

# Combinatorics of non-ambiguous trees

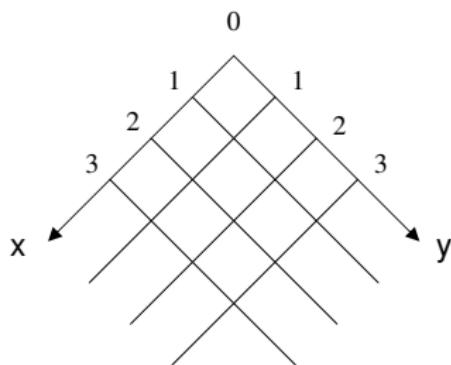
Matteo Silimbani  
LaBRI - Université Bordeaux 1

Journées de combinatoire de Bordeaux 2013 - 14 Février

joint work with J.-C. Aval, A. Boussicault and M. Bouvel

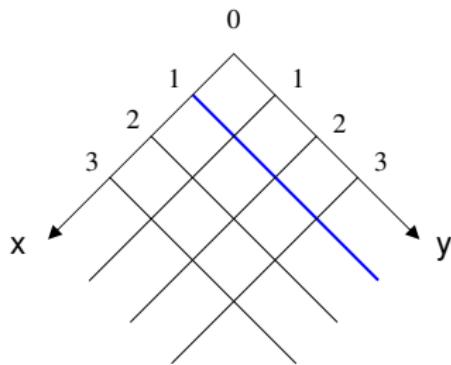
# Some conventions

Objects will be drawn in a  $\mathbb{N} \times \mathbb{N}$  grid.



# Some conventions

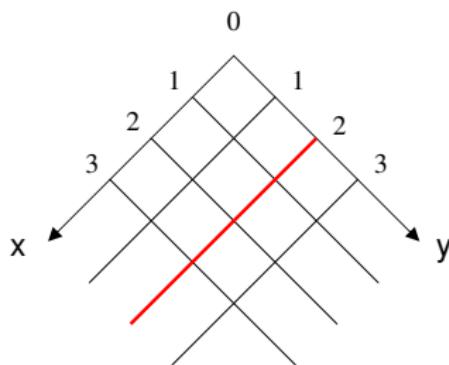
Objects will be drawn in a  $\mathbb{N} \times \mathbb{N}$  grid.



*y*-oriented lines = rows

# Some conventions

Objects will be drawn in a  $\mathbb{N} \times \mathbb{N}$  grid.

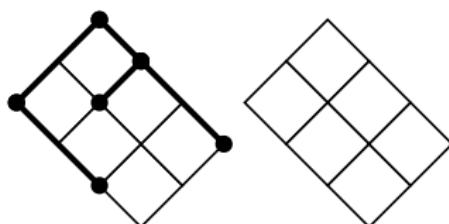


y-oriented lines = rows

x-oriented lines = columns

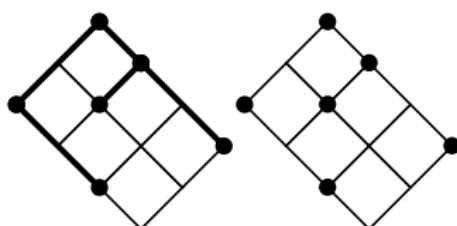
# Intuitive definition of non ambiguous trees

A non-ambiguous tree is a binary tree embedded in the grid in such a way that the embedding of its vertices in the grid determines the tree completely (i.e. determines its edges)



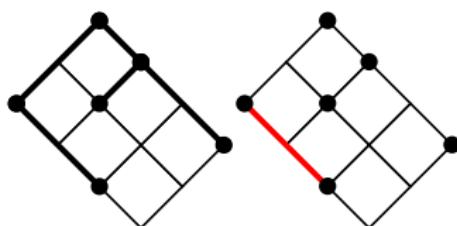
# Intuitive definition of non ambiguous trees

A non-ambiguous tree is a binary tree embedded in the grid in such a way that the embedding of its vertices in the grid determines the tree completely (i.e. determines its edges)



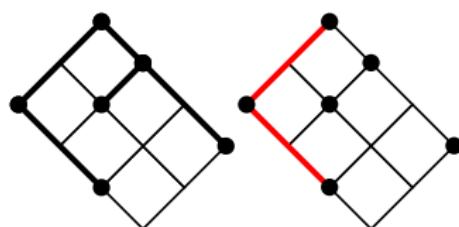
# Intuitive definition of non ambiguous trees

A non-ambiguous tree is a binary tree embedded in the grid in such a way that the embedding of its vertices in the grid determines the tree completely (i.e. determines its edges)



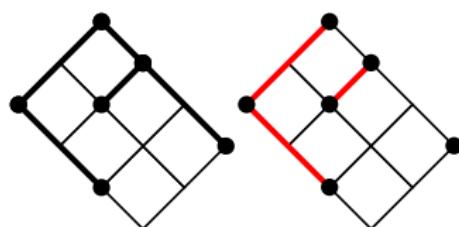
# Intuitive definition of non ambiguous trees

A non-ambiguous tree is a binary tree embedded in the grid in such a way that the embedding of its vertices in the grid determines the tree completely (i.e. determines its edges)



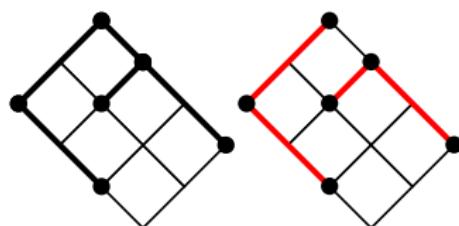
# Intuitive definition of non ambiguous trees

