Multi-parameter hook formula for labelled trees

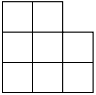
Valentin Féray
joint work with Ian P. Goulden (Waterloo)
and Alain Lascoux (Marne-La-Vallée)

LaBRI, CNRS, Bordeaux

4th biennial Canadian Discrete and Algorithmic Mathematics Conference (CanaDAM)
St John's, Newfoundland, June 13th, 2013

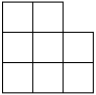
Frame-Robinson-Thrall formula (1954) for counting tableaux

Fix a Young diagram λ with n boxes.

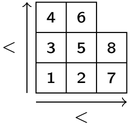


Frame-Robinson-Thrall formula (1954) for counting tableaux

Fix a Young diagram λ with n boxes.



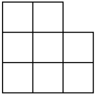
Then the number of standard Young tableaux



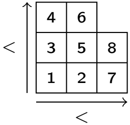
is given by

Frame-Robinson-Thrall formula (1954) for counting tableaux

Fix a Young diagram λ with n boxes.



Then the number of standard Young tableaux

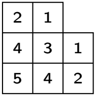


is given by

$$\frac{n!}{\prod_{\square \in \lambda} h_\square}.$$

$h_\square$: hook-length of the box $\square$, *i.e.* number of boxes at its right in the same row or above it in the same column.

In our example: the hook-lengths are



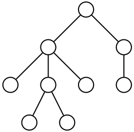
so there are

$8!/(5 * 4 * 4 * 3 * 2 * 2) = 42$ standard Young tableaux of shape λ.

Knuth formula for increasing trees (1973)

The same kind of formula holds for trees!

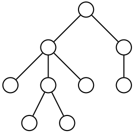
Fix a Tree T with n nodes.



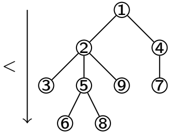
Knuth formula for increasing trees (1973)

The same kind of formula holds for trees!

Fix a Tree T with n nodes.



Then the number of increasing labellings of this tree

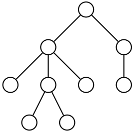


is given by

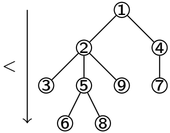
Knuth formula for increasing trees (1973)

The same kind of formula holds for trees!

Fix a Tree T with n nodes.



Then the number of increasing labellings of this tree

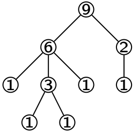


is given by

$$\frac{n!}{\prod_{\circ \in V(T)} h_\circ}.$$

$h_\circ$: hook-length of the vertex $\circ$, *i.e.* the number of vertices in the subtree of T rooted in $\circ$.

In our example: the hook-lengths are



so there are

$9!/(9 * 6 * 3 * 2) = 1120$ increasing labellings of T.

Hook summation formulas

But these objects are in bijection with permutations.

- By Robinson-Schensted algorithm, pairs of standard Young tableaux of the same shape are in bijections with permutations, so

$$\sum_{\lambda \vdash n} \left(\frac{n!}{\prod_{\square \in \lambda} h_{\square}} \right)^2 = n!.$$

Hook summation formulas

But these objects are in bijection with permutations.

- By Robinson-Schensted algorithm, pairs of standard Young tableaux of the same shape are in bijections with permutations, so

$$\sum_{\lambda \vdash n} \left(\frac{n!}{\prod_{\square \in \lambda} h_\square} \right)^2 = n!.$$

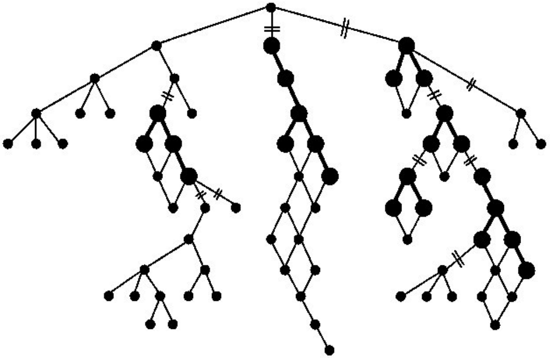
- By binary search tree algorithm, increasing labellings of binary trees are in bijection with permutations, so

$$\sum_{T \text{ binary tree}} \frac{n!}{\prod_{\circ \in V_T} h_\circ} = n!$$

These formulas are called hook summation formulas.

A large amount of work around these hook formulas

- formulas for other objects than trees or Young diagrams: in particular, d-complete posets that include both.



