the right permutohedron.

# Right permutohedron order

Let $\sigma, \nu \in \mathfrak{S}_n$. $\sigma$ is covered by $\nu$ if $\sigma = u\mathtt{a}\mathtt{b}v$ and $\nu = u\mathtt{b}\mathtt{a}v$ where $\mathtt{a} < \mathtt{b}$. This covering relation spans the right permutohedron.

Here is that of permutations of size 4:

1234

# Right permutohedron order

Let $\sigma, \nu \in \mathfrak{S}_n$. $\sigma$ is covered by $\nu$ if $\sigma = u\mathtt{a}\mathtt{b}v$ and $\nu = u\mathtt{b}\mathtt{a}v$ where $\mathtt{a} < \mathtt{b}$. This covering relation spans the right permutohedron.

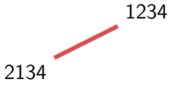
Here is that of permutations of size 4:

1234

# Right permutohedron order

Let $\sigma, \nu \in \mathfrak{S}_n$. $\sigma$ is covered by $\nu$ if $\sigma = u\mathtt{a}\mathtt{b}v$ and $\nu = u\mathtt{b}\mathtt{a}v$ where $\mathtt{a} < \mathtt{b}$. This covering relation spans the right permutohedron.

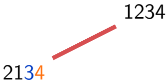
Here is that of permutations of size 4:



# Right permutohedron order

Let $\sigma, \nu \in \mathfrak{S}_n$. $\sigma$ is covered by $\nu$ if $\sigma = u\mathtt{a}\mathtt{b}v$ and $\nu = u\mathtt{b}\mathtt{a}v$ where $\mathtt{a} < \mathtt{b}$. This covering relation spans the right permutohedron.

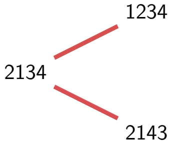
Here is that of permutations of size 4:



# Right permutohedron order

Let $\sigma, \nu \in \mathfrak{S}_n$. $\sigma$ is covered by $\nu$ if $\sigma = u\mathtt{a}\mathtt{b}v$ and $\nu = u\mathtt{b}\mathtt{a}v$ where $\mathtt{a} < \mathtt{b}$. This covering relation spans the right permutohedron.

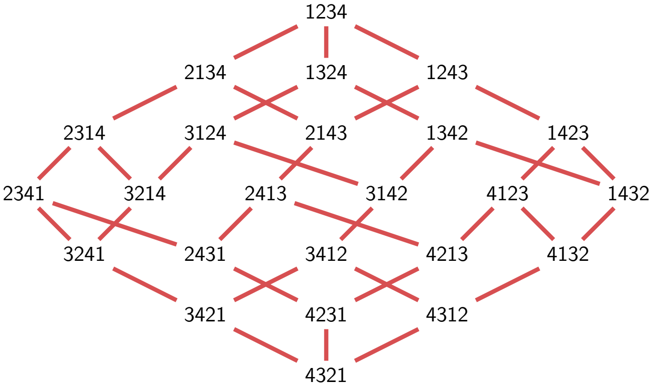
Here is that of permutations of size 4:



# Right permutohedron order

Let $\sigma, \nu \in \mathfrak{S}_n$. $\sigma$ is covered by $\nu$ if $\sigma = u\mathtt{a}\mathtt{b}v$ and $\nu = u\mathtt{b}\mathtt{a}v$ where $\mathtt{a} < \mathtt{b}$. This covering relation spans the right permutohedron.

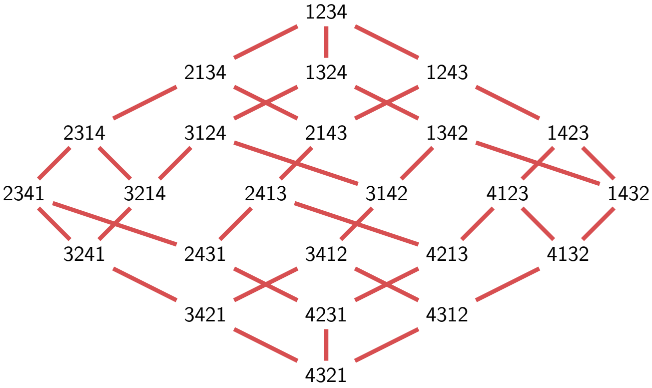
Here is that of permutations of size 4:



# Right permutohedron order

Let $\sigma, \nu \in \mathfrak{S}_n$. $\sigma$ is covered by $\nu$ if $\sigma = u\mathtt{a}\mathtt{b}v$ and $\nu = u\mathtt{b}\mathtt{a}v$ where $\mathtt{a} < \mathtt{b}$. This covering relation spans the right permutohedron.

Here is that of permutations of size 4:




The elements $\mathbf{F}_\pi$ appearing in a product $\mathbf{F}_\sigma \cdot \mathbf{F}_\nu$ form an interval of the permutohedron.

# Standardization process

Let $A := \{\mathtt{a} < \mathtt{b} < \mathtt{c} < \dots\}$ be a totally ordered infinite alphabet.

The standardization process std is an algorithm computing a permutation $\sigma$ from a word $u \in A^*$ such that $\sigma$ has the same inversions than $u$.

# Standardization process

Let $A := \{\mathtt{a} < \mathtt{b} < \mathtt{c} < \dots\}$ be a totally ordered infinite alphabet.

The standardization process std is an algorithm computing a permutation $\sigma$ from a word $u \in A^*$ such that $\sigma$ has the same inversions than $u$.

For example, the computation of std($\mathtt{cabbdaabd}$) gives:

# Standardization process

Let $A := \{\mathtt{a} < \mathtt{b} < \mathtt{c} < \dots\}$ be a totally ordered infinite alphabet.

The standardization process std is an algorithm computing a permutation $\sigma$ from a word $u \in A^*$ such that $\sigma$ has the same inversions than $u$.

For example, the computation of std(cabbdaabd) gives:

c a b b d a a b d

# Standardization process

Let $A := \{\mathtt{a} < \mathtt{b} < \mathtt{c} < \dots\}$ be a totally ordered infinite alphabet.

The standardization process std is an algorithm computing a permutation $\sigma$ from a word $u \in A^*$ such that $\sigma$ has the same inversions than $u$.

For example, the computation of std(cabbdaabd) gives:

| c | a | b | b | d | a | a | b | d |
|---|---|---|---|---|---|---|---|---|
| | 1 | | | | | | | |

# Standardization process

Let $A := \{\mathtt{a} < \mathtt{b} < \mathtt{c} < \ldots\}$ be a totally ordered infinite alphabet.

The standardization process std is an algorithm computing a permutation $\sigma$ from a word $u \in A^*$ such that $\sigma$ has the same inversions than $u$.

For example, the computation of std(cabbdaabd) gives:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| c | a | b | b | d | a | a | b | d |
| | 1 | | | | 2 | | | |

# Standardization process

Let $A := \{\mathtt{a} < \mathtt{b} < \mathtt{c} < \ldots\}$ be a totally ordered infinite alphabet.

The standardization process std is an algorithm computing a permutation $\sigma$ from a word $u \in A^*$ such that $\sigma$ has the same inversions than $u$.

For example, the computation of std(cabbdaabd) gives:

| c | a | b | b | d | a | a | b | d |
|---|---|---|---|---|---|---|---|---|
| | 1 | | | | 2 | 3 | | |

# Standardization process

Let $A := \{\mathtt{a} < \mathtt{b} < \mathtt{c} < \dots\}$ be a totally ordered infinite alphabet.

The standardization process std is an algorithm computing a permutation $\sigma$ from a word $u \in A^*$ such that $\sigma$ has the same inversions than $u$.

For example, the computation of std(cabbdaabd) gives:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| c | a | b | b | d | a | a | b | d |
| | 1 | | | | 2 | 3 | | |
| | | 4 | | | | | | |

# Standardization process

Let $A := \{\mathtt{a} < \mathtt{b} < \mathtt{c} < \dots\}$ be a totally ordered infinite alphabet.