A non-ambiguous tree is a binary tree embedded in the grid in such a way that the embedding of its vertices in the grid determines the tree completely (i.e. determines its edges)



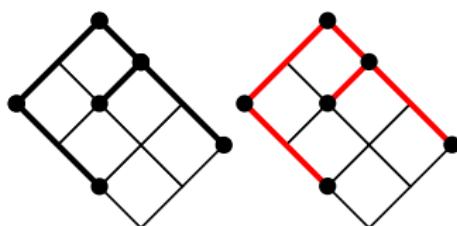
# Intuitive definition of non ambiguous trees

A non-ambiguous tree is a binary tree embedded in the grid in such a way that the embedding of its vertices in the grid determines the tree completely (i.e. determines its edges)



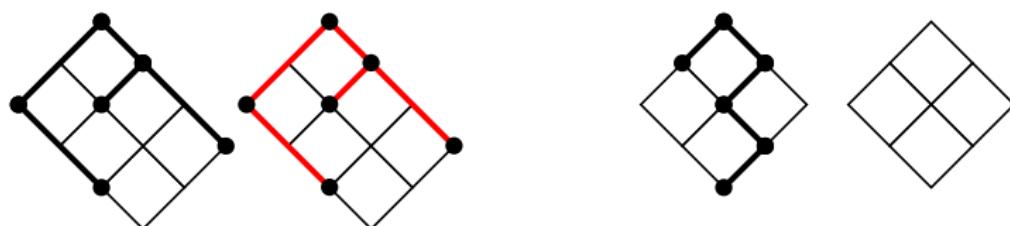
# Intuitive definition of non ambiguous trees

A non-ambiguous tree is a binary tree embedded in the grid in such a way that the embedding of its vertices in the grid determines the tree completely (i.e. determines its edges)



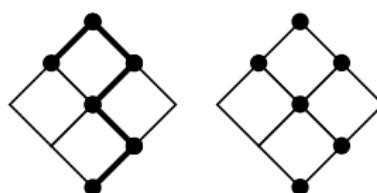
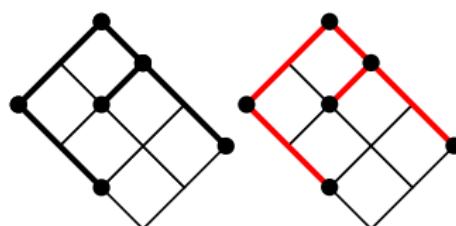
# Intuitive definition of non ambiguous trees

A non-ambiguous tree is a binary tree embedded in the grid in such a way that the embedding of its vertices in the grid determines the tree completely (i.e. determines its edges)



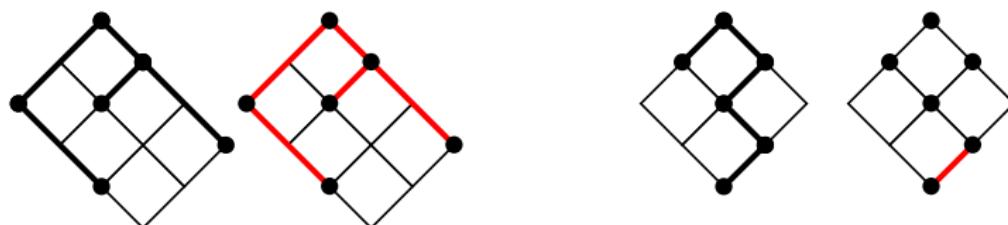
# Intuitive definition of non ambiguous trees

A non-ambiguous tree is a binary tree embedded in the grid in such a way that the embedding of its vertices in the grid determines the tree completely (i.e. determines its edges)



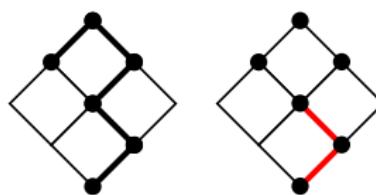
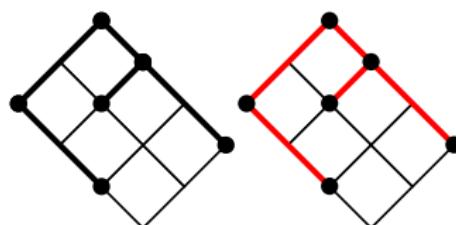
# Intuitive definition of non ambiguous trees

A non-ambiguous tree is a binary tree embedded in the grid in such a way that the embedding of its vertices in the grid determines the tree completely (i.e. determines its edges)



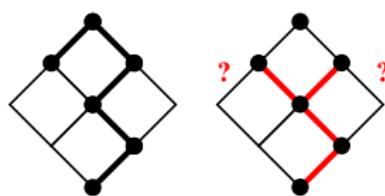
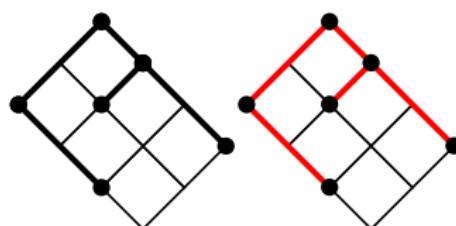
# Intuitive definition of non ambiguous trees

A non-ambiguous tree is a binary tree embedded in the grid in such a way that the embedding of its vertices in the grid determines the tree completely (i.e. determines its edges)