© R. Proctor

A large amount of work around these hook formulas

- formulas for other objects than trees or Young diagrams: in particular, d-complete posets that include both.
- in summation formulas, one can replace $1/h_\square^2$ or $1/h_\circ$ by more involved expressions such that the sum remains simple.

$$\sum_{\substack{T \text{ binary} \\ \text{tree of size } n}} \prod_{v \in T} \left(x + \frac{1}{h_T(v)} \right) = \frac{1}{(n+1)!} \prod_{i=1}^{n-1} \left((n+1+i)x + n+1-i \right).$$

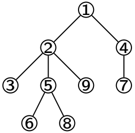
A large amount of work around these hook formulas

- formulas for other objects than trees or Young diagrams: in particular, d-complete posets that include both.
- in summation formulas, one can replace $1/h_\square^2$ or $1/h_\circ$ by more involved expressions such that the sum remains simple.
- interpretations in combinatorial Hopf algebra theory, in convex geometry, in commutative algebra.
- ...

Main result

A hook summation formula over labelled increasing tree with n nodes.

A labelled increasing tree T



Childs of a given vertex are not ordered. By convention, we draw them in increasing order from left to right.

⚠ in our formula, we sum over labelled trees.

Main result

A hook summation formula over labelled increasing tree with n nodes.

Theorem (FGL, 2013)

Let $(x_i)_{1\leq i\leq n}$ and $(y_{i,j})_{1\leq i\leq j\leq n}$ be formal parameters.

$$\sum_T \left[\prod_{i=2}^{n} x_{f_i(T)} \left(\sum_{j\in\mathfrak{h}_i(T)} y_{i,j}\right)\right] = x_1 y_{n,n} \prod_{i=2}^{n-1}\left(y_{i,i}\sum_{j=1}^{i} x_j + x_i \sum_{j=i+1}^{n} y_{i,j}\right).$$

$f_i(T)$: parent of i in T;
$\mathfrak{h}_i(T)$: vertex set of the subtree of T rooted in i.

Example :

$$\text{weight}\left(\begin{array}{c}①\\|\\②\\|\\③\end{array}\right) = x_1(y_{2,2}+y_{2,3})x_2 y_{3,3}$$

Why do we care?

- A specialization of this formula appeared in some computation on irreducible characters of symmetric groups (but *a priori* not related to FRT hook formula)

Why do we care?

- A specialization of this formula appeared in some computation on irreducible characters of symmetric groups (but *a priori* not related to FRT hook formula)
- it has $O(n^2)$ parameters, while formulas in the literature have a fixed number of parameters!

Why do we care?

- A specialization of this formula appeared in some computation on irreducible characters of symmetric groups (but *a priori* not related to FRT hook formula)
- it has $O(n^2)$ parameters, while formulas in the literature have a fixed number of parameters!
- it has a nice product form.

Why do we care?

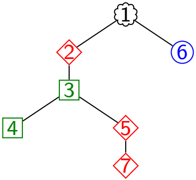
- A specialization of this formula appeared in some computation on irreducible characters of symmetric groups (but *a priori* not related to FRT hook formula)
- it has $O(n^2)$ parameters, while formulas in the literature have a fixed number of parameters!
- it has a nice product form.
- it contains the refined enumeration (by degrees) of Cayley trees. In fact, S. Carrell proved that it is equivalent to some refinement of Cayley formula due to V. Strehl.

Why do we care?

- A specialization of this formula appeared in some computation on irreducible characters of symmetric groups (but *a priori* not related to FRT hook formula)
- it has $O(n^2)$ parameters, while formulas in the literature have a fixed number of parameters!
- it has a nice product form.
- it contains the refined enumeration (by degrees) of Cayley trees. In fact, S. Carrell proved that it is equivalent to some refinement of Cayley formula due to V. Strehl.
- As we shall see, it has a combinatorial flavor.

Combinatorial reformulation

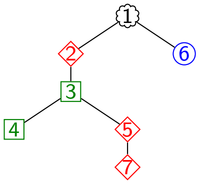
Fix a set-partition of $\{2, \ldots, n\}$ (in the example $\pi = \{\{2,5,7\},\{3,4\},\{6\}\}$). One has to find a bijection between



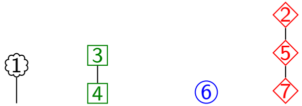
increasing trees T such that, for any two elements in the same part, one is the ancestor of the other.

Combinatorial reformulation

Fix a set-partition of $\{2, \ldots, n\}$ (in the example $\pi = \{\{2,5,7\},\{3,4\},\{6\}\}$). One has to find a bijection between



increasing trees T such that, for any two elements in the same part, one is the ancestor of the other.



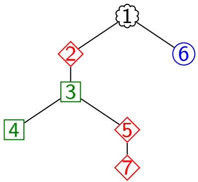
$a_\square = 1 \quad a_\circ = 3 \quad \emptyset$

a number for each part (except the one containing n) less or equal than the maximum of the part (called *anchor point*)

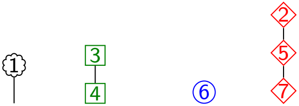
$$a_\square \leq 4, \ \ a_\circ \leq 6.$$

Combinatorial reformulation

Fix a set-partition of $\{2, \dots, n\}$ (in the example $\pi = \{\{2,5,7\},\{3,4\},\{6\}\}$). One has to find a bijection between



increasing trees T such that, for any two elements in the same part, one is the ancestor of the other.