The standardization process std is an algorithm computing a permutation $\sigma$ from a word $u \in A^*$ such that $\sigma$ has the same inversions than $u$.

For example, the computation of std(cabbdaabd) gives:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| c | a | b | b | d | a | a | b | d |
| | 1 | | | | 2 | 3 | | |
| | | 4 | 5 | | | | | |

# Standardization process

Let $A := \{\mathtt{a} < \mathtt{b} < \mathtt{c} < \ldots\}$ be a totally ordered infinite alphabet.

The standardization process std is an algorithm computing a permutation $\sigma$ from a word $u \in A^*$ such that $\sigma$ has the same inversions than $u$.

For example, the computation of std(cabbdaabd) gives:

| c | a | b | b | d | a | a | b | d |
|---|---|---|---|---|---|---|---|---|
| | 1 | | | | 2 | 3 | | |
| | | 4 | 5 | | | | 6 | |

# Standardization process

Let $A := \{\mathtt{a} < \mathtt{b} < \mathtt{c} < \dots\}$ be a totally ordered infinite alphabet.

The standardization process std is an algorithm computing a permutation $\sigma$ from a word $u \in A^*$ such that $\sigma$ has the same inversions than $u$.

For example, the computation of std(cabbdaabd) gives:

| c | a | b | b | d | a | a | b | d |
|---|---|---|---|---|---|---|---|---|
| | 1 | | | | 2 | 3 | | |
| | | 4 | 5 | | | | 6 | |
| 7 | | | | | | | | |

# Standardization process

Let $A := \{\mathtt{a} < \mathtt{b} < \mathtt{c} < \dots\}$ be a totally ordered infinite alphabet.

The standardization process std is an algorithm computing a permutation $\sigma$ from a word $u \in A^*$ such that $\sigma$ has the same inversions than $u$.

For example, the computation of std(cabbdaabd) gives:

| c | a | b | b | d | a | a | b | d |
|---|---|---|---|---|---|---|---|---|
| | 1 | | | | 2 | 3 | | |
| | | 4 | 5 | | | | 6 | |
| 7 | | | | 8 | | | | |

# Standardization process

Let $A := \{\mathtt{a} < \mathtt{b} < \mathtt{c} < \dots\}$ be a totally ordered infinite alphabet.

The standardization process std is an algorithm computing a permutation $\sigma$ from a word $u \in A^*$ such that $\sigma$ has the same inversions than $u$.

For example, the computation of std(cabbdaabd) gives:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| c | a | b | b | d | a | a | b | d |
| | 1 | | | | 2 | 3 | | |
| | | 4 | 5 | | | | 6 | |
| 7 | | | | 8 | | | | 9 |

# Standardization process

Let $A := \{\mathtt{a} < \mathtt{b} < \mathtt{c} < \ldots\}$ be a totally ordered infinite alphabet.

The standardization process std is an algorithm computing a permutation $\sigma$ from a word $u \in A^*$ such that $\sigma$ has the same inversions than $u$.

For example, the computation of std($\mathtt{cabbdaabd}$) gives:

| c | a | b | b | d | a | a | b | d |
|---|---|---|---|---|---|---|---|---|
| | 1 | | | | 2 | 3 | | |
| | | 4 | 5 | | | | 6 | |
| 7 | | | | 8 | | | | 9 |
| | | | | | | | | |
| 7 | 1 | 4 | 5 | 8 | 2 | 3 | 6 | 9 |

# A coproduct in **FQSym**

**FQSym** is endowed by the deconcatenation coproduct:

$$\Delta\left(\mathbf{F}_{\sigma}\right) := \sum_{u.v=\sigma} \mathbf{F}_{\mathsf{std}(u)} \otimes \mathbf{F}_{\mathsf{std}(v)}.$$

# A coproduct in **FQSym**

**FQSym** is endowed by the deconcatenation coproduct:

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

For example:

$\Delta\left(\mathbf{F}_{4123}\right) =$

# A coproduct in **FQSym**

**FQSym** is endowed by the deconcatenation coproduct:

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

For example:

$$\Delta\left(\mathbf{F}_{|4123}\right) = \mathbf{F}_{\epsilon} \otimes \mathbf{F}_{4123}$$

# A coproduct in **FQSym**

**FQSym** is endowed by the deconcatenation coproduct:

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

For example:

$$\Delta\left(\mathbf{F}_{4|123}\right) = \mathbf{F}_{\epsilon} \otimes \mathbf{F}_{4123} + \mathbf{F}_{1} \otimes \mathbf{F}_{123}$$

# A coproduct in **FQSym**

**FQSym** is endowed by the deconcatenation coproduct:

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

For example:

$$\Delta\left(\mathbf{F}_{41|23}\right) = \mathbf{F}_{\epsilon} \otimes \mathbf{F}_{4123} + \mathbf{F}_{1} \otimes \mathbf{F}_{123} + \mathbf{F}_{21} \otimes \mathbf{F}_{12}$$

# A coproduct in **FQSym**

**FQSym** is endowed by the deconcatenation coproduct:

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

For example:

$$\Delta\left(\mathbf{F}_{412|3}\right) = \mathbf{F}_\epsilon \otimes \mathbf{F}_{4123} + \mathbf{F}_1 \otimes \mathbf{F}_{123} + \mathbf{F}_{21} \otimes \mathbf{F}_{12} + \mathbf{F}_{312} \otimes \mathbf{F}_1$$

# A coproduct in **FQSym**

**FQSym** is endowed by the deconcatenation coproduct:

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

For example:

$$\Delta\left(\mathbf{F}_{4123|}\right) = \mathbf{F}_{\epsilon} \otimes \mathbf{F}_{4123} + \mathbf{F}_{1} \otimes \mathbf{F}_{123} + \mathbf{F}_{21} \otimes \mathbf{F}_{12} + \mathbf{F}_{312} \otimes \mathbf{F}_{1} + \mathbf{F}_{4123} \otimes \mathbf{F}_{\epsilon}.$$

# A coproduct in **FQSym**

**FQSym** is endowed by the deconcatenation coproduct:

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

For example:

$$\Delta\left(\mathbf{F}_{4123}\right) = \mathbf{F}_\epsilon \otimes \mathbf{F}_{4123} + \mathbf{F}_1 \otimes \mathbf{F}_{123} + \mathbf{F}_{21} \otimes \mathbf{F}_{12} + \mathbf{F}_{312} \otimes \mathbf{F}_1 + \mathbf{F}_{4123} \otimes \mathbf{F}_\epsilon.$$

- This coproduct is coassociative;
- It is non-cocommutative;
- It is graded: $\Delta : \mathbf{FQSym}_n \to \bigoplus_{i+j=n} \mathbf{FQSym}_i \otimes \mathbf{FQSym}_j$.

Hence, $(\mathbf{FQSym}, \Delta)$ is a graded coassociative coalgebra.

# **FQSym** as a combinatorial Hopf algebra

These algebra and coalgebra structures are compatible, *i.e.*, $\Delta$ is an algebra morphism:

$$\Delta(\mathbf{F}_\sigma \cdot \mathbf{F}_\nu) = \Delta(\mathbf{F}_\sigma) \cdot \Delta(\mathbf{F}_\nu).$$

Hence, **FQSym** is a bialgebra.

Since **FQSym** is graded and connected, it is a Hopf algebra.

We use the heuristic term of combinatorial Hopf algebra (CHA) for graded and connected Hopf algebras based on combinatorial objects.

# Contents

Hopf algebra of permutations and construction of subalgebras
The Hopf algebra of permutations
Equivalence relations and quotients of the free monoid
More structure on quotient monoids
From quotient monoids to Hopf subalgebras

# Equivalence relations on words

We define equivalence relations on words of $A^*$ by taking the reflexive, symmetric, and transitive closure of a rewriting rule $\rightleftarrows$.

We are interested by equivalence relations on $A^*$ that are congruences:

**Definition**

The equivalence relation $\equiv$ is a congruence if for all $u, u', v, v' \in A^*$,

$$u \equiv v \quad \text{and} \quad u' \equiv v' \qquad \text{imply} \qquad u.u' \equiv v.v'.$$

In this way, $A^*/_{\equiv}$ is a quotient monoid of the free monoid.

