Crystals & Trees

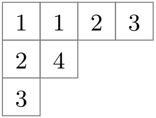
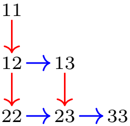
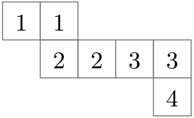
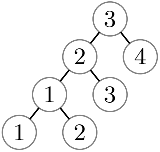
António Malheiro
(joint work with Alan Cain)

CMA/FCT
Universidade Nova de Lisboa

November 30, 2020

Acknowledgement:
This work was supported by CMA within the projects
UIDB/00297/2020
PTDC/MAT-PUR/31174/2017
financed by 'Fundação para a Ciência e a Tecnologia'

Some ‘plactic-like’ monoids

<table>
<tr><td></td><td></td><td></td><td>Crystals</td></tr>
<tr>
<td>Symmetric functions</td>
<td>Young tableaux
</td>
<td>Plactic monoid</td>
<td></td>
</tr>
<tr>
<td>Quasi-symmetric functions</td>
<td>Quasi-ribbon tableaux
</td>
<td>Hypoplactic monoid</td>
<td>?</td>
</tr>
<tr>
<td>Loday–Ronco Hopf algebra</td>
<td>Binary search trees
</td>
<td>Sylvester monoid</td>
<td>?</td>
</tr>
</table>

Plactic

1	1	2	4
2	3	6	
5			

Young tableau:

- Rows non-decreasing left to right.
- Columns increasing top to bottom.
- Longer columns to the left.

1	3	5	7
2	4	8	
6			

Standard Young tableau:

- contains each symbol in $\{1, \ldots, n\}$ (for some n) exactly once.

Plactic

$$\begin{array}{|c|c|c|c|}\hline 1 & 1 & 2 & 4 \\ \hline 2 & 3 & 6 \\ \cline{1-3} 5 \\ \cline{1-1}\end{array} \quad \leftarrow 3$$

Schensted's algorithm inserts a symbol $a \in \mathbb{N}$ into a tableau T:

1. If appending a to the end of the top row gives a tableau, this is the result.
2. Otherwise, let b the leftmost symbol of the top row such that $b > a$. Replace b with a ('bumping b').
3. Recursively insert b into the tableau formed by all rows below the topmost.

Result is denoted $T \leftarrow a$.

Plactic

$$\begin{array}{|c|c|c|c|}\hline 1 & 1 & 2 & 3 \\ \hline 2 & 3 & 6 & \multicolumn{1}{c}{} \\ \cline{1-3} 5 & \multicolumn{3}{c}{} \\ \cline{1-1}\end{array} \leftarrow 4$$

Schensted's algorithm inserts a symbol $a \in \mathbb{N}$ into a tableau T:

1. If appending a to the end of the top row gives a tableau, this is the result.
2. Otherwise, let b the leftmost symbol of the top row such that $b > a$. Replace b with a ('bumping b').
3. Recursively insert b into the tableau formed by all rows below the topmost.

Result is denoted $T \leftarrow a$.

Plactic

$$\begin{array}{|c|c|c|c|}\hline 1 & 1 & 2 & 3 \\ \hline 2 & 3 & 4 \\ \cline{1-3} 5 & \multicolumn{3}{l}{\leftarrow 6} \\ \cline{1-1} \end{array}$$

Schensted's algorithm inserts a symbol $a \in \mathbb{N}$ into a tableau T:

1. If appending a to the end of the top row gives a tableau, this is the result.
2. Otherwise, let b the leftmost symbol of the top row such that $b > a$. Replace b with a ('bumping b').
3. Recursively insert b into the tableau formed by all rows below the topmost.

Result is denoted $T \leftarrow a$.

Plactic

1	1	2	3
2	3	4	
5	6		

Schensted's algorithm inserts a symbol $a \in \mathbb{N}$ into a tableau T:

1. If appending a to the end of the top row gives a tableau, this is the result.
2. Otherwise, let b the leftmost symbol of the top row such that $b > a$. Replace b with a ('bumping b').
3. Recursively insert b into the tableau formed by all rows below the topmost.

Result is denoted $T \leftarrow a$.

Plactic

1	1	2	3
2	3	4	
5	6		

Schensted's algorithm inserts a symbol $a \in \mathbb{N}$ into a tableau T:

1. If appending a to the end of the top row gives a tableau, this is the result.
2. Otherwise, let b the leftmost symbol of the top row such that $b > a$. Replace b with a ('bumping b').
3. Recursively insert b into the tableau formed by all rows below the topmost.

Result is denoted $T \leftarrow a$.

- This process can be reversed if we know how to begin.

Plactic

1	1	2	3
2	3	4	
5	6		

Schensted's algorithm inserts a symbol $a \in \mathbb{N}$ into a tableau T:

1. If appending a to the end of the top row gives a tableau, this is the result.
2. Otherwise, let b the leftmost symbol of the top row such that $b > a$. Replace b with a ('bumping b').
3. Recursively insert b into the tableau formed by all rows below the topmost.

Result is denoted $T \leftarrow a$.

- This process can be reversed if we know how to begin.

Plactic

1	1	2	3
2	3	4	
5	$\leftarrow 6$		

Schensted's algorithm inserts a symbol $a \in \mathbb{N}$ into a tableau T:

1. If appending a to the end of the top row gives a tableau, this is the result.
2. Otherwise, let b the leftmost symbol of the top row such that $b > a$. Replace b with a ('bumping b').
3. Recursively insert b into the tableau formed by all rows below the topmost.

Result is denoted $T \leftarrow a$.

- This process can be reversed if we know how to begin.

Plactic

$$\begin{array}{|c|c|c|c|}\hline 1 & 1 & 2 & 3 \\ \hline 2 & 3 & 6 \\ \cline{1-3} 5 \\ \cline{1-1}\end{array} \leftarrow 4$$

Schensted's algorithm inserts a symbol $a \in \mathbb{N}$ into a tableau T:

1. If appending a to the end of the top row gives a tableau, this is the result.
2. Otherwise, let b the leftmost symbol of the top row such that $b > a$. Replace b with a ('bumping b').
3. Recursively insert b into the tableau formed by all rows below the topmost.

Result is denoted $T \leftarrow a$.

- This process can be reversed if we know how to begin.

Plactic

$$\begin{array}{|c|c|c|c|}\hline 1 & 1 & 2 & 4 \\ \hline 2 & 3 & 6 \\ \cline{1-3} 5 \\ \cline{1-1}\end{array} \leftarrow 3$$

Schensted's algorithm inserts a symbol $a \in \mathbb{N}$ into a tableau T:

1. If appending a to the end of the top row gives a tableau, this is the result.
2. Otherwise, let b the leftmost symbol of the top row such that $b > a$. Replace b with a ('bumping b').
3. Recursively insert b into the tableau formed by all rows below the topmost.

Result is denoted $T \leftarrow a$.

- This process can be reversed if we know how to begin.

Plactic

$$\begin{array}{|c|c|c|c|}
\hline
1 & 1 & 2 & 4 \\
\hline
2 & 3 & 6 \\
\cline{1-3}
5 \\
\cline{1-1}
\end{array}$$

$\mathrm{P}_{\mathrm{plac}}(21513264)$

Let $\mathcal{A}_n = \{1 < 2 < \ldots < n\}$.
From a word $w = w_1 \cdots w_n \in \mathcal{A}_n^*$:

- compute a tableau $\mathrm{P}_{\mathrm{plac}}(w)$ by inserting $w_1, \ldots, w_n$.

Plactic

1	1	2	4
2	3	6	
5			

$\mathrm{P}_{\mathrm{plac}}(21513264)$

1	3	5	7
2	4	8	
6			

$\mathrm{Q}_{\mathrm{plac}}(21513264)$

Let $\mathcal{A}_n = \{1 < 2 < \ldots < n\}$.
From a word $w = w_1 \cdots w_n \in \mathcal{A}_n^*$:

- compute a tableau $\mathrm{P}_{\mathrm{plac}}(w)$ by inserting $w_1, \ldots, w_n$.
- compute a standard tableau $\mathrm{Q}_{\mathrm{plac}}(w)$ (the recording tableau) by labelling in order the new squares.

Plactic

1	1	2	4
2	3	6	
5			

$\leftarrow 3$

$\mathrm{P}_{\mathrm{plac}}(21513264) \leftarrow 3$

1	3	5	7
2	4	8	
6			

$\mathrm{Q}_{\mathrm{plac}}(21513264)$

Let $\mathcal{A}_n = \{1 < 2 < \ldots < n\}$.
From a word $w = w_1 \cdots w_n \in \mathcal{A}_n^*$:

- compute a tableau $\mathrm{P}_{\mathrm{plac}}(w)$ by inserting $w_1, \ldots, w_n$.
- compute a standard tableau $\mathrm{Q}_{\mathrm{plac}}(w)$ (the recording tableau) by labelling in order the new squares.

Plactic

1	1	2	3
2	3	4	
5	6		

$\mathrm{P}_{\mathrm{plac}}(215132643)$

1	3	5	7
2	4	8	
6	9		

$\mathrm{Q}_{\mathrm{plac}}(215132643)$

Let $\mathcal{A}_n = \{1 < 2 < \ldots < n\}$.
From a word $w = w_1 \cdots w_n \in \mathcal{A}_n^*$:

- compute a tableau $\mathrm{P}_{\mathrm{plac}}(w)$ by inserting $w_1, \ldots, w_n$.
- compute a standard tableau $\mathrm{Q}_{\mathrm{plac}}(w)$ (the recording tableau) by labelling in order the new squares.