# Intuitive definition of non ambiguous trees

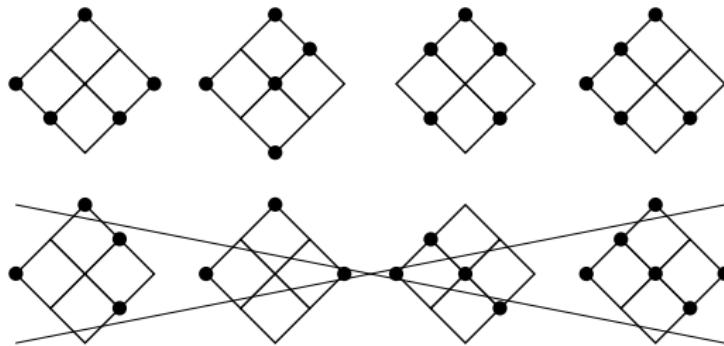
A non-ambiguous tree is a binary tree embedded in the grid in such a way that the embedding of its vertices in the grid determines the tree completely (i.e. determines its edges)



# Definition of a non-ambiguous tree

A non-ambiguous tree of size  $n$  is a set  $A$  of  $n$  points  $(x, y) \in \mathbb{N} \times \mathbb{N}$  such that:

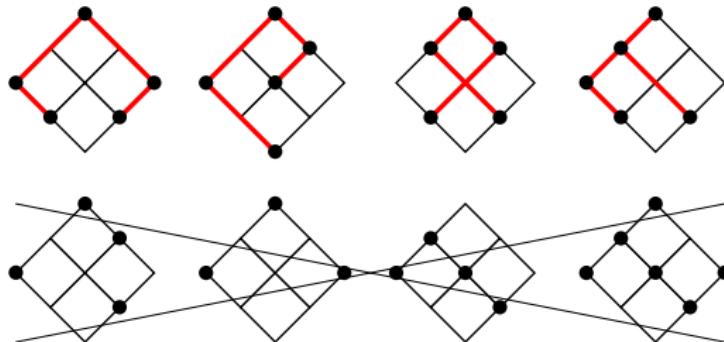
- $(0, 0) \in A$  (the root)
- every point (except the root) has a “parent”
- the pattern  is forbidden
- compactness: there is no empty row/column



# Definition of a non-ambiguous tree

A non-ambiguous tree of size  $n$  is a set  $A$  of  $n$  points  $(x, y) \in \mathbb{N} \times \mathbb{N}$  such that:

- $(0, 0) \in A$  (the root)
- every point (except the root) has a “parent”
- the pattern  is forbidden
- compactness: there is no empty row/column



# The context

Permutation tableaux and alternative tableaux are objects used to study:

- the PASEP model in physics;
- 2 – 31 pattern inside a permutation;
- excedences and cycles of a permutation;
- Laguerre polynomials;
- ...

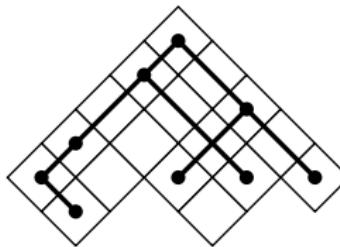
[Burstein, Corteel, Dasse-Hartaut, Hitczenko, Josuat-Vergès, Nadeau, Postnikov, Steingrímsson, Viennot, Williams, Kim, 2005-2011]

# The context

Tree-like tableaux have been introduced to simplify and to explain some of the previous results.

[Aval, Boussicault, Nadeau, 2011]

A tree-like tableau can be defined as a Ferrers diagram containing a non-ambiguous tree



# How many?

$a_n = \#$  of non-ambiguous trees of size  $n$

$$n = 2 \quad \rightarrow \quad a_2 = 2$$

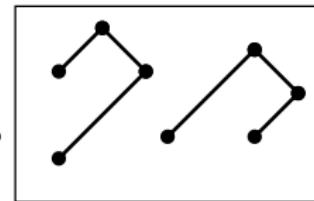
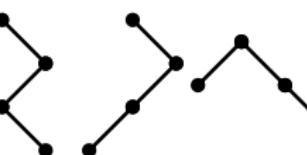
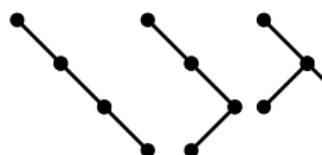
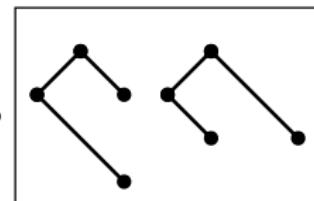
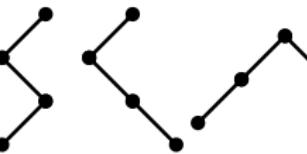
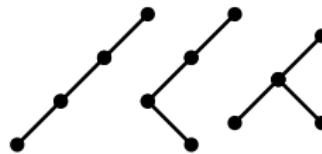


$$n = 3 \quad \rightarrow \quad a_3 = 5$$



# How many?

$$n = 4 \quad \leftarrow \quad a_4 = 16$$



# How many?

On the OEIS we find

$$\text{A136127} = (a_k)_{k \geq 1} = (1, 2, 5, 16, 63, 294, 1585, 9692, \dots)$$

# How many?

On the OEIS we find

$$\text{A136127} = (a_k)_{k \geq 1} = (1, 2, 5, 16, 63, 294, 1585, 9692, \dots)$$

The number  $a_n$  of non-ambiguous trees of size  $n$  is equal to the number of permutations having all their excedences placed at the beginning.