# An example: The plactic equivalence relation

For example, the rewriting rule

$$\ldots \mathtt{acb} \ldots \rightleftarrows \ldots \mathtt{cab} \ldots \qquad \text{if } \mathtt{a} \leq \mathtt{b} < \mathtt{c},$$
$$\ldots \mathtt{bac} \ldots \rightleftarrows \ldots \mathtt{bca} \ldots \qquad \text{if } \mathtt{a} < \mathtt{b} \leq \mathtt{c},$$

defines the well-known plactic equivalence relation $\equiv_{\mathsf{P}}$ . $A^*/_{\equiv_{\mathsf{P}}}$ is the plactic monoid.

# An example: The plactic equivalence relation

For example, the rewriting rule

$$\ldots \mathtt{acb} \ldots \rightleftarrows \ldots \mathtt{cab} \ldots \qquad \text{if } \mathtt{a} \leq \mathtt{b} < \mathtt{c},$$
$$\ldots \mathtt{bac} \ldots \rightleftarrows \ldots \mathtt{bca} \ldots \qquad \text{if } \mathtt{a} < \mathtt{b} \leq \mathtt{c},$$

defines the well-known plactic equivalence relation $\equiv_{\mathsf{P}}$. $A^*/_{\equiv_{\mathsf{P}}}$ is the plactic monoid.

Here is the plactic equivalence class of 31542:

31542

# An example: The plactic equivalence relation

For example, the rewriting rule

$$\ldots \mathtt{acb} \ldots \rightleftarrows \ldots \mathtt{cab} \ldots \qquad \text{if } \mathtt{a} \leq \mathtt{b} < \mathtt{c},$$
$$\ldots \mathtt{bac} \ldots \rightleftarrows \ldots \mathtt{bca} \ldots \qquad \text{if } \mathtt{a} < \mathtt{b} \leq \mathtt{c},$$

defines the well-known plactic equivalence relation $\equiv_{\mathsf{P}}$. $A^*/_{\equiv_{\mathsf{P}}}$ is the plactic monoid.

Here is the plactic equivalence class of 31542:

31542

# An example: The plactic equivalence relation

For example, the rewriting rule

$$\ldots \mathtt{acb} \ldots \rightleftarrows \ldots \mathtt{cab} \ldots \qquad \text{if } \mathtt{a} \leq \mathtt{b} < \mathtt{c},$$
$$\ldots \mathtt{bac} \ldots \rightleftarrows \ldots \mathtt{bca} \ldots \qquad \text{if } \mathtt{a} < \mathtt{b} \leq \mathtt{c},$$

defines the well-known plactic equivalence relation $\equiv_{\mathsf{P}}$. $A^*/_{\equiv_{\mathsf{P}}}$ is the plactic monoid.

Here is the plactic equivalence class of 31542:

31542

|

35142

# An example: The plactic equivalence relation

For example, the rewriting rule

$$\ldots \mathtt{acb} \ldots \rightleftarrows \ldots \mathtt{cab} \ldots \qquad \text{if } \mathtt{a} \leq \mathtt{b} < \mathtt{c},$$
$$\ldots \mathtt{bac} \ldots \rightleftarrows \ldots \mathtt{bca} \ldots \qquad \text{if } \mathtt{a} < \mathtt{b} \leq \mathtt{c},$$

defines the well-known plactic equivalence relation $\equiv_{\mathrm{P}}$. $A^*/_{\equiv_{\mathrm{P}}}$ is the plactic monoid.

Here is the plactic equivalence class of 31542:

31542
|
35142

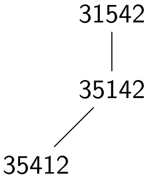
# An example: The plactic equivalence relation

For example, the rewriting rule

$$\dots \mathtt{acb} \dots \rightleftarrows \dots \mathtt{cab} \dots \qquad \text{if } \mathtt{a} \leq \mathtt{b} < \mathtt{c},$$
$$\dots \mathtt{bac} \dots \rightleftarrows \dots \mathtt{bca} \dots \qquad \text{if } \mathtt{a} < \mathtt{b} \leq \mathtt{c},$$

defines the well-known plactic equivalence relation $\equiv_{\mathsf{P}}$. $A^*/_{\equiv_{\mathsf{P}}}$ is the plactic monoid.

Here is the plactic equivalence class of 31542:



# An example: The plactic equivalence relation

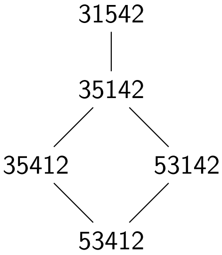
For example, the rewriting rule

$$\ldots \mathtt{acb} \ldots \rightleftarrows \ldots \mathtt{cab} \ldots \qquad \text{if } \mathtt{a} \leq \mathtt{b} < \mathtt{c},$$

$$\ldots \mathtt{bac} \ldots \rightleftarrows \ldots \mathtt{bca} \ldots \qquad \text{if } \mathtt{a} < \mathtt{b} \leq \mathtt{c},$$

defines the well-known plactic equivalence relation $\equiv_{\mathsf{P}}$. $A^*/_{\equiv_{\mathsf{P}}}$ is the plactic monoid.

Here is the plactic equivalence class of 31542:



# Contents


Hopf algebra of permutations and construction of subalgebras

The Hopf algebra of permutations

Equivalence relations and quotients of the free monoid

More structure on quotient monoids

From quotient monoids to Hopf subalgebras

# Compatibility with the destandardization process

Denote by $\mathrm{ev}(u)$ the non-decreasing rearrangement of $u$. For example:

$$\mathrm{ev}(\mathtt{babcaac}) = \mathtt{aaabbcc}.$$

## Definition

Let $\equiv$ be a congruence. The monoid $A^*/_\equiv$ is compatible with the destandardization process if for all $u, v \in A^*$,

$$u \equiv v \qquad \text{iff} \qquad \mathrm{std}(u) \equiv \mathrm{std}(v) \quad \text{and} \quad \mathrm{ev}(u) = \mathrm{ev}(v).$$

# Compatibility with the restriction of alphabet intervals

Let $S \subseteq A$ and $u$ be a word. Denote by $u_{|S}$ the longest subword of $u$ made of letters of $S$. For example:

$$\mathtt{bcacca}_{|\{\mathtt{a},\mathtt{b}\}} = \mathtt{baa}.$$

## Definition

Let $\equiv$ be a congruence. The monoid $A^*/_\equiv$ is compatible with the restriction of alphabet intervals if for all interval $L$ of $A$ and $u, v \in A^*$,

$$u \equiv v \qquad \text{implies} \qquad u_{|L} \equiv v_{|L}.$$

# Contents

Hopf algebra of permutations and construction of subalgebras
The Hopf algebra of permutations
Equivalence relations and quotients of the free monoid
More structure on quotient monoids
From quotient monoids to Hopf subalgebras

# Construction of Hopf subalgebras of **FQSym**

Let $\equiv$ be an equivalence relation. For all equivalence class $C$ of $\mathfrak{S}/_{\equiv}$, let us define the following element of **FQSym**:

$$\mathbf{P}_C := \sum_{\sigma \in C} \mathbf{F}_\sigma .$$

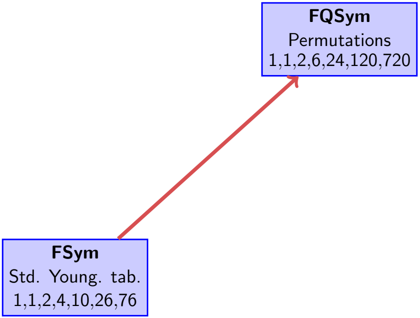
**Theorem** [**Hivert**, **Nzeutchap**, 2007]

If $\equiv$ is a congruence and $A^*/_{\equiv}$ is compatible with the destandardization process and with the restriction of alphabet intervals, then, the family $\{\mathbf{P}_C\}_{C \in \mathfrak{S}/_{\equiv}}$ spans a Hopf subalgebra of **FQSym**.