Plactic

1	1	2	3
2	3	4	
5	6		

Let $\mathcal{A}_n = \{1 < 2 < \ldots < n\}$.
From a word $w = w_1 \cdots w_n \in \mathcal{A}_n^*$:

- compute a tableau $\mathrm{P}_{\mathrm{plac}}(w)$ by inserting $w_1, \ldots, w_n$.

1	3	5	7
2	4	8	
6	9		

- compute a standard tableau $\mathrm{Q}_{\mathrm{plac}}(w)$ (the recording tableau) by labelling in order the new squares.

Plactic

Let $\mathcal{A}_n = \{1 < 2 < \ldots < n\}$.
From a word $w = w_1 \cdots w_n \in \mathcal{A}_n^*$:

- compute a tableau $\mathrm{P}_{\mathrm{plac}}(w)$ by inserting $w_1, \ldots, w_n$.

$$\begin{array}{|c|c|c|c|}\hline 1 & 1 & 2 & 3 \\ \hline 2 & 3 & 4 \\ \cline{1-3} 5 & 6 \\ \cline{1-2} \end{array}$$

- compute a standard tableau $\mathrm{Q}_{\mathrm{plac}}(w)$ (the recording tableau) by labelling in order the new squares.

$$\begin{array}{|c|c|c|c|}\hline 1 & 3 & 5 & 7 \\ \hline 2 & 4 & 8 \\ \cline{1-3} 6 & 9 \\ \cline{1-2} \end{array}$$

$$w \overset{1:1}{\longleftrightarrow} \left(\mathrm{P}_{\mathrm{plac}}(w), \mathrm{Q}_{\mathrm{plac}}(w)\right)$$

Robinson–Schensted correspondence

Plactic

1	1	2	3
2	3	4	
5	6		

$P_{\mathrm{plac}}(521\,631\,42\,3)$

Column reading of a tableau:

- From leftmost column to rightmost.
- In each column, from bottom to top.

Result is a tableau word.

Plactic

1	1	2	3
2	3	4	
5	6		

$\mathrm{P}_{\mathrm{plac}}(521\,631\,42\,3)$

1	4	7	9
2	5	8	
3	6		

$\mathrm{Q}_{\mathrm{plac}}(521\,631\,42\,3)$

Column reading of a tableau:

- From leftmost column to rightmost.
- In each column, from bottom to top.

Result is a tableau word.

Recording tableau where columns are filled from top to bottom with consecutive numbers.

Plactic

1	1	2	3
2	3	4	
5	6		

$\mathrm{P}_{\mathrm{plac}}(521\,631\,42\,3)$

1	4	7	9
2	5	8	
3	6		

$\mathrm{Q}_{\mathrm{plac}}(521\,631\,42\,3)$

Column reading of a tableau:

- From leftmost column to rightmost.
- In each column, from bottom to top.

Result is a tableau word.

Recording tableau where columns are filled from top to bottom with consecutive numbers.

By holding P and varying Q over all standard tableau of the same shape, we get all words w such that $\mathrm{P}_{\mathrm{plac}}(w) = \mathrm{P}$. (Hook length formula)

Plactic monoid

Theorem (Knuth 1970)

The relation $\equiv_{\mathrm{plac}}$ defined by $u \equiv_{\mathrm{plac}} v \iff \mathrm{P}_{\mathrm{plac}}(u) = \mathrm{P}_{\mathrm{plac}}(v)$ is a congruence on $\mathcal{A}_n^*$.

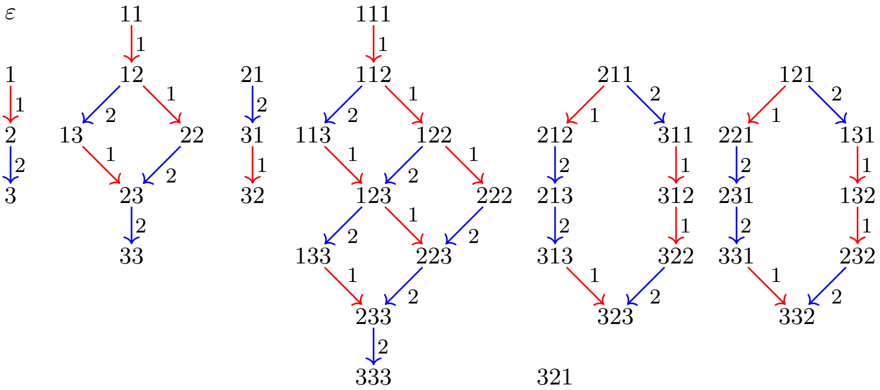
The factor monoid $\mathrm{plac}_n = \mathcal{A}_n^* / {\equiv_{\mathrm{plac}}}$ is the Plactic monoid of rank n.

plac_n is presented by $\langle\, \mathcal{A}_n \,|\, \mathcal{R}_{\mathrm{plac}} \,\rangle$, where

$$\begin{aligned}\mathcal{R}_{\mathrm{plac}} = {} & \{\, (acb, cab) : a \leq b < c \,\} \\ & \cup \{\, (bac, bca) : a < b \leq c \,\}.\end{aligned}$$

- Tableaux form a cross-section.
- Indexes representations of the special linear Lie algebra $\mathfrak{sl}_{n+1}$.
- Analogous plactic monoids corresponding to other Lie algebras $\mathfrak{sp}_n$, $\mathfrak{so}_{2n+1}$, $\mathfrak{so}_{2n}$, and the exceptional Lie algebra G_2.
- Each of these monoids has a notion of tableaux and an insertion algorithm.

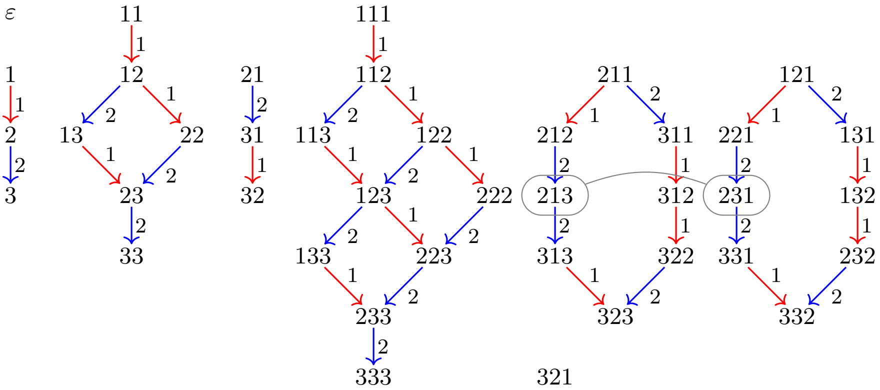
Crystal graph for plac_n



In purely combinatorial, monoid-theoretic terms:

- A crystal graph is a labelled directed graph, with vertices being words in the free monoid $\mathcal{A}_n^*$.
- Isomorphisms between connected components correspond to the congruence $\equiv_{\mathrm{plac}}$ on $\mathcal{A}_n^*$.
- ‘Isomorphism’ means ‘weight-preserving labelled digraph isomorphism’

Crystal graph for plac_n



In purely combinatorial, monoid-theoretic terms:

- A crystal graph is a labelled directed graph, with vertices being words in the free monoid $\mathcal{A}_n^*$.
- Isomorphisms between connected components correspond to the congruence $\equiv_{\mathrm{plac}}$ on $\mathcal{A}_n^*$.
- ‘Isomorphism’ means ‘weight-preserving labelled digraph isomorphism’

Building the crystal graph

The crystal graph for plac_n is defined by:

- vertex set $\mathcal{A}_n^*$
- edges $w \xrightarrow{i} \tilde{f}_i(w)$ and $\tilde{e}_i(w) \xrightarrow{i} w$, whenever defined.

Building the crystal graph

The crystal graph for plac_n is defined by:

- vertex set $\mathcal{A}_n^*$
- edges $w \xrightarrow{i} \tilde{f}_i(w)$ and $\tilde{e}_i(w) \xrightarrow{i} w$, whenever defined.

The operators $\tilde{e}_i$ and $\tilde{f}_i$ are mutually inverse when defined:

- if $\tilde{e}_i(w)$ is defined, then $w = \tilde{f}_i\tilde{e}_i(w)$;
- if $\tilde{f}_i(w)$ is defined, then $w = \tilde{e}_i\tilde{f}_i(w)$.

Building the crystal graph

The crystal graph for plac_n is defined by:

- vertex set $\mathcal{A}_n^*$
- edges $w \xrightarrow{i} \tilde{f}_i(w)$ and $\tilde{e}_i(w) \xrightarrow{i} w$, whenever defined.

The operators $\tilde{e}_i$ and $\tilde{f}_i$ are mutually inverse when defined:

- if $\tilde{e}_i(w)$ is defined, then $w = \tilde{f}_i\tilde{e}_i(w)$;
- if $\tilde{f}_i(w)$ is defined, then $w = \tilde{e}_i\tilde{f}_i(w)$.

The definition of the (partial) Kashiwara operators $\tilde{e}_i$ and $\tilde{f}_i$ starts from the crystal basis for plac_n:

$$1 \xrightarrow{1} 2 \xrightarrow{2} 3 \xrightarrow{3} \ldots \xrightarrow{n-2} n-1 \xrightarrow{n-1} n$$

Building the crystal graph

The crystal graph for plac_n is defined by:

- vertex set $\mathcal{A}_n^*$
- edges $w \xrightarrow{i} \tilde{f}_i(w)$ and $\tilde{e}_i(w) \xrightarrow{i} w$, whenever defined.