$a_\square = 1 \quad a_\circ = 3 \quad \emptyset$

a number for each part (except the one containing n) less or equal than the maximum of the part (called *anchor point*)

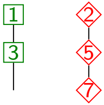
$$a_\square \leq 4, \; a_\circ \leq 6.$$

which respects the degree:

$$\deg_{\text{left}}(i) = \deg_{\text{right}}(i) + |a^{-1}(i)|.$$

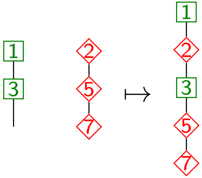
Elementary splicing

Consider two chains, the first one ending by a free edge.



Elementary splicing

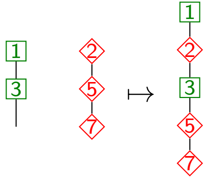
Consider two chains, the first one ending by a free edge.



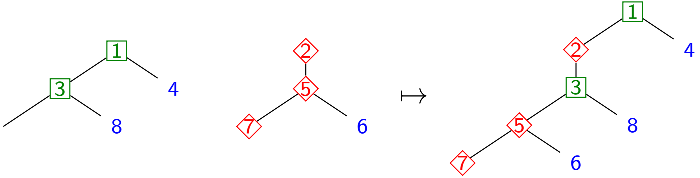
If $\max_{\diamond} \geq \max_{\square}$, we can splice the two chains in a canonical way.

Elementary splicing

Consider two chains, the first one ending by a free edge.



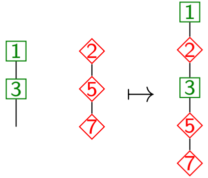
If $\max_{\diamond} \geq \max_{\square}$, we can splice the two chains in a canonical way.



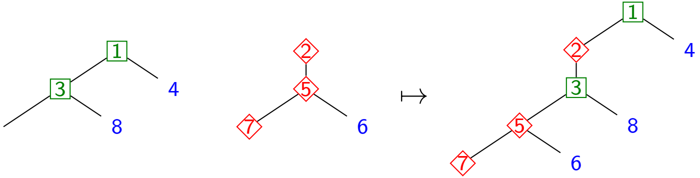
Works with additional vertices.

Elementary splicing

Consider two chains, the first one ending by a free edge.



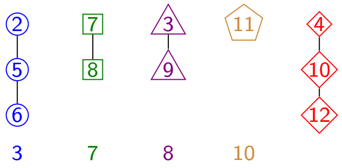
If $\max_{\diamond} \geq \max_{\square}$, we can splice the two chains in a canonical way.



Works with additional vertices.

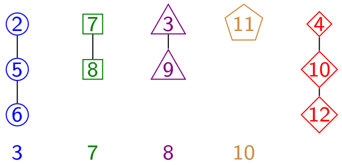
⚠ We must specify a chain on the second tree. We will always choose the one ending by the vertex with maximum label.

The bijection on an example



Start with the set of chains above with anchor points.

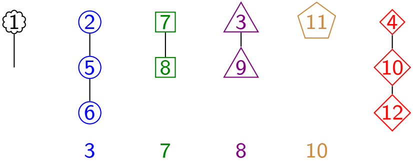
The bijection on an example



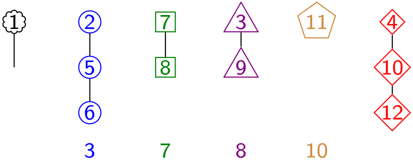
Start with the set of chains above with anchor points.

Step 0: we add a root labeled 1 with a free edge to the list.

The bijection on an example



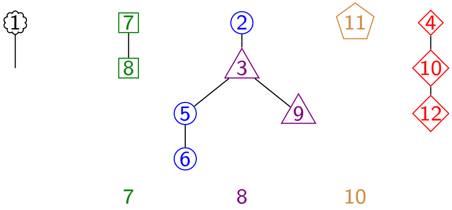
The bijection on an example



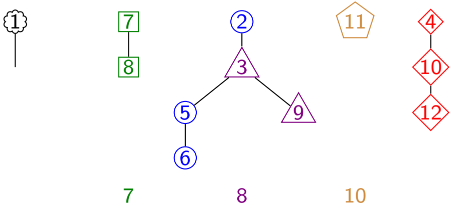
We will splice successively the chains together.

First step: we add a free edge to 3 and splice 2, 5, 6 with 3, 9 (*external splice*).