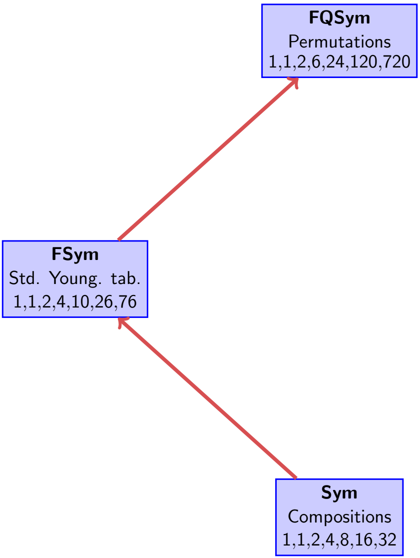
# Some Hopf subalgebras of **FQSym**

**FQSym**
Permutations
1,1,2,6,24,120,720

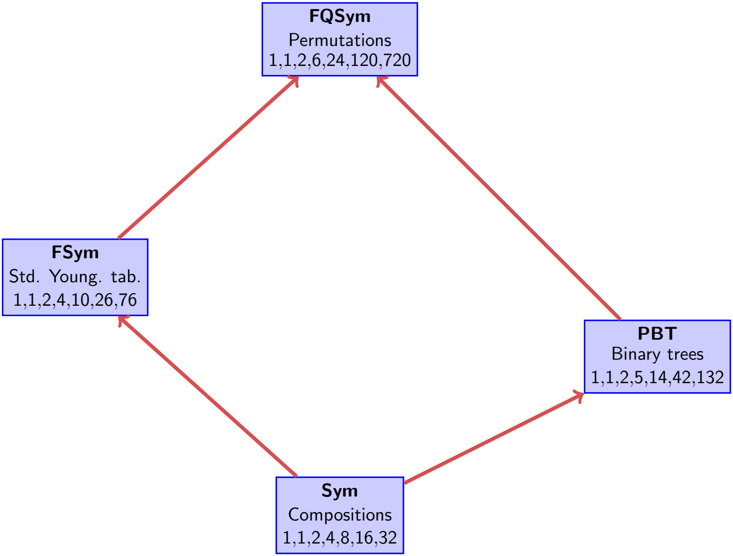
# Some Hopf subalgebras of **FQSym**



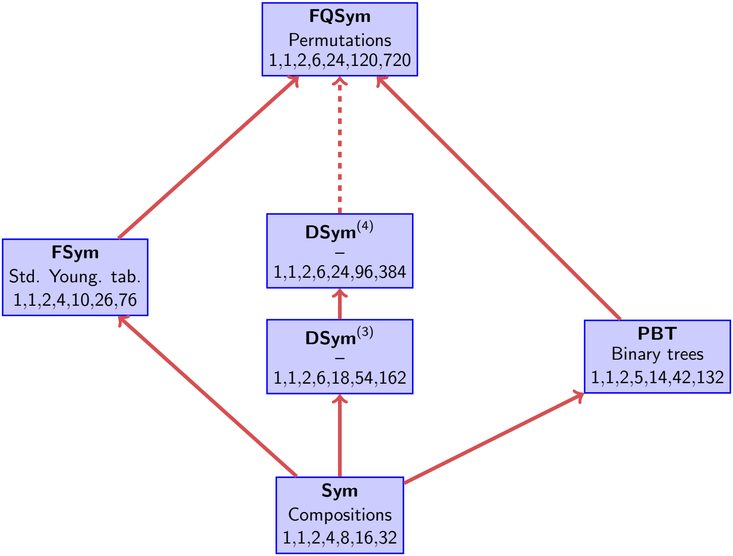
# Some Hopf subalgebras of **FQSym**



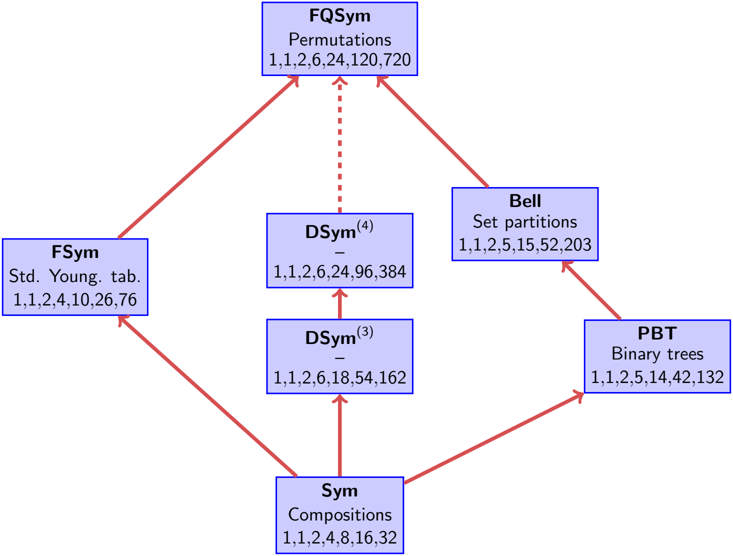
# Some Hopf subalgebras of **FQSym**



# Some Hopf subalgebras of **FQSym**



# Some Hopf subalgebras of **FQSym**



# Combinatorial structures and Hopf subalgebras

The construction of Hopf subalgebras of **FQSym** from an equivalence relation often leads to the construction of new combinatorial structures:

| CHA | Objects | Monoid | Ins. Alg. | Partial order |
|---|---|---|---|---|
| **FQSym** | permutations | $A^*$ | trivial | permutohedron |
| **FSym** | std. Young tab. | plactic | R-S | Reiner order |
| **PBT** | binary trees | sylvester | bst$^{\sim}$ | Tamari lattice |
| **Sym** | compositions | hypoplactic | K-T | hypercube |
| **Bell** | set partitions | Bell | Bell | Bell order |

## Aim of this work

Provide similar structures on Baxter permutations.

# Contents

Algebraic constructions on Baxter permutations

The Baxter combinatorial family

The Baxter monoid

A Robinson-Schensted-like correspondence

The Baxter lattice

The Baxter Hopf algebra

# Baxter permutations

## Definition

The permutation $\sigma$ is a Baxter permutation [**Baxter**, 1964] if it avoids the generalized permutation patterns

$$2-41-3$$

and

$$3-14-2.$$

# Baxter permutations

**Definition**

The permutation $\sigma$ is a Baxter permutation [Baxter, 1964] if it avoids the generalized permutation patterns

$$2 - 41 - 3$$

and

$$3 - 14 - 2.$$

For example:

- 561382479
- $\epsilon$, 1, 1234, 2143

# Baxter permutations

## Definition

The permutation $\sigma$ is a Baxter permutation [Baxter, 1964] if it avoids the generalized permutation patterns

$$2 - 41 - 3$$

and

$$3 - 14 - 2.$$

For example:

- 561382479 is not a Baxter permutation;
- $\epsilon$, 1, 1234, 2143

# Baxter permutations

## Definition

The permutation $\sigma$ is a Baxter permutation [Baxter, 1964] if it avoids the generalized permutation patterns

$$2 - 41 - 3$$

and

$$3 - 14 - 2.$$

For example:

- 561382479 is not a Baxter permutation;
- $\epsilon$, 1, 1234, 2143 are Baxter permutations.

# Baxter objects

**Theorem** [Chung and al., 1978]

The number $b_n$ of Baxter permutations of size $n$ is

$$b_n = \sum_{k=1}^{n} \frac{\binom{n+1}{k-1}\binom{n+1}{k}\binom{n+1}{k+1}}{\binom{n+1}{1}\binom{n+1}{2}}.$$

The sequence $(b_n)_{n\geq 0}$ begins as

$$1, 1, 2, 6, 22, 92, 422, 2074, 10754.$$

This enumerates also

- pairs of twin binary trees [Dulucq, Guibert, 1994];
- rectangular partitions [Yao and al., 2003];
- planar bipolar orientations [Bousquet-Mélou and al., 2010];

and many other objects.