The operators $\tilde{e}_i$ and $\tilde{f}_i$ are mutually inverse when defined:

- if $\tilde{e}_i(w)$ is defined, then $w = \tilde{f}_i\tilde{e}_i(w)$;
- if $\tilde{f}_i(w)$ is defined, then $w = \tilde{e}_i\tilde{f}_i(w)$.

The definition of the (partial) Kashiwara operators $\tilde{e}_i$ and $\tilde{f}_i$ starts from the crystal basis for plac_n:

$$1 \xrightarrow{1} 2 \xrightarrow{2} 3 \xrightarrow{3} \dots \xrightarrow{n-2} n-1 \xrightarrow{n-1} n$$

Define on $\mathcal{A}_n^* \setminus \mathcal{A}_n$ by using the bracketing rule: compute $\tilde{e}_2(w)$ and $\tilde{f}_2(w)$;

$$1\ 2\ 2\ 3\ 1\ 2\ 3\ 3\ 1\ 1\ 2\ 2\ 3$$

Building the crystal graph

The crystal graph for plac_n is defined by:

- vertex set $\mathcal{A}_n^*$
- edges $w \xrightarrow{i} \tilde{f}_i(w)$ and $\tilde{e}_i(w) \xrightarrow{i} w$, whenever defined.

The operators $\tilde{e}_i$ and $\tilde{f}_i$ are mutually inverse when defined:

- if $\tilde{e}_i(w)$ is defined, then $w = \tilde{f}_i\tilde{e}_i(w)$;
- if $\tilde{f}_i(w)$ is defined, then $w = \tilde{e}_i\tilde{f}_i(w)$.

The definition of the (partial) Kashiwara operators $\tilde{e}_i$ and $\tilde{f}_i$ starts from the crystal basis for plac_n:

$$1 \xrightarrow{1} 2 \xrightarrow{2} 3 \xrightarrow{3} \dots \xrightarrow{n-2} n-1 \xrightarrow{n-1} n$$

Define on $\mathcal{A}_n^* \setminus \mathcal{A}_n$ by using the bracketing rule: compute $\tilde{e}_2(w)$ and $\tilde{f}_2(w)$;

$$\begin{array}{ccccccccccccc} 1 & 2 & 2 & 3 & 1 & 2 & 3 & 3 & 1 & 1 & 2 & 2 & 3 \\ & + & + & - & & + & - & - & & & + & + & - \end{array}$$

Building the crystal graph

The crystal graph for plac_n is defined by:

- vertex set $\mathcal{A}_n^*$
- edges $w \xrightarrow{i} \tilde{f}_i(w)$ and $\tilde{e}_i(w) \xrightarrow{i} w$, whenever defined.

The operators $\tilde{e}_i$ and $\tilde{f}_i$ are mutually inverse when defined:

- if $\tilde{e}_i(w)$ is defined, then $w = \tilde{f}_i\tilde{e}_i(w)$;
- if $\tilde{f}_i(w)$ is defined, then $w = \tilde{e}_i\tilde{f}_i(w)$.

The definition of the (partial) Kashiwara operators $\tilde{e}_i$ and $\tilde{f}_i$ starts from the crystal basis for plac_n:

$$1 \xrightarrow{1} 2 \xrightarrow{2} 3 \xrightarrow{3} \ldots \xrightarrow{n-2} n-1 \xrightarrow{n-1} n$$

Define on $\mathcal{A}_n^* \setminus \mathcal{A}_n$ by using the bracketing rule: compute $\tilde{e}_2(w)$ and $\tilde{f}_2(w)$;

$$\begin{array}{ccccccccccccc} 1 & 2 & 2 & 3 & 1 & 2 & 3 & 3 & 1 & 1 & 2 & 2 & 3 \\ & + & + & & & & - & & & & & + & - \end{array}$$

Building the crystal graph

The crystal graph for plac_n is defined by:

- vertex set $\mathcal{A}_n^*$
- edges $w \xrightarrow{i} \tilde{f}_i(w)$ and $\tilde{e}_i(w) \xrightarrow{i} w$, whenever defined.

The operators $\tilde{e}_i$ and $\tilde{f}_i$ are mutually inverse when defined:

- if $\tilde{e}_i(w)$ is defined, then $w = \tilde{f}_i \tilde{e}_i(w)$;
- if $\tilde{f}_i(w)$ is defined, then $w = \tilde{e}_i \tilde{f}_i(w)$.

The definition of the (partial) Kashiwara operators $\tilde{e}_i$ and $\tilde{f}_i$ starts from the crystal basis for plac_n:

$$1 \xrightarrow{1} 2 \xrightarrow{2} 3 \xrightarrow{3} \ldots \xrightarrow{n-2} n-1 \xrightarrow{n-1} n$$

Define on $\mathcal{A}_n^* \setminus \mathcal{A}_n$ by using the bracketing rule: compute $\tilde{e}_2(w)$ and $\tilde{f}_2(w)$;

$$\begin{array}{ccccccccccccc} 1 & 2 & 2 & 3 & 1 & 2 & 3 & 3 & 1 & 1 & 2 & 2 & 3 \\ & + & + & & & & & & & & & & - \end{array}$$

Building the crystal graph

The crystal graph for plac_n is defined by:

- vertex set $\mathcal{A}_n^*$
- edges $w \xrightarrow{i} \tilde{f}_i(w)$ and $\tilde{e}_i(w) \xrightarrow{i} w$, whenever defined.

The operators $\tilde{e}_i$ and $\tilde{f}_i$ are mutually inverse when defined:

- if $\tilde{e}_i(w)$ is defined, then $w = \tilde{f}_i\tilde{e}_i(w)$;
- if $\tilde{f}_i(w)$ is defined, then $w = \tilde{e}_i\tilde{f}_i(w)$.

The definition of the (partial) Kashiwara operators $\tilde{e}_i$ and $\tilde{f}_i$ starts from the crystal basis for plac_n:

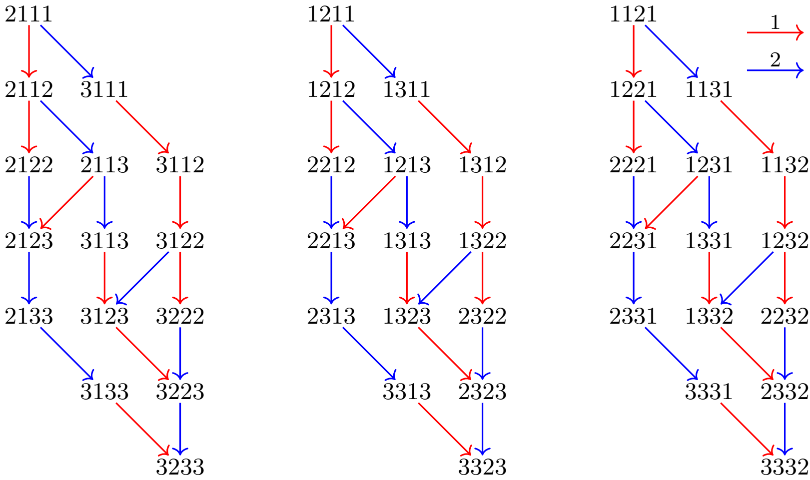
$$1 \xrightarrow{1} 2 \xrightarrow{2} 3 \xrightarrow{3} \ldots \xrightarrow{n-2} n-1 \xrightarrow{n-1} n$$

Define on $\mathcal{A}_n^* \setminus \mathcal{A}_n$ by using the bracketing rule: compute $\tilde{e}_2(w)$ and $\tilde{f}_2(w)$;

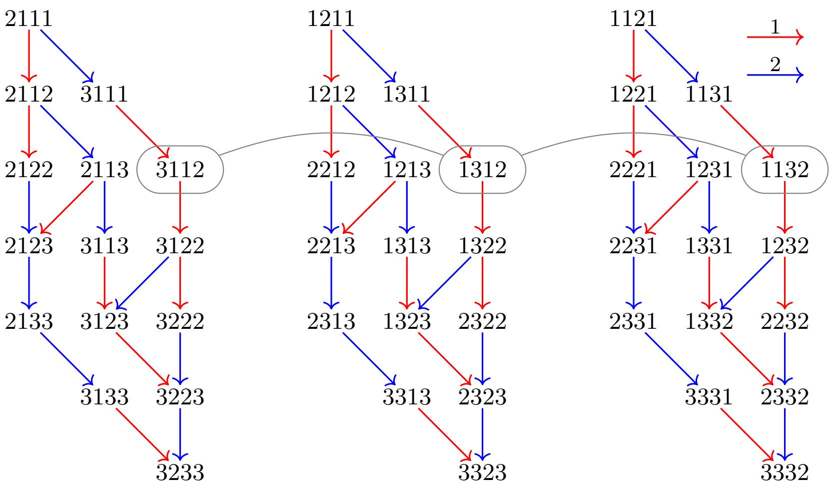
$$\begin{array}{ccccccccccccc} 1 & 2 & 2 & 3 & 1 & 2 & 3 & 3 & 1 & 1 & 2 & 2 & 3 \\ & + & + & & & & & & & & & & - \end{array}$$