[Steingrímsson, Williams, 07]

# How many?

On the OEIS we find

$$\text{A136127} = (a_k)_{k \geq 1} = (1, 2, 5, 16, 63, 294, 1585, 9692, \dots)$$

The number  $a_n$  of non-ambiguous trees of size  $n$  is equal to the number of permutations having all their excedences placed at the beginning.

[Steingrímsson, Williams, 07]

$$a_n = \sum_{k+l=n-1} \left( \sum_{i=1}^k (-1)^{k-i} S(k, i) i! i^{l-1} \right)$$

where  $S(k, i)$  are the Stirling numbers of the second kind.

[Ehrenborg, Steingrímsson, 2000]

Combinatorial proof with non ambiguous trees and tree-like tableaux

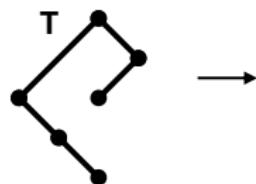
[Aval, Boussicault, Bouvel, and S. 2012]

# Outline of the talk

- 1 Enumeration of non-ambiguous trees with a given underlying binary tree
- 2 Complete non-ambiguous trees and the Bessel function
- 3 A bijection between parallelogram polyominoes and binary trees

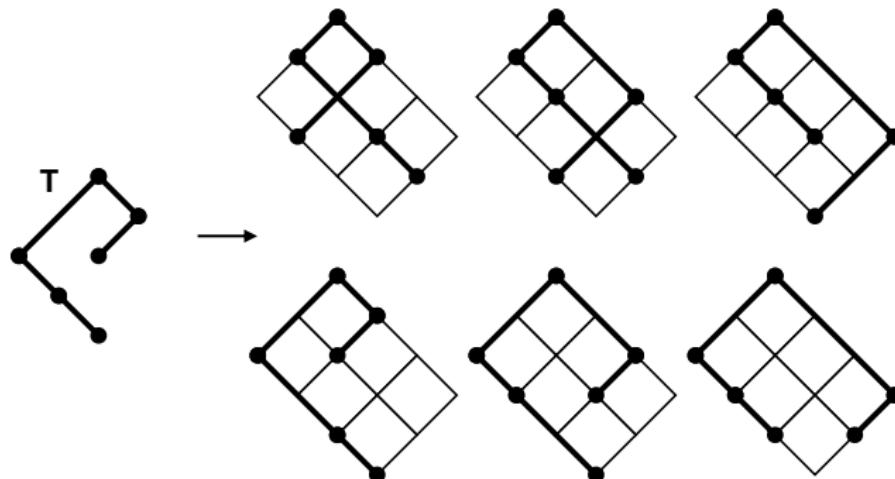
# Non-ambiguous trees associated with a given binary tree

$NA(T) = \#$  non-ambiguous trees whose underlying binary tree is  $T$ .



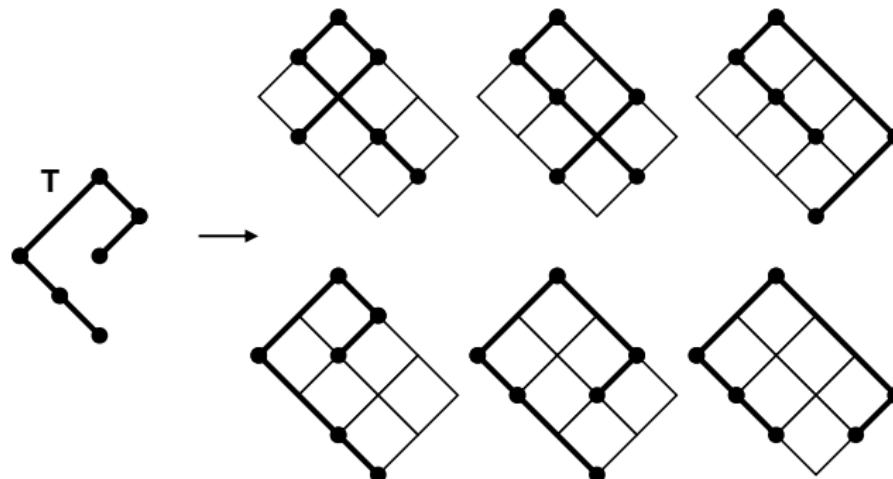
# Non-ambiguous trees associated with a given binary tree

$NA(T) = \#$  non-ambiguous trees whose underlying binary tree is  $T$ .



# Non-ambiguous trees associated with a given binary tree

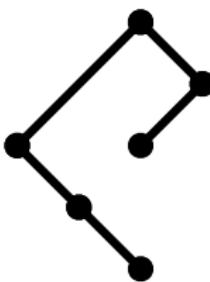
$NA(T) = \#$  non-ambiguous trees whose underlying binary tree is  $T$ .



In this example,  $NA(T) = 6$ .