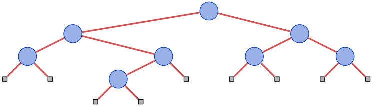
# Viennot's canopy

The canopy of a binary tree $T$ is the word on the alphabet $\{0, 1\}$ obtained by browsing the leaves of $T$ from left to right except the first and the last one, writing 0 if the considered leaf is right-oriented, 1 otherwise.

# Viennot's canopy

The canopy of a binary tree $T$ is the word on the alphabet $\{0, 1\}$ obtained by browsing the leaves of $T$ from left to right except the first and the last one, writing 0 if the considered leaf is right-oriented, 1 otherwise.

For example:

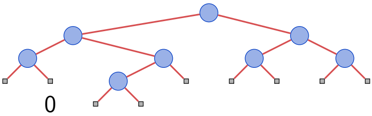


The canopy of this binary tree is

# Viennot's canopy

The canopy of a binary tree $T$ is the word on the alphabet $\{0, 1\}$ obtained by browsing the leaves of $T$ from left to right except the first and the last one, writing 0 if the considered leaf is right-oriented, 1 otherwise.

For example:

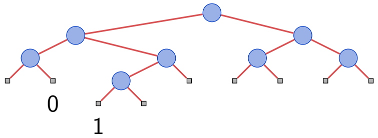



The canopy of this binary tree is 0

# Viennot's canopy

The canopy of a binary tree $T$ is the word on the alphabet $\{0, 1\}$ obtained by browsing the leaves of $T$ from left to right except the first and the last one, writing 0 if the considered leaf is right-oriented, 1 otherwise.

For example:

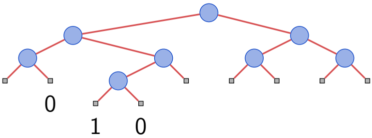



The canopy of this binary tree is 01

# Viennot's canopy

The canopy of a binary tree $T$ is the word on the alphabet $\{0, 1\}$ obtained by browsing the leaves of $T$ from left to right except the first and the last one, writing 0 if the considered leaf is right-oriented, 1 otherwise.

For example:

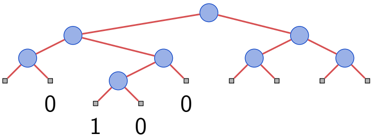



The canopy of this binary tree is 010

# Viennot's canopy

The canopy of a binary tree $T$ is the word on the alphabet $\{0,1\}$ obtained by browsing the leaves of $T$ from left to right except the first and the last one, writing 0 if the considered leaf is right-oriented, 1 otherwise.

For example:

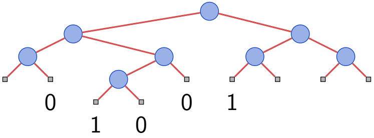



The canopy of this binary tree is 0100

# Viennot's canopy

The canopy of a binary tree $T$ is the word on the alphabet $\{0, 1\}$ obtained by browsing the leaves of $T$ from left to right except the first and the last one, writing 0 if the considered leaf is right-oriented, 1 otherwise.

For example:

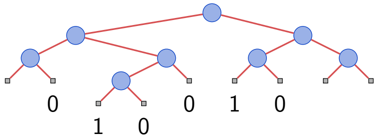



The canopy of this binary tree is 01001

# Viennot's canopy

The canopy of a binary tree $T$ is the word on the alphabet $\{0, 1\}$ obtained by browsing the leaves of $T$ from left to right except the first and the last one, writing 0 if the considered leaf is right-oriented, 1 otherwise.

For example:

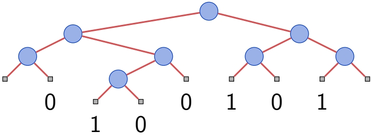



The canopy of this binary tree is 010010

# Viennot's canopy

The canopy of a binary tree $T$ is the word on the alphabet $\{0, 1\}$ obtained by browsing the leaves of $T$ from left to right except the first and the last one, writing 0 if the considered leaf is right-oriented, 1 otherwise.

For example:



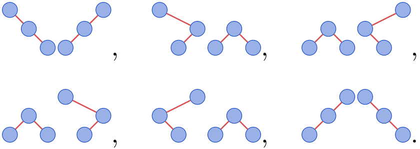

The canopy of this binary tree is 0100101.

# Pairs of twin binary trees

## Definition

A pair $(T_L, T_R)$ of binary trees is a pair of twin binary trees if the canopies of $T_L$ and $T_R$ are complementary.

The six pairs of twin binary trees with 3 nodes are



## Theorem [Dulucq, Guibert, 1994]

The set of pairs of twin binary trees with $n$ nodes is in bijection with the set of Baxter permutations of size $n$.

# Contents


Algebraic constructions on Baxter permutations

The Baxter combinatorial family

The Baxter monoid

A Robinson-Schensted-like correspondence

The Baxter lattice

The Baxter Hopf algebra

# The Baxter monoid

## Definition

The Baxter equivalence relation $\equiv_{\mathrm{B}}$ is defined from the Baxter rewriting rule $\rightleftarrows$ where:

$$\ldots \mathtt{c}\ u\ \mathtt{ad}\ v\ \mathtt{b} \ldots \rightleftarrows \ldots \mathtt{c}\ u\ \mathtt{da}\ v\ \mathtt{b} \ldots \qquad \text{if } \mathtt{a} \leq \mathtt{b} < \mathtt{c} \leq \mathtt{d},$$

$$\ldots \mathtt{b}\ u\ \mathtt{da}\ v\ \mathtt{c} \ldots \rightleftarrows \ldots \mathtt{b}\ u\ \mathtt{ad}\ v\ \mathtt{c} \ldots \qquad \text{if } \mathtt{a} < \mathtt{b} \leq \mathtt{c} < \mathtt{d}.$$

On permutations, one has $\sigma \rightleftarrows \nu$ iff

$$\sigma = u.\mathtt{ad}.v \quad \text{and} \quad \nu = u.\mathtt{da}.v \qquad \text{with} \qquad u_{|[\mathtt{a},\mathtt{d}]} \neq \epsilon \neq v_{|[\mathtt{a},\mathtt{d}]}.$$

We call $A^*/_{\equiv_{\mathrm{B}}}$ the Baxter monoid.

# An example of Baxter equivalence class

The Baxter equivalence class of 3125647 is

3125647

# An example of Baxter equivalence class

The Baxter equivalence class of 3125647 is

3125647

# An example of Baxter equivalence class

The Baxter equivalence class of 3125647 is

3125647

3152647

# An example of Baxter equivalence class

The Baxter equivalence class of 3125647 is

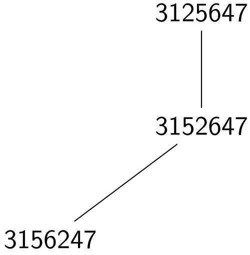
3125647

|

3152647

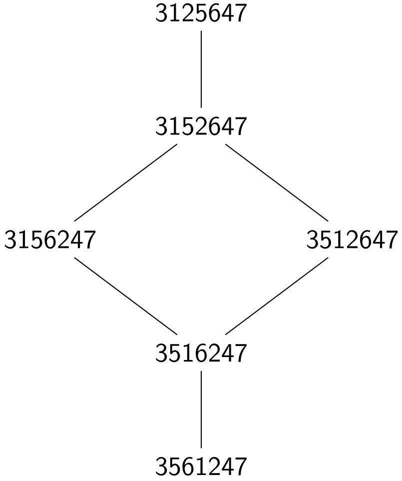
# An example of Baxter equivalence class

The Baxter equivalence class of 3125647 is



# An example of Baxter equivalence class

The Baxter equivalence class of 3125647 is



# Structure properties of the Baxter monoid

### Proposition

The Baxter monoid is compatible with the destandardization process.

### Proposition

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

# Link with the sylvester monoid

The Schützenberger involution $\#\colon \mathfrak{S} \to \mathfrak{S}$ is the composition of two involutions: The mirror image $\sim$ and the complementation $c$.

# Link with the sylvester monoid

The Schützenberger involution $\#\colon \mathfrak{S} \to \mathfrak{S}$ is the composition of two involutions: The mirror image $\sim$ and the complementation $c$.