$$\tilde{e}_2(w) = 1223123311222 \qquad \tilde{f}_2(w) = 1233123311223$$

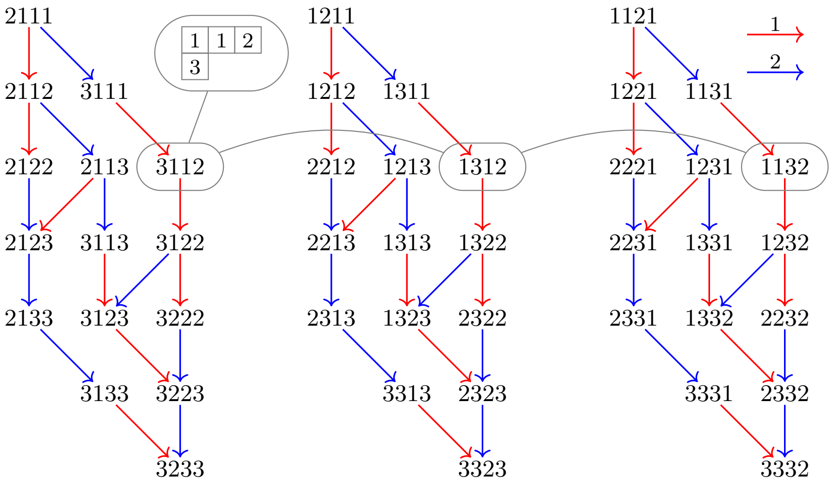
Some isomorphic components of the graph for plac_3



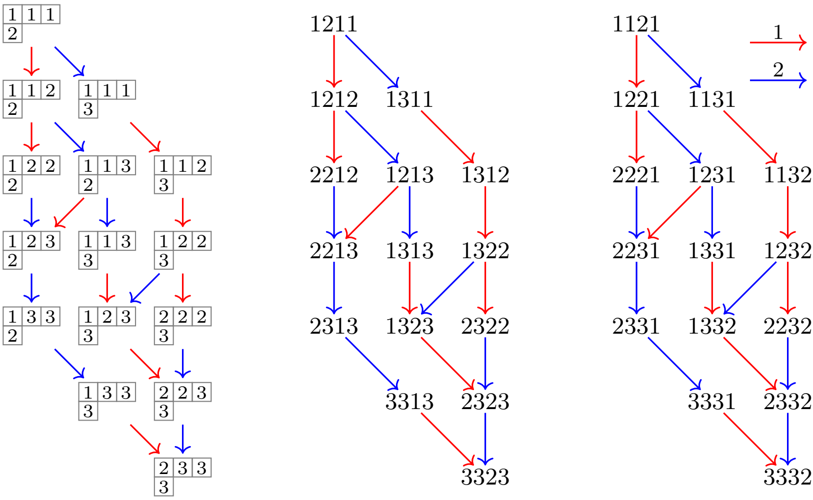
Some isomorphic components of the graph for plac_3



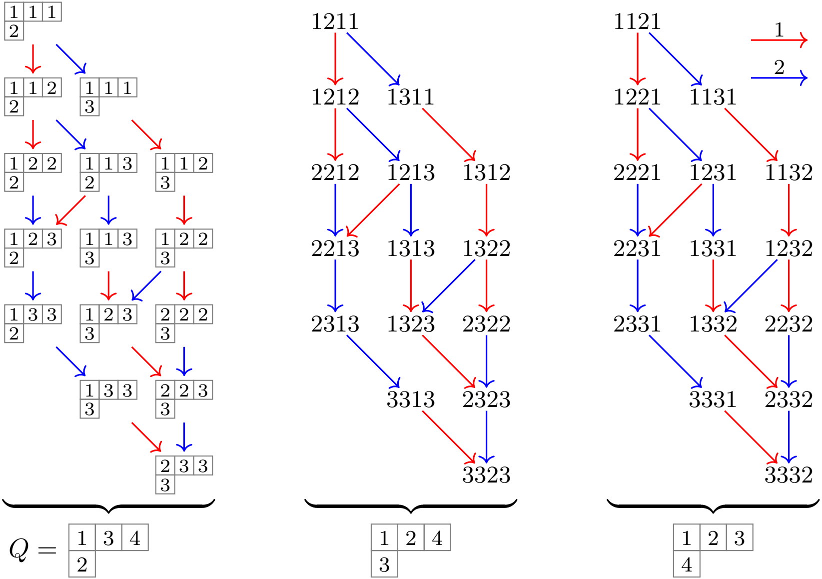
Some isomorphic components of the graph for plac_3



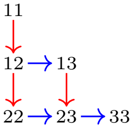
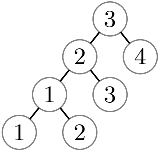
Some isomorphic components of the graph for plac_3



Some isomorphic components of the graph for plac_3



Some ‘plactic-like’ monoids

Symmetric functions	Young tableaux $\begin{array}{cccc} 1 & 1 & 2 & 3 \\ 2 & 4 & & \\ 3 & & & \end{array}$	Plactic monoid	Crystals 
Quasi-symmetric functions	Quasi-ribbon tableaux $\begin{array}{ccccc} 1 & 1 & & & \\ & 2 & 2 & 3 & 3 \\ & & & & 4 \end{array}$	Hypoplactic monoid	?
Loday–Ronco Hopf algebra	Binary search trees 	Sylvester monoid	?

Properties of crystals

For the plactic monoid:

- Kashiwara operators $\tilde{e}_i$ and $\tilde{f}_i$ raise and lower weights.
- $\tilde{e}_i$ and $\tilde{f}_i$ preserve shapes of tableaux.
- Every connected component contains a unique highest-weight word.
- Isomorphisms of connected components correspond to $\equiv_{\mathrm{plac}}$.
- Connected components are indexed by recording tableaux.

Question

Do the hypoplactic and the sylvester monoids admit a ‘quasi-crystal structure’ with these features? (Mutatis mutandis)

Quasi-Kashiwara operators

$$1 \xrightarrow{1} 2 \xrightarrow{2} 3 \xrightarrow{3} \ldots \xrightarrow{n-2} n-1 \xrightarrow{n-1} n$$

Define $\ddot{e}_i$, $\ddot{f}_i$ on $\mathcal{A}_n$ by $a \xrightarrow{i} \ddot{f}_i(a)$ and $\ddot{e}_i(a) \xrightarrow{i} a$.

For $w \in \mathcal{A}_n^* \setminus \mathcal{A}_n$, compute $\ddot{e}_i(w)$ and $\ddot{f}_i(w)$:

- Replace each symbol a of w with
$$+ \quad \text{if } a = i, \qquad - \quad \text{if } a = i+1, \qquad \varepsilon \quad \text{if } a \notin \{i, i+1\}.$$
- If there is a subword $-+$, then $\ddot{e}_i(w)$ and $\ddot{f}_i(w)$ are undefined.
- $\ddot{e}_i(w)$ is obtained by applying $\ddot{e}_i$ to the symbol that was replaced by the leftmost $-$ (if present).
- $\ddot{f}_i(w)$ is obtained by applying $\ddot{f}_i$ to the symbol that was replaced by the rightmost $+$ (if present).

Let $w = 231311$. Computing $\ddot{e}_2(w)$ and $\ddot{f}_2(w)$:

$$2\ 3\ 1\ 3\ 1\ 1$$

Quasi-Kashiwara operators

$$1 \xrightarrow{1} 2 \xrightarrow{2} 3 \xrightarrow{3} \dots \xrightarrow{n-2} n-1 \xrightarrow{n-1} n$$

Define $\ddot{e}_i$, $\ddot{f}_i$ on $\mathcal{A}_n$ by $a \xrightarrow{i} \ddot{f}_i(a)$ and $\ddot{e}_i(a) \xrightarrow{i} a$.

For $w \in \mathcal{A}_n^* \setminus \mathcal{A}_n$, compute $\ddot{e}_i(w)$ and $\ddot{f}_i(w)$:

- Replace each symbol a of w with

$$+ \quad \text{if } a = i, \qquad - \quad \text{if } a = i+1, \qquad \varepsilon \quad \text{if } a \notin \{i, i+1\}.$$

- If there is a subword $-+$, then $\ddot{e}_i(w)$ and $\ddot{f}_i(w)$ are undefined.
- $\ddot{e}_i(w)$ is obtained by applying $\ddot{e}_i$ to the symbol that was replaced by the leftmost $-$ (if present).
- $\ddot{f}_i(w)$ is obtained by applying $\ddot{f}_i$ to the symbol that was replaced by the rightmost $+$ (if present).

Let $w = 231311$. Computing $\ddot{e}_2(w)$ and $\ddot{f}_2(w)$:

$$\begin{array}{cccccc} 2 & 3 & 1 & 3 & 1 & 1 \\ + & - & & - & & \end{array}$$

Quasi-Kashiwara operators

$$1 \xrightarrow{1} 2 \xrightarrow{2} 3 \xrightarrow{3} \ldots \xrightarrow{n-2} n-1 \xrightarrow{n-1} n$$

Define $\ddot{e}_i$, $\ddot{f}_i$ on $\mathcal{A}_n$ by $a \xrightarrow{i} \ddot{f}_i(a)$ and $\ddot{e}_i(a) \xrightarrow{i} a$.