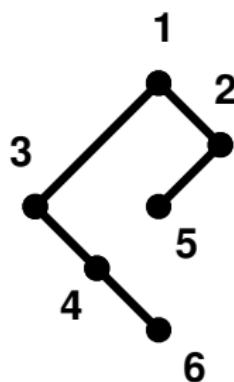
# Non-ambiguous trees associated with a given binary tree

To distinguish the vertices of a binary tree  $T$ , we label them by integers to 1 to  $|T|$ :



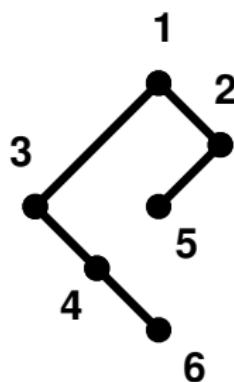
# Non-ambiguous trees associated with a given binary tree

To distinguish the vertices of a binary tree  $T$ , we label them by integers to 1 to  $|T|$ :



# Non-ambiguous trees associated with a given binary tree

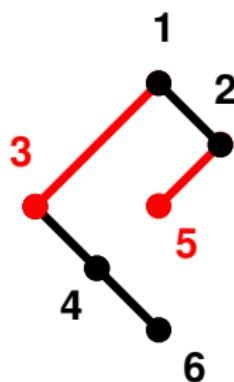
To distinguish the vertices of a binary tree  $T$ , we label them by integers to 1 to  $|T|$ :



$$V_L = \{ \text{ left children } \}$$

# Non-ambiguous trees associated with a given binary tree

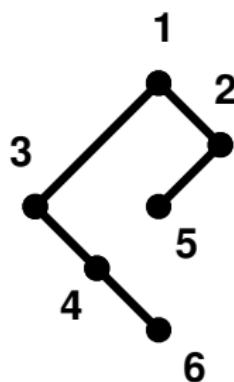
To distinguish the vertices of a binary tree  $T$ , we label them by integers to 1 to  $|T|$ :



$$V_L = \{ \text{left children} \} \text{ (here } V_L = \{3, 5\}\text{)}$$

# Non-ambiguous trees associated with a given binary tree

To distinguish the vertices of a binary tree  $T$ , we label them by integers to 1 to  $|T|$ :

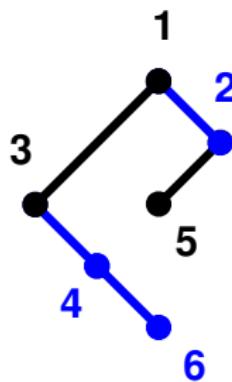


$$V_L = \{ \text{left children} \} \text{ (here } V_L = \{3, 5\}\text{)}$$

$$V_R = \{ \text{right children} \}$$

# Non-ambiguous trees associated with a given binary tree

To distinguish the vertices of a binary tree  $T$ , we label them by integers to 1 to  $|T|$ :

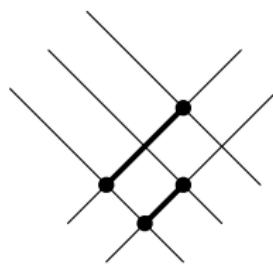
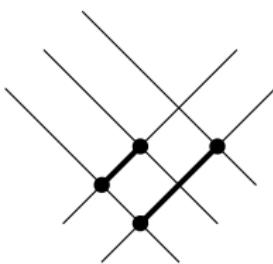


$$V_L = \{ \text{left children} \} \text{ (here } V_L = \{3, 5\}\text{)}$$

$$V_R = \{ \text{right children} \} \text{ (here } V_R = \{2, 4, 6\}\text{)}$$

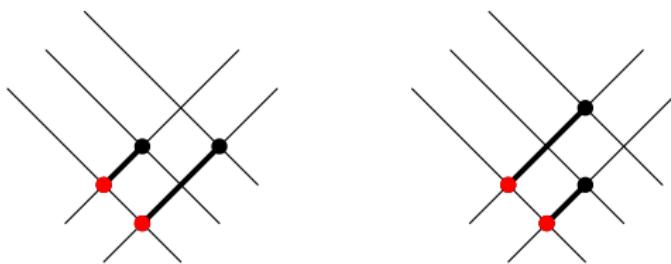
# The coordinates of left and right children

Two left children cannot belong to the same row



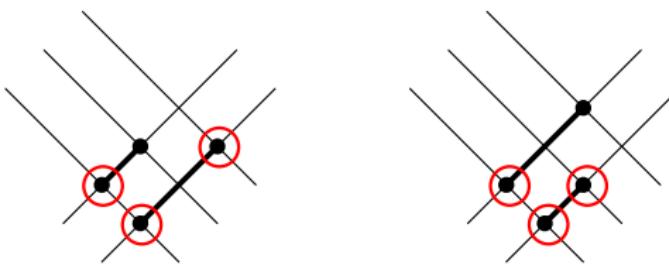
# The coordinates of left and right children

Two left children cannot belong to the same row



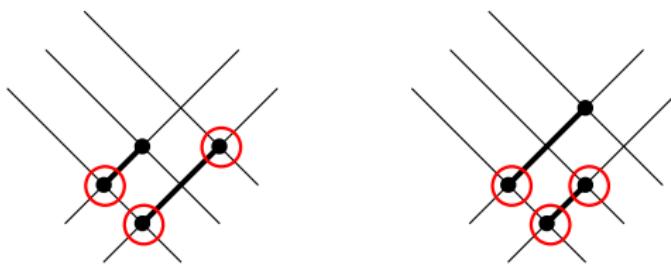
# The coordinates of left and right children

Two left children cannot belong to the same row



# The coordinates of left and right children

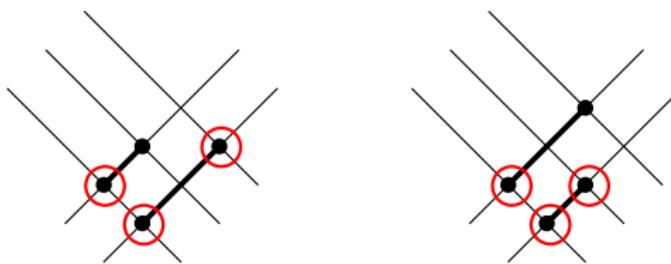
Two left children cannot belong to the same row



Hence, all the elements in  $V_L$  have different  $x$ -coordinates, and these coordinates form the interval  $[1, |V_L|]$ .