For example:

- $123 \xrightarrow{\sim}$

# Link with the sylvester monoid

The Schützenberger involution $\#: \mathfrak{S} \to \mathfrak{S}$ is the composition of two involutions: The mirror image $\sim$ and the complementation $c$.

For example:

- $123 \xrightarrow{\sim} 321 \xrightarrow{c}$

# Link with the sylvester monoid

The Schützenberger involution $\#\colon \mathfrak{S} \to \mathfrak{S}$ is the composition of two involutions: The mirror image $\sim$ and the complementation $c$.

For example:

- $123 \xrightarrow{\sim} 321 \xrightarrow{c} 123$,

# Link with the sylvester monoid

The Schützenberger involution $\#\colon \mathfrak{S} \to \mathfrak{S}$ is the composition of two involutions: The mirror image $\sim$ and the complementation $c$.

For example:

- $123 \xrightarrow{\sim} 321 \xrightarrow{c} 123$, and hence $(123)^{\#} = 123$;

# Link with the sylvester monoid

The Schützenberger involution $\#: \mathfrak{S} \to \mathfrak{S}$ is the composition of two involutions: The mirror image $\sim$ and the complementation $c$.

For example:

- $123 \xrightarrow{\sim} 321 \xrightarrow{c} 123$, and hence $(123)^{\#} = 123$;
- $4312 \xrightarrow{\sim}$

# Link with the sylvester monoid

The Schützenberger involution $\#\colon \mathfrak{S} \to \mathfrak{S}$ is the composition of two involutions: The mirror image $\sim$ and the complementation $c$.

For example:

- $123 \xrightarrow{\sim} 321 \xrightarrow{c} 123$, and hence $(123)^{\#} = 123$;
- $4312 \xrightarrow{\sim} 2134 \xrightarrow{c}$

# Link with the sylvester monoid

The Schützenberger involution $\#\colon \mathfrak{S} \to \mathfrak{S}$ is the composition of two involutions: The mirror image $\sim$ and the complementation $c$.

For example:

- $123 \xrightarrow{\sim} 321 \xrightarrow{c} 123$, and hence $(123)^\# = 123$;
- $4312 \xrightarrow{\sim} 2134 \xrightarrow{c} 3421$,

# Link with the sylvester monoid

The Schützenberger involution $\#\colon \mathfrak{S} \to \mathfrak{S}$ is the composition of two involutions: The mirror image $\sim$ and the complementation $c$.

For example:

- $123 \xrightarrow{\sim} 321 \xrightarrow{c} 123$, and hence $(123)^{\#} = 123$;
- $4312 \xrightarrow{\sim} 2134 \xrightarrow{c} 3421$, and hence $(4312)^{\#} = 3421$.

# Link with the sylvester monoid

The Schützenberger involution $\#\colon \mathfrak{S} \to \mathfrak{S}$ is the composition of two involutions: The mirror image $\sim$ and the complementation $c$.

For example:

- $123 \xrightarrow{\sim} 321 \xrightarrow{c} 123$, and hence $(123)^{\#} = 123$;
- $4312 \xrightarrow{\sim} 2134 \xrightarrow{c} 3421$, and hence $(4312)^{\#} = 3421$.

Let $\equiv_{\mathsf{S}}$ be the sylvester equivalence and $\equiv_{\mathsf{S}^{\#}}$ be the $\#$-sylvester equivalence.

## Proposition

Let $\sigma$ and $\nu$ be two permutations. Then,

$$\sigma \equiv_{\mathsf{B}} \nu \qquad \text{iff} \qquad \sigma \equiv_{\mathsf{S}} \nu \quad \text{and} \quad \sigma \equiv_{\mathsf{S}^{\#}} \nu.$$

# Contents


Algebraic constructions on Baxter permutations

The Baxter combinatorial family

The Baxter monoid

A Robinson-Schensted-like correspondence

The Baxter lattice

The Baxter Hopf algebra

# The $\mathbb{P}$-symbol

## Definition

The $\mathbb{P}$-symbol of a permutation $\sigma$ is the pair $(\text{bst}(\sigma), \text{bst}(\sigma^{\sim}))$.

This definition is based on the previous proposition and the following theorem:

## Theorem [Hivert, Novelli, Thibon, 2005]

$$\text{bst}(u^{\sim}) = \text{bst}(v^{\sim}) \qquad \text{iff} \qquad u \equiv_{S} v.$$

Let $\sigma$ be a permutation. The left member of $\mathbb{P}(\sigma)$ encodes the #-sylvester class of $\sigma$ while the second member encodes its sylvester class.

# Insertion algorithm

$\mathbb{P}(\sigma)$ is constructed by iteratively inserting the letters of $\sigma$ and by making well-known leaf insertions and root insertions in binary search trees.

# Insertion algorithm

$\mathbb{P}(\sigma)$ is constructed by iteratively inserting the letters of $\sigma$ and by making well-known leaf insertions and root insertions in binary search trees.

For example, for $\sigma := 6317425$ one has

$\perp \perp$

# Insertion algorithm

$\mathbb{P}(\sigma)$ is constructed by iteratively inserting the letters of $\sigma$ and by making well-known leaf insertions and root insertions in binary search trees.

For example, for $\sigma := 6317425$ one has

$\perp\perp \quad \xrightarrow{6}$ 

# Insertion algorithm

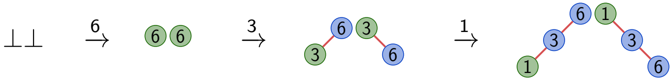
$\mathbb{P}(\sigma)$ is constructed by iteratively inserting the letters of $\sigma$ and by making well-known leaf insertions and root insertions in binary search trees.

For example, for $\sigma := 6317425$ one has

# Insertion algorithm

$\mathbb{P}(\sigma)$ is constructed by iteratively inserting the letters of $\sigma$ and by making well-known leaf insertions and root insertions in binary search trees.

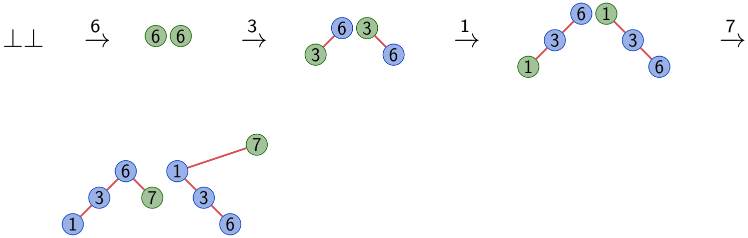
For example, for $\sigma := 6317425$ one has



# Insertion algorithm

$\mathbb{P}(\sigma)$ is constructed by iteratively inserting the letters of $\sigma$ and by making well-known leaf insertions and root insertions in binary search trees.

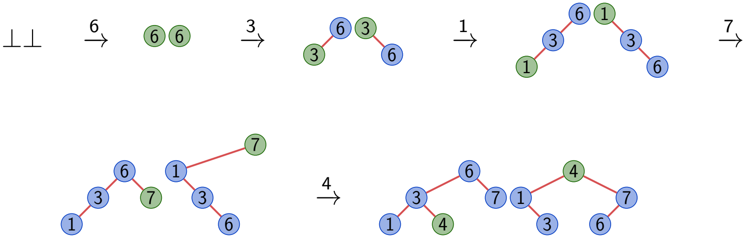
For example, for $\sigma := 6317425$ one has



# Insertion algorithm

$\mathbb{P}(\sigma)$ is constructed by iteratively inserting the letters of $\sigma$ and by making well-known leaf insertions and root insertions in binary search trees.