For $w \in \mathcal{A}_n^* \setminus \mathcal{A}_n$, compute $\ddot{e}_i(w)$ and $\ddot{f}_i(w)$:

- Replace each symbol a of w with

  $$+ \quad \text{if } a = i, \qquad - \quad \text{if } a = i+1, \qquad \varepsilon \quad \text{if } a \notin \{i, i+1\}.$$
- If there is a subword $-+$, then $\ddot{e}_i(w)$ and $\ddot{f}_i(w)$ are undefined.
- $\ddot{e}_i(w)$ is obtained by applying $\ddot{e}_i$ to the symbol that was replaced by the leftmost $-$ (if present).
- $\ddot{f}_i(w)$ is obtained by applying $\ddot{f}_i$ to the symbol that was replaced by the rightmost $+$ (if present).

Let $w = 231311$. Computing $\ddot{e}_2(w)$ and $\ddot{f}_2(w)$:

$$\begin{array}{cccccc} 2 & 3 & 1 & 3 & 1 & 1 \\ + & - & & - & & \end{array}$$

$$\ddot{e}_2(w) = 221311 \qquad \ddot{f}_2(w) = 331311$$

Quasi-Kashiwara operators

$$1 \xrightarrow{1} 2 \xrightarrow{2} 3 \xrightarrow{3} \dots \xrightarrow{n-2} n-1 \xrightarrow{n-1} n$$

Define $\ddot{e}_i$, $\ddot{f}_i$ on $\mathcal{A}_n$ by $a \xrightarrow{i} \ddot{f}_i(a)$ and $\ddot{e}_i(a) \xrightarrow{i} a$.

For $w \in \mathcal{A}_n^* \setminus \mathcal{A}_n$, compute $\ddot{e}_i(w)$ and $\ddot{f}_i(w)$:

- Replace each symbol a of w with

$$+ \quad \text{if } a = i, \qquad - \quad \text{if } a = i+1, \qquad \varepsilon \quad \text{if } a \notin \{i, i+1\}.$$

- If there is a subword $-+$, then $\ddot{e}_i(w)$ and $\ddot{f}_i(w)$ are undefined.
- $\ddot{e}_i(w)$ is obtained by applying $\ddot{e}_i$ to the symbol that was replaced by the leftmost $-$ (if present).
- $\ddot{f}_i(w)$ is obtained by applying $\ddot{f}_i$ to the symbol that was replaced by the rightmost $+$ (if present).

Let $w = 231311$. Computing $\ddot{e}_1(w)$ and $\ddot{f}_1(w)$:

$$2\ 3\ 1\ 3\ 1\ 1$$

Quasi-Kashiwara operators

$$1 \xrightarrow{1} 2 \xrightarrow{2} 3 \xrightarrow{3} \ldots \xrightarrow{n-2} n{-}1 \xrightarrow{n-1} n$$

Define $\ddot{e}_i$, $\ddot{f}_i$ on $\mathcal{A}_n$ by $a \xrightarrow{i} \ddot{f}_i(a)$ and $\ddot{e}_i(a) \xrightarrow{i} a$.

For $w \in \mathcal{A}_n^* \setminus \mathcal{A}_n$, compute $\ddot{e}_i(w)$ and $\ddot{f}_i(w)$:

- Replace each symbol a of w with

$$+ \quad \text{if } a = i, \qquad - \quad \text{if } a = i+1, \qquad \varepsilon \quad \text{if } a \notin \{i, i+1\}.$$

- If there is a subword $-+$, then $\ddot{e}_i(w)$ and $\ddot{f}_i(w)$ are undefined.
- $\ddot{e}_i(w)$ is obtained by applying $\ddot{e}_i$ to the symbol that was replaced by the leftmost $-$ (if present).
- $\ddot{f}_i(w)$ is obtained by applying $\ddot{f}_i$ to the symbol that was replaced by the rightmost $+$ (if present).

Let $w = 231311$. Computing $\ddot{e}_1(w)$ and $\ddot{f}_1(w)$:

$$\begin{array}{cccccc} 2 & 3 & 1 & 3 & 1 & 1 \\ - & & + & & + & + \end{array}$$

Quasi-Kashiwara operators

$$1 \xrightarrow{1} 2 \xrightarrow{2} 3 \xrightarrow{3} \ldots \xrightarrow{n-2} n{-}1 \xrightarrow{n-1} n$$

Define $\ddot{e}_i$, $\ddot{f}_i$ on $\mathcal{A}_n$ by $a \xrightarrow{i} \ddot{f}_i(a)$ and $\ddot{e}_i(a) \xrightarrow{i} a$.

For $w \in \mathcal{A}_n^* \setminus \mathcal{A}_n$, compute $\ddot{e}_i(w)$ and $\ddot{f}_i(w)$:

- Replace each symbol a of w with

$$+ \quad \text{if } a = i, \qquad - \quad \text{if } a = i+1, \qquad \varepsilon \quad \text{if } a \notin \{i, i+1\}.$$

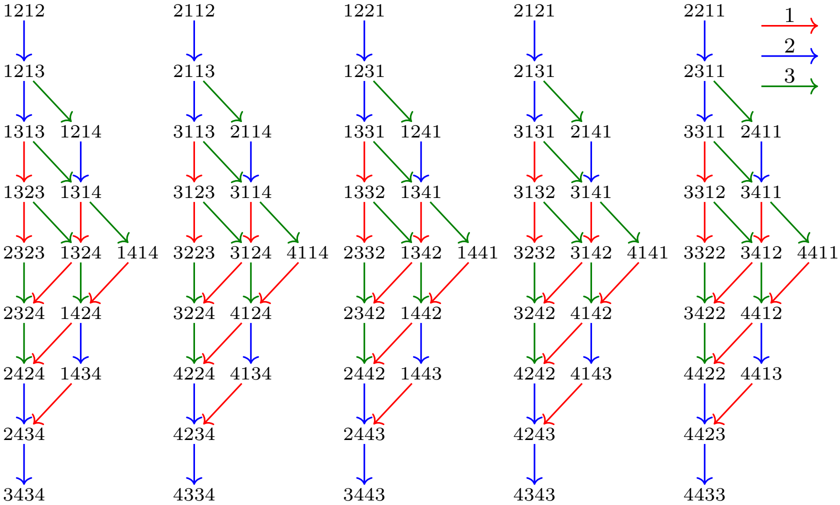
- If there is a subword $-+$, then $\ddot{e}_i(w)$ and $\ddot{f}_i(w)$ are undefined.
- $\ddot{e}_i(w)$ is obtained by applying $\ddot{e}_i$ to the symbol that was replaced by the leftmost $-$ (if present).
- $\ddot{f}_i(w)$ is obtained by applying $\ddot{f}_i$ to the symbol that was replaced by the rightmost $+$ (if present).

Let $w = 231311$. Computing $\ddot{e}_1(w)$ and $\ddot{f}_1(w)$:

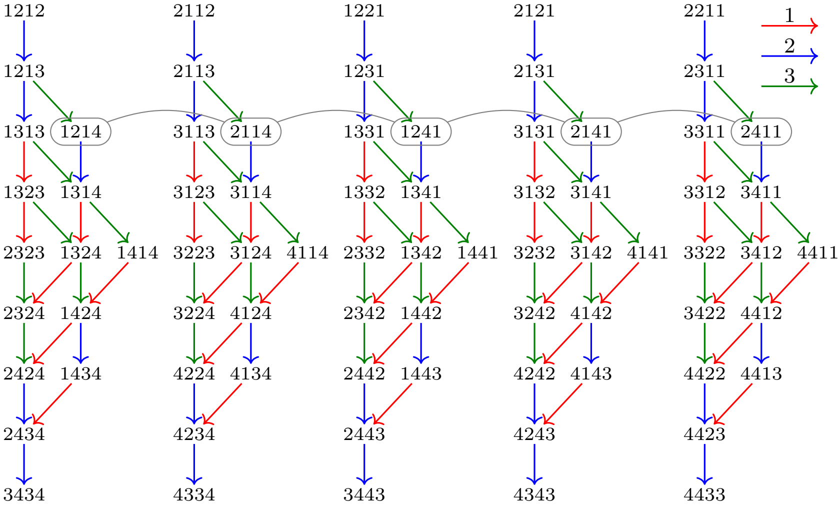
$$\begin{array}{cccccc} 2 & 3 & 1 & 3 & 1 & 1 \\ - & & + & & + & + \end{array}$$

$\ddot{e}_1(w)$ and $\ddot{f}_1(w)$ are undefined.