# The coordinates of left and right children

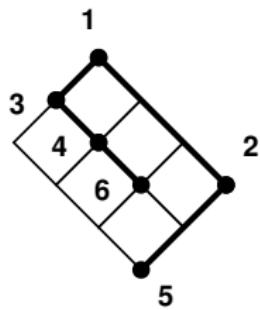
Two left children cannot belong to the same row



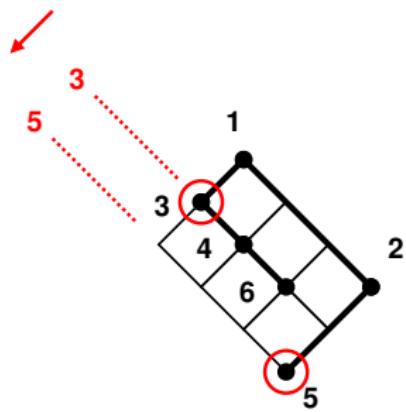
Hence, all the elements in  $V_L$  have different  $x$ -coordinates, and these coordinates form the interval  $[1, |V_L|]$ .

For the same reason, all the elements in  $V_R$  have different  $y$ -coordinates, and these coordinates form the interval  $[1, |V_R|]$

# The words $\alpha_L$ and $\alpha_R$

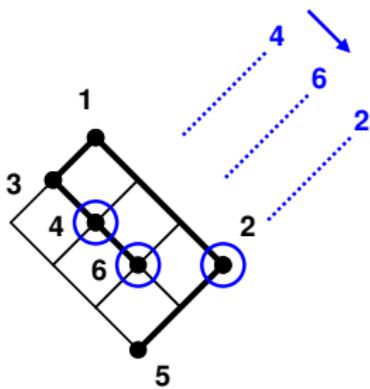


# The words $\alpha_L$ and $\alpha_R$



$$\alpha_L = 35$$

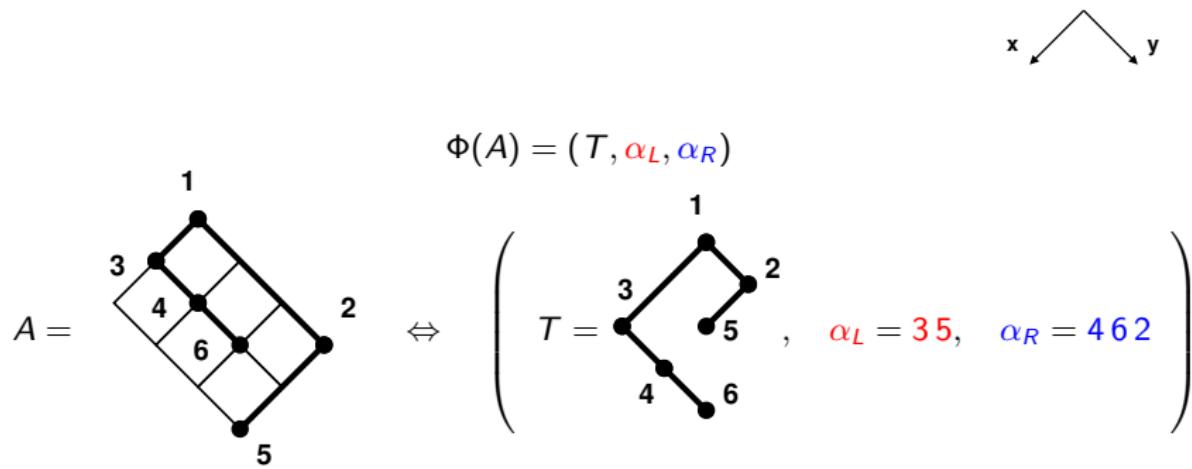
# The words $\alpha_L$ and $\alpha_R$



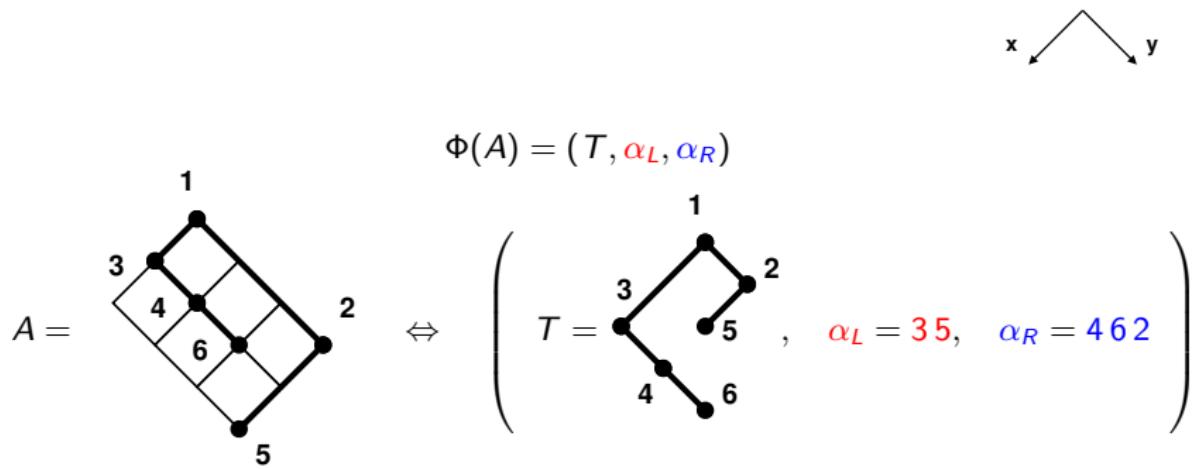
$$\alpha_L = 35$$

$$\alpha_R = 462$$

# A triplet representing a non-ambiguous tree



# A triplet representing a non-ambiguous tree



We can reconstruct  $A$  from  $\Phi(A)$ :  $X(1) = Y(1) = 0$  and