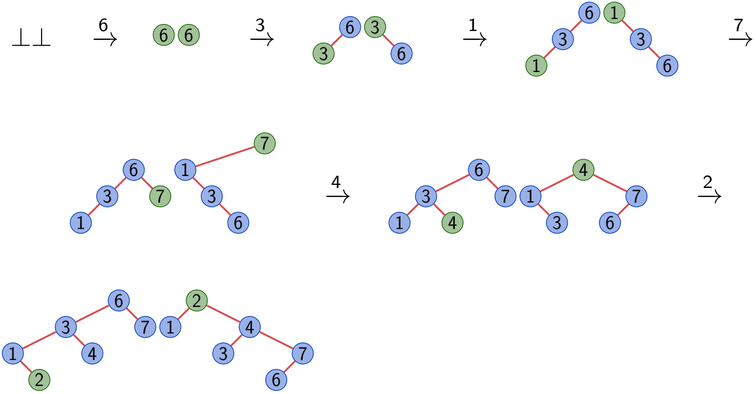
For example, for $\sigma := 6317425$ one has



# Insertion algorithm

$\mathbb{P}(\sigma)$ is constructed by iteratively inserting the letters of $\sigma$ and by making well-known leaf insertions and root insertions in binary search trees.

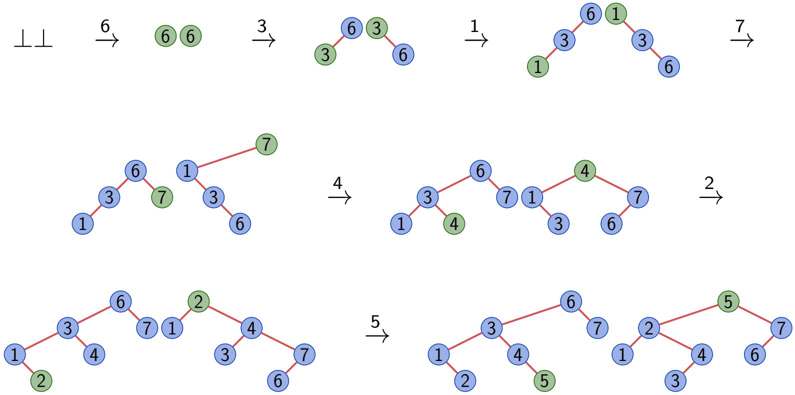
For example, for $\sigma := 6317425$ one has



# Insertion algorithm

$\mathbb{P}(\sigma)$ is constructed by iteratively inserting the letters of $\sigma$ and by making well-known leaf insertions and root insertions in binary search trees.

For example, for $\sigma := 6317425$ one has



# The $\mathbb{P}$-symbol

For all permutation $\sigma$, $\mathbb{P}(\sigma)$ is a pair of twin binary trees.

**Theorem**

$$\mathbb{P}(\sigma) = \mathbb{P}(\nu) \qquad \text{iff} \qquad \sigma \equiv_{\mathsf{B}} \nu.$$

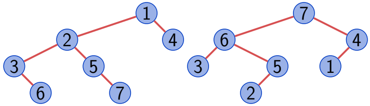
Hence, the application $\mathbb{P} : \mathfrak{S}/_{\equiv_{\mathsf{B}}} \to \mathcal{TBT}$ is an injection.

# The $\mathbb{Q}$-symbol

## Definition

Let $\sigma$ be a permutation and $(T_L, T_R) := \mathbb{P}(\sigma)$. The $\mathbb{Q}$-symbol of $\sigma$ is the pair of twin binary trees $(S_L, S_R)$ where the nodes of $S_L$ (resp. $S_R$) are labeled by the moment of creation of the corresponding node of $T_L$ (resp. $T_R$).

For example, the $\mathbb{Q}$-symbol of $\sigma := 6317425$ is



# A Robinson-Schensted-like correspondence

### Theorem

The map $\sigma \longmapsto (\mathbb{P}(\sigma), \mathbb{Q}(\sigma))$ yields a bijection between $\mathfrak{S}_n$ and the set of pairs $((T_L, T_R), (S_L, S_R))$ where:

1. $(T_L, T_R)$ and $(S_L, S_R)$ are pairs of twin binary trees with same shape;
2. $S_L$ (resp. $S_R$) is an increasing (resp. decreasing) binary trees;
3. $S_L$ and $S_R$ have the same infix reading.

### Theorem

There is a bijection between $\mathfrak{S}_n/_{\equiv_{\mathrm{B}}}$ and the set of pairs of twin binary trees with $n$ nodes.

One can encode equivalence classes of $\mathfrak{S}_n/_{\equiv_{\mathrm{B}}}$ through pairs of twin binary trees.

# Contents

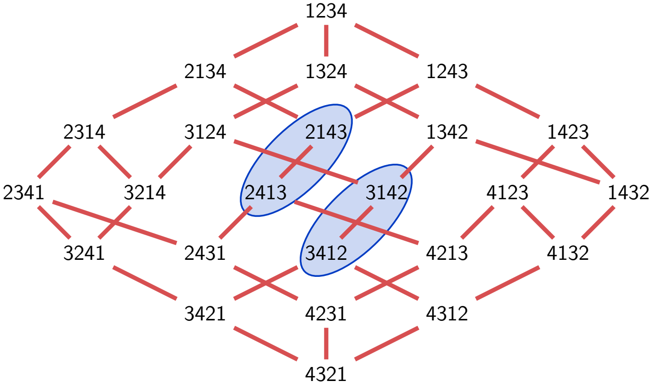
Algebraic constructions on Baxter permutations
The Baxter combinatorial family
The Baxter monoid
A Robinson-Schensted-like correspondence
The Baxter lattice
The Baxter Hopf algebra

# The Baxter lattice

## Proposition

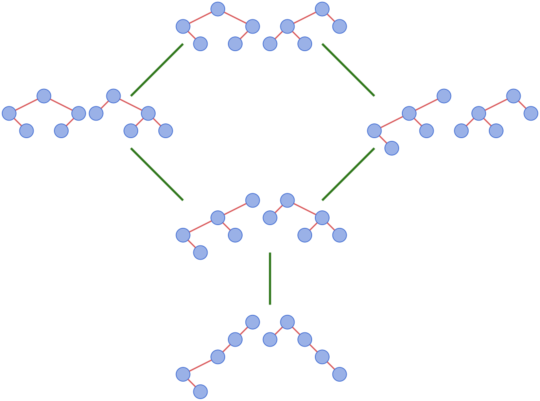
The Baxter equivalence relation is a lattice congruence of the permutohedron.

Here is the Baxter lattice of order 4:



# The Baxter lattice

Covering relations are rotations in binary trees. Here is an interval of the lattice of the pairs of twin binary trees of order 5:



# Contents

Algebraic constructions on Baxter permutations
The Baxter combinatorial family
The Baxter monoid
A Robinson-Schensted-like correspondence
The Baxter lattice
The Baxter Hopf algebra

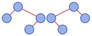
# Construction of **Baxter**

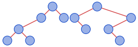
For all pair of twin binary trees $J$, let us define the element $\mathbf{P}_J$ of **FQSym** by:

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

For example:

$\mathbf{P}$  $= \mathbf{F}_{12}$,

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

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

# Construction of **Baxter**

### Theorem

The vector space spanned by the family $\{\mathbf{P}_J\}_{J \in \mathcal{TBT}}$ is a Hopf subalgebra of **FQSym**.

This is the CHA **Baxter**. Its product and its coproduct are well-defined since

- $\equiv_B$ is a congruence,
- $A^*/_{\equiv_B}$ is compatible with the destandardization process,
- $A^*/_{\equiv_B}$ is compatible with the restriction of alphabet intervals.

Moreover, the elements $\mathbf{P}_J$ that appear in a product $\mathbf{P}_{J_0} \cdot \mathbf{P}_{J_1}$ form an interval of the Baxter lattice.

# Multiplicative bases of **Baxter**

Let the following elements of **Baxter**:

$$\mathbf{E}_J := \sum_{J \leq_B J'} \mathbf{P}_{J'} \quad \text{and} \quad \mathbf{H}_J := \sum_{J' \leq_B J} \mathbf{P}_{J'}.$$

By triangularity, the families $\{\mathbf{E}_J\}_{J \in \mathcal{TBT}}$ and $\{\mathbf{H}_J\}_{J \in \mathcal{TBT}}$ are bases of **Baxter**.