The bijection on an example

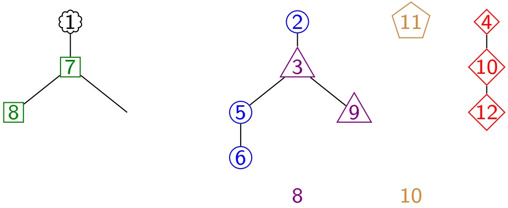


The bijection on an example

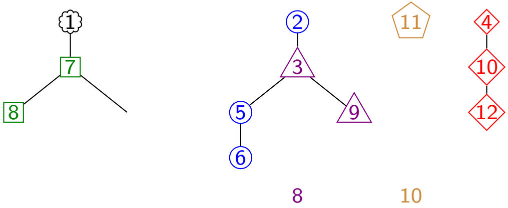


Second step: 7 is in the component we must splice. Thus, we splice 7, 8 on the free edge and add a free edge to 7 (*internal splice*).

The bijection on an example

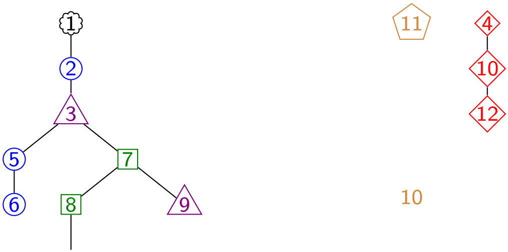


The bijection on an example

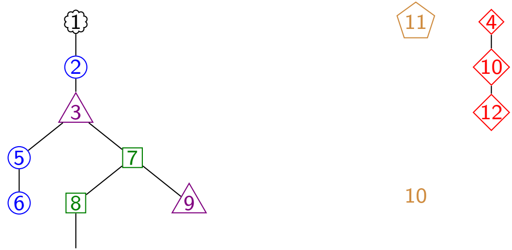


Third step: 8 is in the root component ⇒ again an internal splice. We splice the tree $2, 3, 5, 6, 9$ onto the free edge and add a free edge to 8.

The bijection on an example

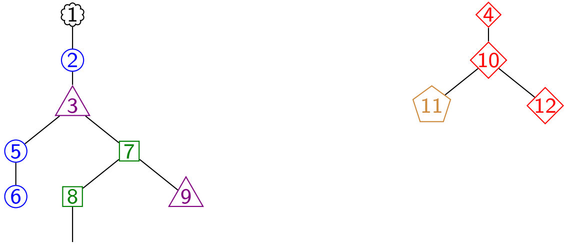


The bijection on an example



Fourth step: an external splice. We add a free edge to 10 and splice 11 onto it.

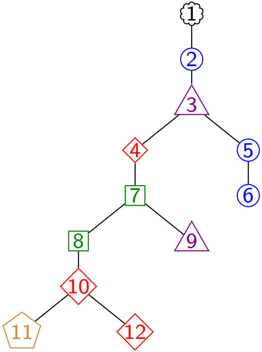
The bijection on an example



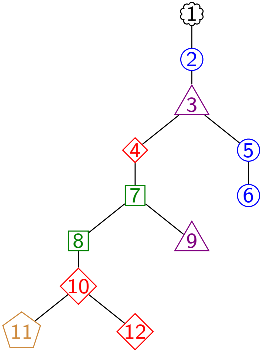
The bijection on an example

Last step: we splice the tree containing the maximum onto the free edge.

The bijection on an example



The bijection on an example



Here is the resulting partitioned tree.
The degree condition is fulfilled by construction.

Summary and conclusion

Construction by successive splicings:

- if the anchor point is in the component itself or in the root component, we splice onto the free edge and add an edge to the anchor point (internal splicing).
- if the anchor point is in another component, we add a free edge to the anchor point and splice the tree on this free edge (external splicing).

Summary and conclusion

Construction by successive splicings:

- if the anchor point is in the component itself or in the root component, we splice onto the free edge and add an edge to the anchor point (internal splicing).
- if the anchor point is in another component, we add a free edge to the anchor point and splice the tree on this free edge (external splicing).

Theorem (FGL, 2013)

The described procedure defines a bijection.

Summary and conclusion

Construction by successive splicings.

- if the anchor point is in the component itself or in the root component, we splice onto the free edge and add an edge to the anchor point (internal splicing).
- if the anchor point is in another component, we add a free edge to the anchor point and splice the tree on this free edge (external splicing).

Theorem (FGL, 2013)

The described procedure defines a bijection.

Corollary (FGL, 2013)

$$\sum_T \left[\prod_{i=1}^{n} x_{f_i(T)} \left(\sum_{j \in \mathfrak{h}_i(T)} y_{i,j} \right) \right] = x_1 y_{n,n} \prod_{i=2}^{n-1} \left(y_{i,i} \sum_{j=1}^{i} x_j + x_i \sum_{j=i+1}^{n} y_{i,j} \right).$$