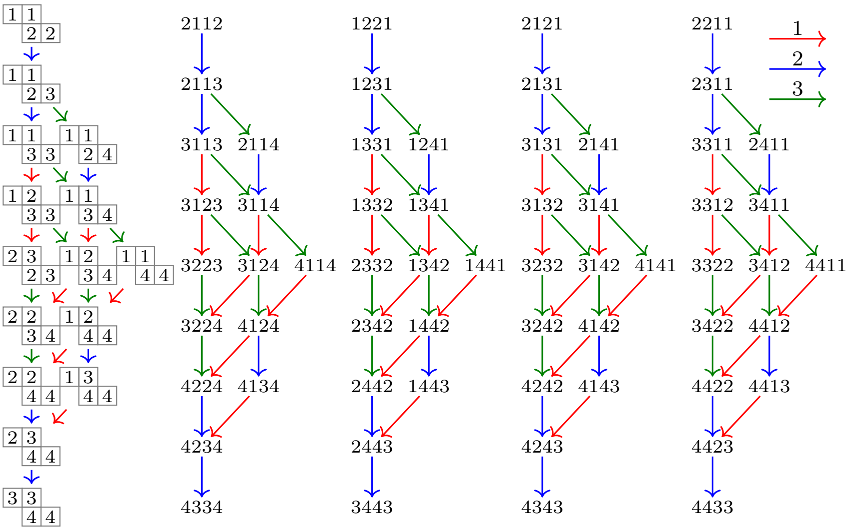
Quasi-crystal graph for hypo_4



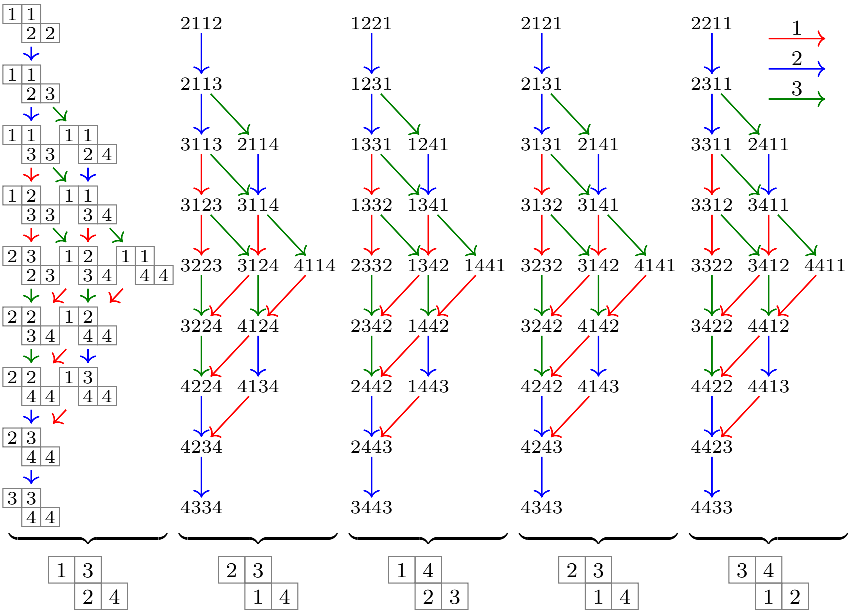
Quasi-crystal graph for hypo_4



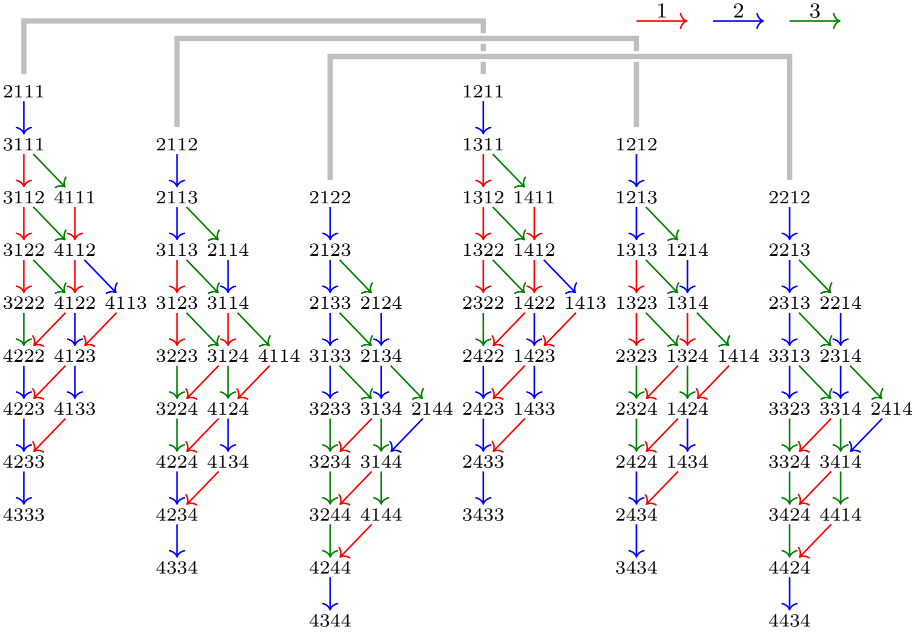
Quasi-crystal graph for hypo_4



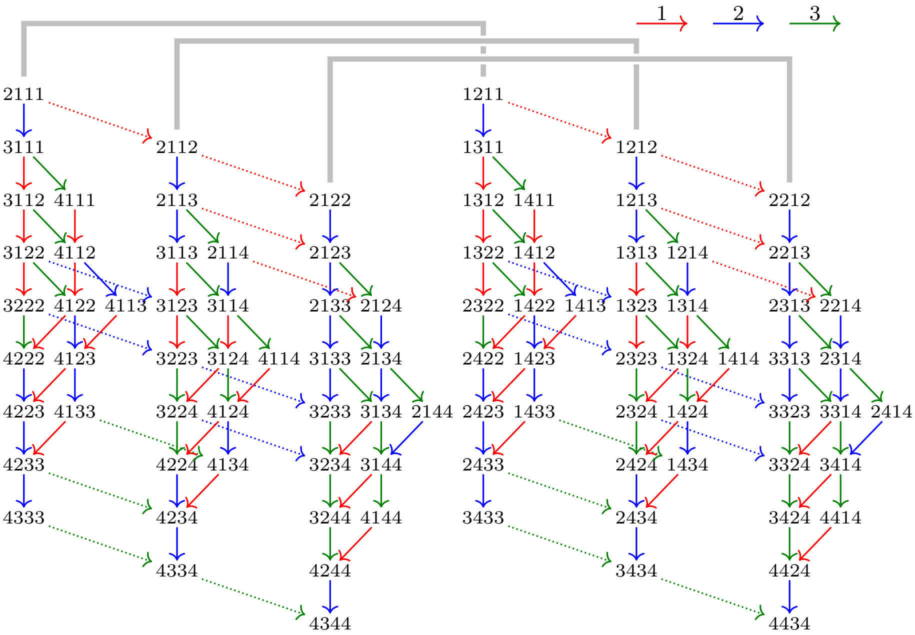
Quasi-crystal graph for hypo_4



Crystals and quasi-crystals



Crystals and quasi-crystals



Standardization and quasi-Kashiwara operators

To standardize 241341:

$$2 \;\; 4 \;\; 1 \;\; 3 \;\; 4 \;\; 1$$

Standardization and quasi-Kashiwara operators

To standardize 241341:

$$\begin{array}{cccccc} 2 & 4 & 1 & 3 & 4 & 1 \end{array}$$
$$2_1 4_1 1_1 3_1 4_2 1_2$$

Standardization and quasi-Kashiwara operators

To standardize 241341:

$$\begin{array}{cccccc} 2 & 4 & 1 & 3 & 4 & 1 \\ 2_1 & 4_1 & 1_1 & 3_1 & 4_2 & 1_2 \\ 3 & 5 & 1 & 4 & 6 & 2 \end{array}$$

Standardization and quasi-Kashiwara operators

To standardize 241341:

$$\begin{array}{cccccc} 2 & 4 & 1 & 3 & 4 & 1 \\ 2_1 & 4_1 & 1_1 & 3_1 & 4_2 & 1_2 \\ 3 & 5 & 1 & 4 & 6 & 2 \end{array}$$

So $\mathrm{std}(241341) = 351462$.

Standardization and quasi-Kashiwara operators

To standardize 241341:

$$\begin{array}{cccccc} 2 & 4 & 1 & 3 & 4 & 1 \\ 2_1 & 4_1 & 1_1 & 3_1 & 4_2 & 1_2 \\ 3 & 5 & 1 & 4 & 6 & 2 \end{array}$$

So $\mathrm{std}(241341) = 351462$.

$$\begin{tikzcd} \begin{array}{ccccccc} 2 & 2 & 4 & 1 & 3 & 4 & 1 \\ + & + & & & - & & \end{array} \arrow[r, "2"] \arrow[d, "\mathrm{std}"] & \begin{array}{ccccccc} 2 & 3 & 4 & 1 & 3 & 4 & 1 \end{array} \arrow[d, "\mathrm{std}"] \\ \begin{array}{ccccccc} 2_1 & 2_2 & 4_1 & 1_1 & 3_1 & 4_2 & 1_2 \\ 3 & 4 & 6 & 1 & 5 & 7 & 2 \end{array} & \begin{array}{ccccccc} 2_1 & 3_1 & 4_1 & 1_1 & 3_2 & 4_2 & 1_2 \\ 3 & 4 & 6 & 1 & 5 & 7 & 2 \end{array} \end{tikzcd}$$

Standardization and quasi-Kashiwara operators

To standardize 241341:

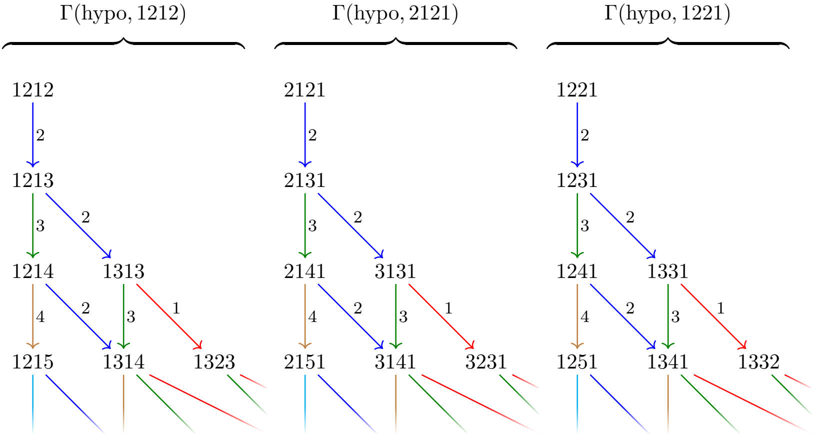
$$\begin{array}{l} 2\ 4\ 1\ 3\ 4\ 1 \\ 2_1 4_1 1_1 3_1 4_2 1_2 \\ 3\ 5\ 1\ 4\ 6\ 2 \end{array}$$