$$\begin{cases} X(i) = \alpha_L^{-1}(i) \text{ and } Y(i) = Y(\text{parent}(i)) & i \text{ left child} \\ Y(j) = \alpha_R^{-1}(j) \text{ and } X(j) = X(\text{parent}(j)) & j \text{ right child} \end{cases}$$

# A triplet representing a non-ambiguous tree

$$A = \begin{array}{c} \text{1} \\ \diagdown \quad \diagup \\ \square \quad \square \\ \diagup \quad \diagdown \\ \square \quad \square \\ \diagdown \quad \diagup \\ \square \end{array} \Leftrightarrow \left( \Phi(A) = (T, \alpha_L, \alpha_R) \right)$$

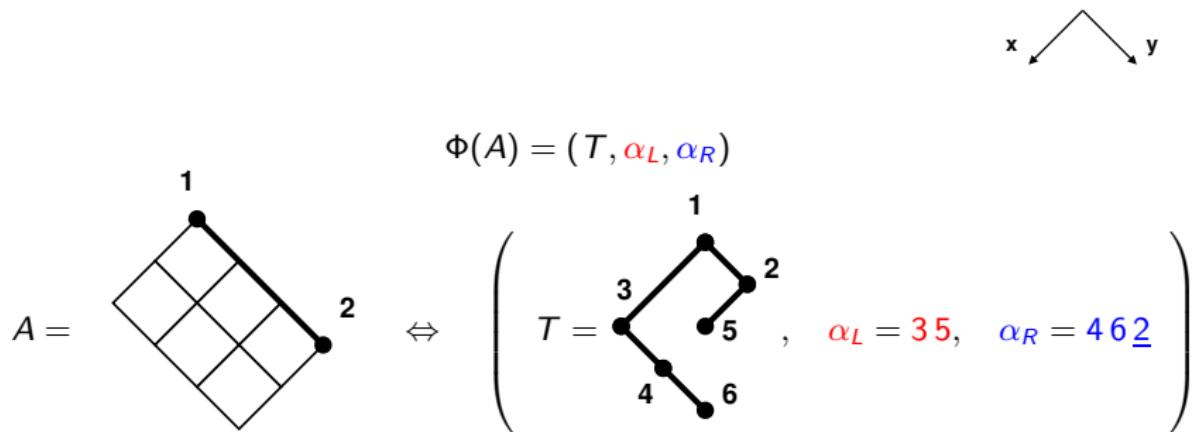
$T = \begin{array}{c} \text{1} \\ \diagdown \quad \diagup \\ \text{3} \quad \text{5} \\ \diagup \quad \diagdown \\ \text{4} \quad \text{6} \end{array}, \quad \alpha_L = 35, \quad \alpha_R = 462$



We can reconstruct  $A$  from  $\Phi(A)$ :  $X(1) = Y(1) = 0$  and

$$\begin{cases} X(i) = \alpha_L^{-1}(i) \text{ and } Y(i) = Y(\text{parent}(i)) & i \text{ left child} \\ Y(j) = \alpha_R^{-1}(j) \text{ and } X(j) = X(\text{parent}(j)) & j \text{ right child} \end{cases}$$

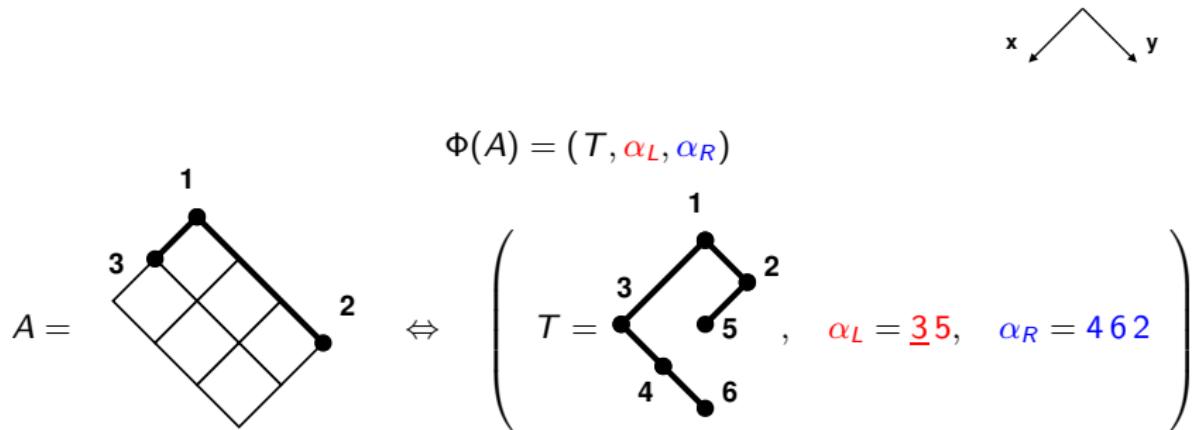
# A triplet representing a non-ambiguous tree