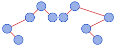
## Proposition

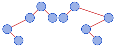
The families $\{\mathbf{E}_J\}_{J \in \mathcal{TBT}}$ and $\{\mathbf{H}_J\}_{J \in \mathcal{TBT}}$ are multiplicative bases of **Baxter**. In particular:

$$\mathbf{E}_{J_0} \cdot \mathbf{E}_{J_1} = \mathbf{E}_{J_0 \diagup J_1} \quad \text{and} \quad \mathbf{H}_{J_0} \cdot \mathbf{H}_{J_1} = \mathbf{H}_{J_0 \diagdown J_1}.$$

# Multiplicative bases of **Baxter**

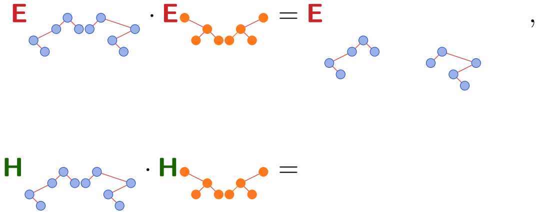
For example:

**E**  · **E**  =

**H**  · **H**  =

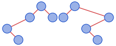
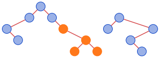
# Multiplicative bases of **Baxter**

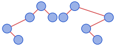
For example:



# Multiplicative bases of **Baxter**

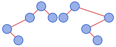
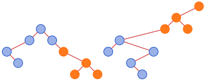
For example:

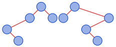
**E**  · **E**  = **E**  ,

**H**  · **H**  =

# Multiplicative bases of **Baxter**

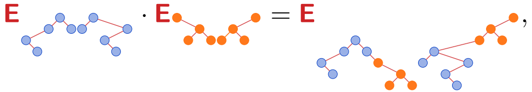
For example:

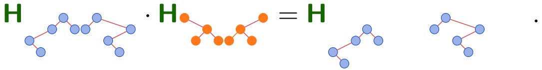
**E**  · **E**  = **E** ,

**H**  · **H**  =

# Multiplicative bases of **Baxter**

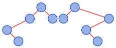
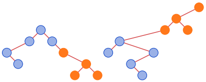
For example:

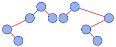
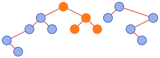


# Multiplicative bases of **Baxter**

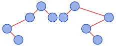
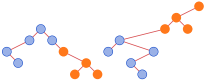
For example:

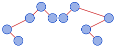
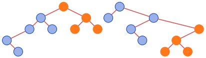
**E**  · **E**  = **E** ,

**H**  · **H**  = **H** .

# Multiplicative bases of **Baxter**

For example:

**E**  · **E**  = **E**  ,

**H**  · **H**  = **H**  .

# Algebraic structure of **Baxter**

Multiplicative bases of **Baxter** and freeness of **FQSym** imply

### Proposition

**Baxter** is free as an algebra.

The results of Foissy [**Foissy**, 2005] on the bidendriform structure of **FQSym** imply

### Proposition

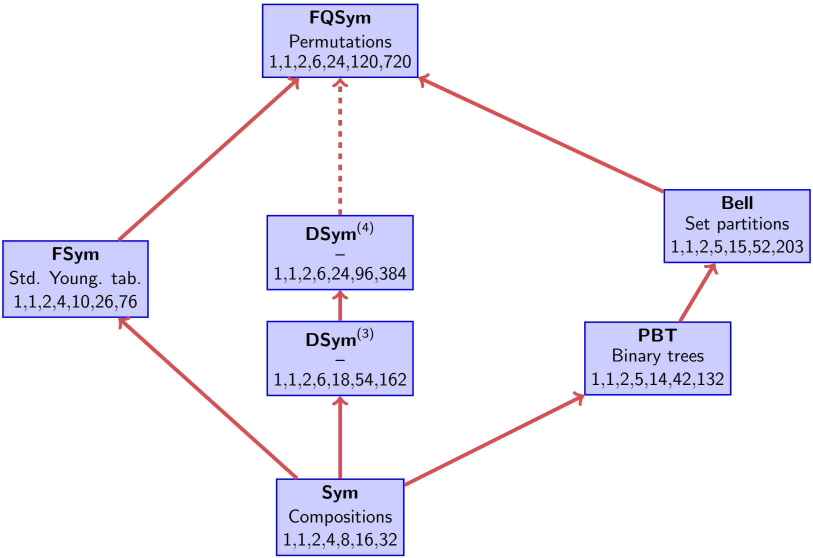
The primitive Lie algebra of **Baxter** is free.

### Proposition

**Baxter** is self-dual.

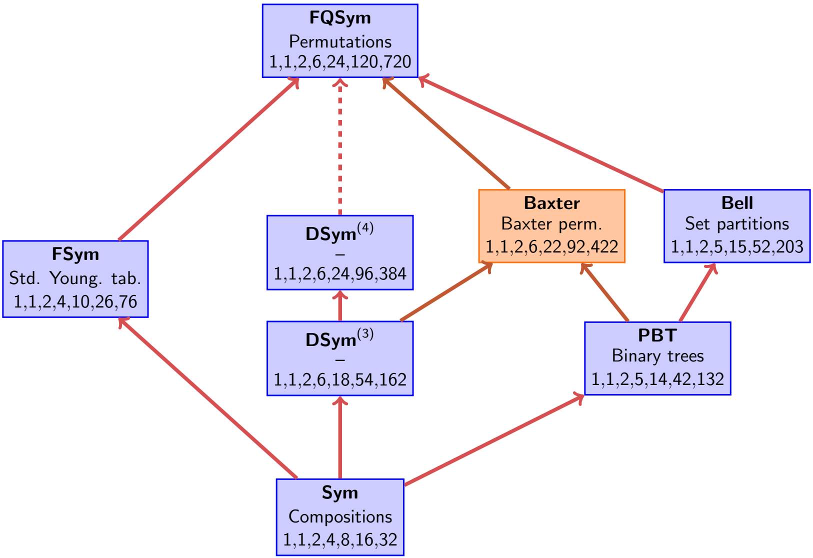
Nevertheless, no isomorphism between **Baxter** and **Baxter**$^\star$ is known.

# Conclusion



**FQSym**
Permutations
1,1,2,6,24,120,720
**FSym**
Std. Young. tab.
1,1,2,4,10,26,76
**DSym**$^{(4)}$
–
1,1,2,6,24,96,384
**DSym**$^{(3)}$
–
1,1,2,6,18,54,162
**Bell**
Set partitions
1,1,2,5,15,52,203
**PBT**
Binary trees
1,1,2,5,14,42,132
**Sym**
Compositions
1,1,2,4,8,16,32

# Conclusion


FQSym
Permutations
1,1,2,6,24,120,720
DSym$^{(4)}$
–
1,1,2,6,24,96,384
Baxter
Baxter perm.
1,1,2,6,22,92,422
Bell
Set partitions
1,1,2,5,15,52,203
FSym
Std. Young. tab.
1,1,2,4,10,26,76
DSym$^{(3)}$
–
1,1,2,6,18,54,162
PBT
Binary trees
1,1,2,5,14,42,132
Sym
Compositions
1,1,2,4,8,16,32

# Conclusion

| CHA | Objects | Monoid | Ins. Alg. | Partial order |
|---|---|---|---|---|
| **FQSym** | permutations | $A^*$ | trivial | permutohedron |
| **FSym** | std. Young tab. | plactic | R-S | Reiner order |
| **PBT** | binary trees | sylvester | bst$^\sim$ | Tamari lattice |
| **Sym** | compositions | hypoplactic | K-T | hypercube |
| **Bell** | set partitions | Bell | Bell | Bell order |

# Conclusion

| CHA | Objects | Monoid | Ins. Alg. | Partial order |
|---|---|---|---|---|
| **FQSym** | permutations | $A^*$ | trivial | permutohedron |
| **FSym** | std. Young tab. | plactic | R-S | Reiner order |
| **PBT** | binary trees | sylvester | $\text{bst}^{\sim}$ | Tamari lattice |
| **Sym** | compositions | hypoplactic | K-T | hypercube |
| **Bell** | set partitions | Bell | Bell | Bell order |
| **Baxter** | pairs of twin b. t. | Baxter | $\text{bst} + \text{bst}^{\sim}$ | Baxter lattice |