So $\mathrm{std}(241341) = 351462$.

$$\begin{tikzcd}
\begin{array}{l} 2\ 2\ 4\ 1\ 3\ 4\ 1 \\ +\,+\qquad\ - \end{array} \arrow[r, "2"] \arrow[d, "\mathrm{std}"] & 2\ 3\ 4\ 1\ 3\ 4\ 1 \arrow[d, "\mathrm{std}"] \\
\begin{array}{l} 2_1 2_2 4_1 1_1 3_1 4_2 1_2 \\ 3\ 4\ 6\ 1\ 5\ 7\ 2 \end{array} & \begin{array}{l} 2_1 3_1 4_1 1_1 3_2 4_2 1_2 \\ 3\ 4\ 6\ 1\ 5\ 7\ 2 \end{array}
\end{tikzcd}$$

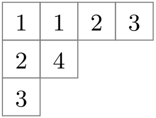
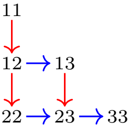
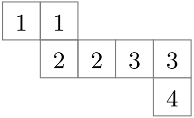
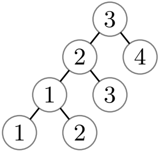
- So $\mathrm{std} \circ \ddot{e}_i = \mathrm{std}$.

Infinite–rank crystals

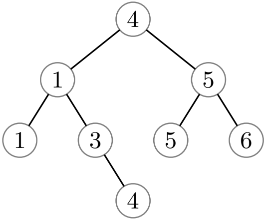


- u and v lie in the same connected component iff $\mathrm{std}(u) = \mathrm{std}(v)$.

Some ‘plactic-like’ monoids

Symmetric functions	Young tableaux 	Plactic monoid	Crystals 
Quasi-symmetric functions	Quasi-ribbon tableaux 	Hypoplactic monoid	?
Loday–Ronco Hopf algebra	Binary search trees 	Sylvester monoid	?

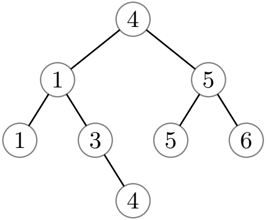
Sylvester

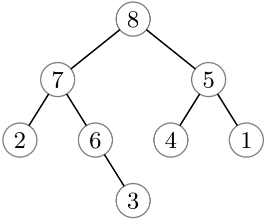


Binary search tree (BST):

- labelled rooted binary tree;
- the label of each node is
  - greater than or equal to the label of every node in its left subtree, and
  - strictly less than the label of every node in its right subtree.

Sylvester





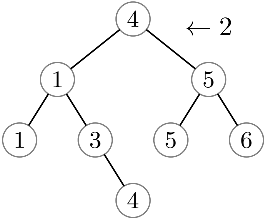
Binary search tree (BST):

- labelled rooted binary tree;
- the label of each node is
  - greater than or equal to the label of every node in its left subtree, and
  - strictly less than the label of every node in its right subtree.

Decreasing tree:

- labelled rooted binary tree;
- the label of each node is greater than the label of its children.

Sylvester

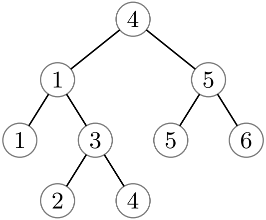


To insert x into a BST T:

- Add x as a leaf node in the unique position that yields a BST.

Result is denoted $T \leftarrow a$.

Sylvester

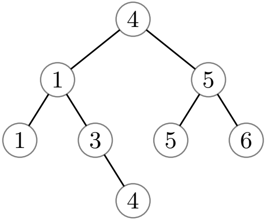


To insert x into a BST T:

- Add x as a leaf node in the unique position that yields a BST.

Result is denoted $T \leftarrow a$.

Sylvester

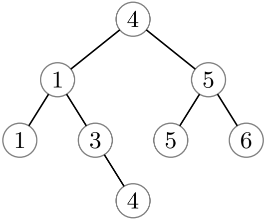


$\mathrm{P}_{\mathrm{sylv}}(61455314)$

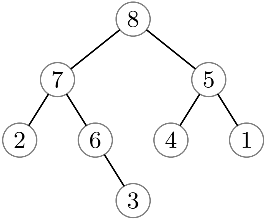
From a word $w_n \cdots w_1 \in \mathcal{A}_n^*$:

- start with an empty BST;
- insert w_1, then w_2, ..., finally w_n;
- call the resulting BST $\mathrm{P}_{\mathrm{sylv}}(w)$.

Sylvester



$\mathrm{P}_{\mathrm{sylv}}(61455314)$



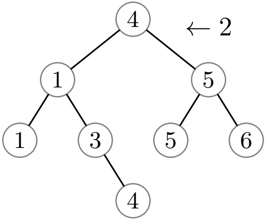
$\mathrm{Q}_{\mathrm{sylv}}(61455314)$

From a word $w_n \cdots w_1 \in \mathcal{A}_n^*$:

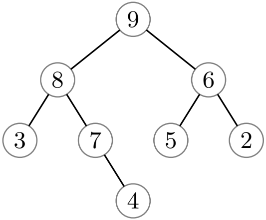
- start with an empty BST;
- insert w_1, then w_2, ..., finally w_n;
- call the resulting BST $\mathrm{P}_{\mathrm{sylv}}(w)$.

- compute a decreasing tree by labelling in order the nodes where symbols are inserted

Sylvester



$\mathrm{P}_{\mathrm{sylv}}(61455314) \leftarrow 2$



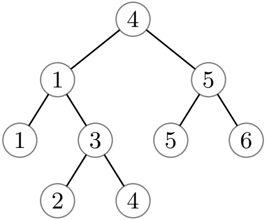
$\mathrm{Q}_{\mathrm{sylv}}(261455314)$

From a word $w_n \cdots w_1 \in \mathcal{A}_n^*$:

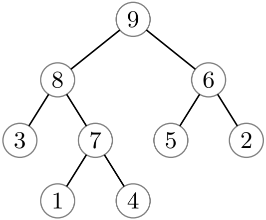
- start with an empty BST;
- insert w_1, then w_2, ..., finally w_n;
- call the resulting BST $\mathrm{P}_{\mathrm{sylv}}(w)$.

- compute a decreasing tree by labelling in order the nodes where symbols are inserted

Sylvester



$\mathrm{P}_{\mathrm{sylv}}(261455314)$



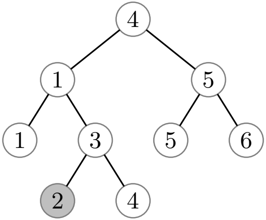
$\mathrm{Q}_{\mathrm{sylv}}(261455314)$

From a word $w_n \cdots w_1 \in \mathcal{A}_n^*$:

- start with an empty BST;
- insert w_1, then w_2, ..., finally w_n;
- call the resulting BST $\mathrm{P}_{\mathrm{sylv}}(w)$.

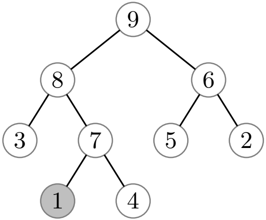
- compute a decreasing tree by labelling in order the nodes where symbols are inserted

Sylvester



From a word $w_n \cdots w_1 \in \mathcal{A}_n^*$:

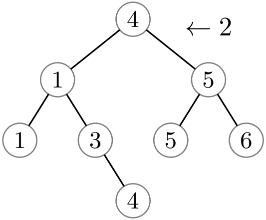
- start with an empty BST;
- insert w_1, then w_2, ..., finally w_n;
- call the resulting BST $\mathrm{P}_{\mathrm{sylv}}(w)$.



- compute a decreasing tree by labelling in order the nodes where symbols are inserted

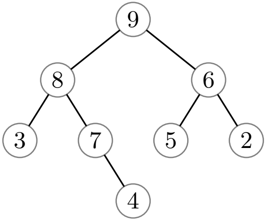
This process can be reversed.

Sylvester



From a word $w_n \cdots w_1 \in \mathcal{A}_n^*$:

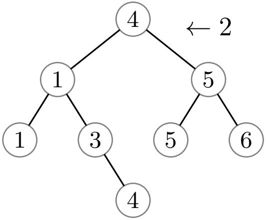
- start with an empty BST;
- insert w_1, then w_2, ..., finally w_n;
- call the resulting BST $\mathrm{P}_{\mathrm{sylv}}(w)$.



- compute a decreasing tree by labelling in order the nodes where symbols are inserted

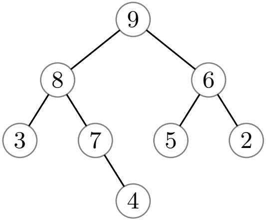
This process can be reversed.

Sylvester



From a word $w_n \cdots w_1 \in \mathcal{A}_n^*$:

- start with an empty BST;
- insert w_1, then w_2, ..., finally w_n;
- call the resulting BST $\mathrm{P}_{\mathrm{sylv}}(w)$.



- compute a decreasing tree by labelling in order the nodes where symbols are inserted

This process can be reversed.

$$w \overset{1:1}{\longleftrightarrow} \big(\mathrm{P}_{\mathrm{sylv}}(w), \mathrm{Q}_{\mathrm{sylv}}(w)\big)$$

Robinson–Schensted correspondence

Sylvester monoid

Define $\equiv_{\mathrm{sylv}}$ on A^* by $u \equiv_{\mathrm{sylv}} v \iff \mathrm{P}_{\mathrm{sylv}}(u) = \mathrm{P}_{\mathrm{sylv}}(v)$.

Theorem (Hivert et al. 2005)

The relation $\equiv_{\mathrm{sylv}}$ is a congruence on $\mathcal{A}_n^*$.

The factor monoid $\mathrm{sylv}_n = \mathcal{A}_n^*/{\equiv_{\mathrm{sylv}}}$ is the sylvester monoid of rank n

sylv_n is presented by

$$\langle\, \mathcal{A}_n \mid \mathcal{R}_{\mathrm{sylv}} \,\rangle,$$

where

$$\mathcal{R}_{\mathrm{sylv}} = \{\, (cavb, acvb) : a \leq b < c, v \in \mathcal{A}_n^* \,\};$$

- Binary search trees form a cross-section.
- hypo_n is a quotient of sylv_n.

Abstract shape

An abstract shape is a map $\sigma : \mathcal{A}^* \to S$ satisfying:

S1 The map σ is invariant under standardization, i.e., for all $u \in \mathcal{A}^*$, $\sigma(u) = \sigma(\mathrm{std}(u))$.

S2 For all $u, v \in \mathcal{A}^*$ and $a \in \mathcal{A}$, if $\mathrm{wt}(u) = \mathrm{wt}(v)$ and $\sigma(u) = \sigma(v)$, then $\sigma(ua) = \sigma(va)$ and $\sigma(au) = \sigma(av)$.

Abstract shape

An abstract shape is a map $\sigma : \mathcal{A}^* \to S$ satisfying:

- **S1** The map σ is invariant under standardization, i.e., for all $u \in \mathcal{A}^*$, $\sigma(u) = \sigma(\mathrm{std}(u))$.
- **S2** For all $u, v \in \mathcal{A}^*$ and $a \in \mathcal{A}$, if $\mathrm{wt}(u) = \mathrm{wt}(v)$ and $\sigma(u) = \sigma(v)$, then $\sigma(ua) = \sigma(va)$ and $\sigma(au) = \sigma(av)$.

A quasi-crystal isomorphism $\theta : \Gamma(\mathrm{hypo}, u) \to \Gamma(\mathrm{hypo}, v)$ is shape-preserving if

$$\sigma(u) = \sigma(v).$$

Define a relation $\sim_\sigma$ on $\mathcal{A}^*$ by:

$$u \sim_\sigma v \quad \Leftrightarrow \exists\, \theta : \Gamma(\mathrm{hypo}, u) \to \Gamma(\mathrm{hypo}, v),\ \theta(u) = v,\ \text{s.-p. isom..}$$

Abstract shape

An abstract shape is a map $\sigma : \mathcal{A}^* \to S$ satisfying:

- **S1** The map σ is invariant under standardization, i.e., for all $u \in \mathcal{A}^*$, $\sigma(u) = \sigma(\mathrm{std}(u))$.
- **S2** For all $u, v \in \mathcal{A}^*$ and $a \in \mathcal{A}$, if $\mathrm{wt}(u) = \mathrm{wt}(v)$ and $\sigma(u) = \sigma(v)$, then $\sigma(ua) = \sigma(va)$ and $\sigma(au) = \sigma(av)$.

A quasi-crystal isomorphism $\theta : \Gamma(\mathrm{hypo}, u) \to \Gamma(\mathrm{hypo}, v)$ is shape-preserving if

$$\sigma(u) = \sigma(v).$$

Define a relation $\sim_\sigma$ on $\mathcal{A}^*$ by:

$$u \sim_\sigma v \quad \Leftrightarrow \exists\, \theta : \Gamma(\mathrm{hypo}, u) \to \Gamma(\mathrm{hypo}, v),\ \theta(u) = v,\ \text{s.-p. isom..}$$

Proposition

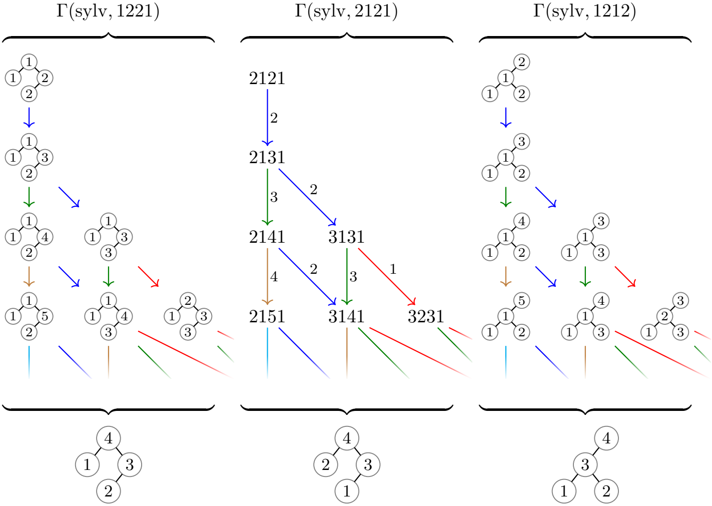
The relation $\sim_\sigma$ is a congruence on the free monoid $\mathcal{A}^*$.

Sylvester monoid

- S - the set of unlabelled binary trees;
- development of notion of abstract shape - σ_{sylv};
- $\sigma_{\mathrm{sylv}}(u) = \mathrm{Sh}(\mathrm{DecT}(\mathrm{std}(u)^{-1}))$ for all $u \in \mathcal{A}^*$.

Sylvester monoid

- S - the set of unlabelled binary trees;
- development of notion of abstract shape - σ_{sylv};
- $\sigma_{\mathrm{sylv}}(u) = \mathrm{Sh}(\mathrm{DecT}(\mathrm{std}(u)^{-1}))$ for all $u \in \mathcal{A}^*$.



Counting

- The number of right strict binary search trees of shape T labelled by symbols from $\mathcal{A}_n$ is

$$\begin{cases} \binom{n+|T|-\ell}{n-\ell} & \text{if } \ell \leq n \\ 0 & \text{if } \ell > n, \end{cases}$$

where ℓ is the number of parts of the right interval partition.

Counting

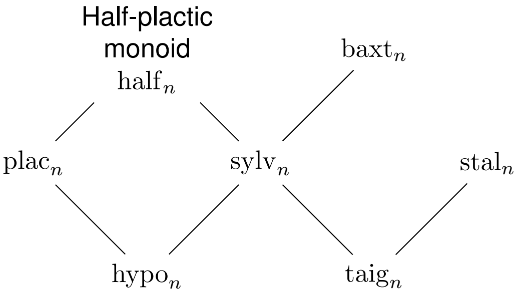
- The number of right strict binary search trees of shape T labelled by symbols from $\mathcal{A}_n$ is

$$\begin{cases} \binom{n+|T|-\ell}{n-\ell} & \text{if } \ell \leq n \\ 0 & \text{if } \ell > n, \end{cases}$$

  where ℓ is the number of parts of the right interval partition.

- The number of distinct factorizations of a right strict binary search tree of shape T, into elements that correspond to right strict binary search trees of shapes U and V, is dependent only of T, U, and V, and not on the content of the element.

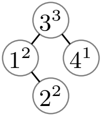
Relationship of plac_n, hypo_n, sylv_n and friends



Taiga monoid

taig_n

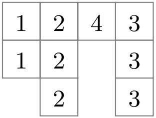
BSTs with multiplicities



Stalactite monoid

stal_n

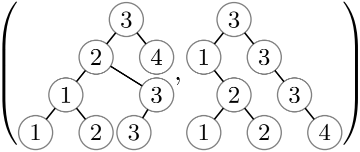
Stalactite tableaux



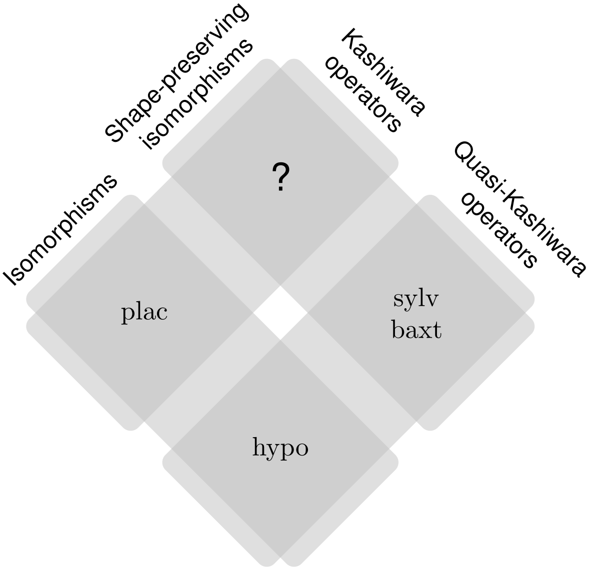
Baxter monoid

baxt_n

Pairs of twin binary search trees



(Quasi-)-Kashiwara operators & shapes



$$\boxed{T}\boxed{H}\boxed{E}$$
$$\boxed{E}\boxed{N}\boxed{D}$$