We can reconstruct  $A$  from  $\Phi(A)$ :  $X(1) = Y(1) = 0$  and

$$\begin{cases} X(i) = \alpha_L^{-1}(i) \text{ and } Y(i) = Y(\text{parent}(i)) & i \text{ left child} \\ Y(j) = \alpha_R^{-1}(j) \text{ and } X(j) = X(\text{parent}(j)) & j \text{ right child} \end{cases}$$

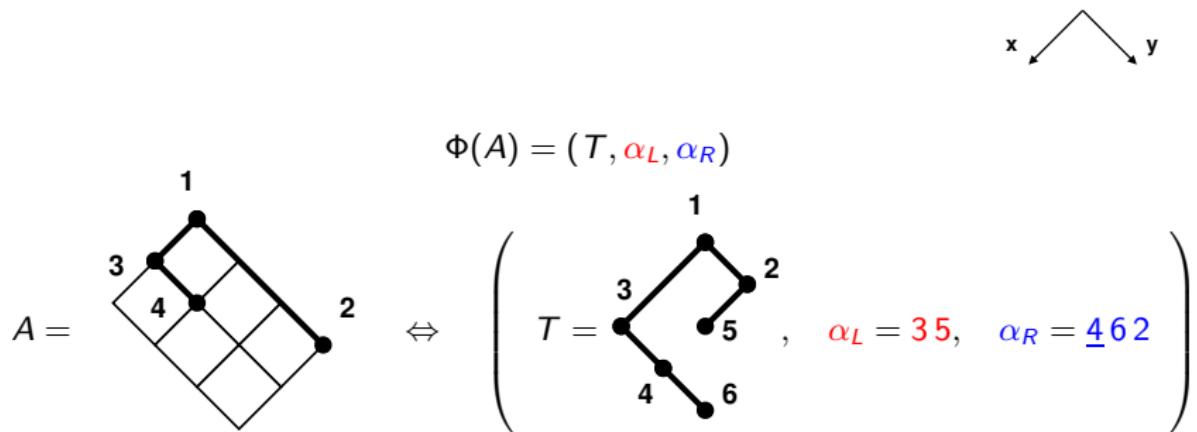
# A triplet representing a non-ambiguous tree



We can reconstruct  $A$  from  $\Phi(A)$ :  $X(1) = Y(1) = 0$  and

$$\begin{cases} X(i) = \alpha_L^{-1}(i) \text{ and } Y(i) = Y(\text{parent}(i)) & i \text{ left child} \\ Y(j) = \alpha_R^{-1}(j) \text{ and } X(j) = X(\text{parent}(j)) & j \text{ right child} \end{cases}$$

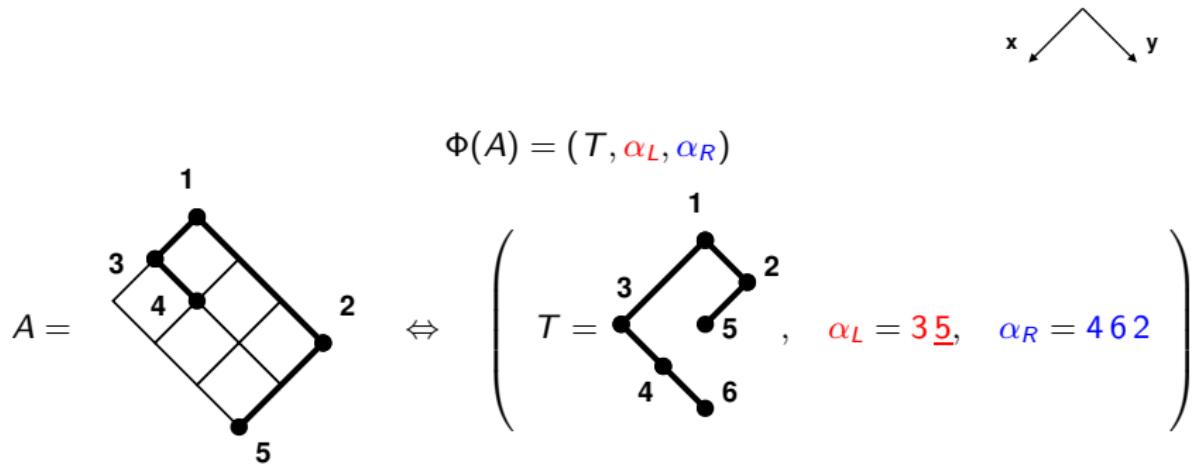
# A triplet representing a non-ambiguous tree



We can reconstruct  $A$  from  $\Phi(A)$ :  $X(1) = Y(1) = 0$  and

$$\begin{cases} X(i) = \alpha_L^{-1}(i) \text{ and } Y(i) = Y(\text{parent}(i)) & i \text{ left child} \\ Y(j) = \alpha_R^{-1}(j) \text{ and } X(j) = X(\text{parent}(j)) & j \text{ right child} \end{cases}$$

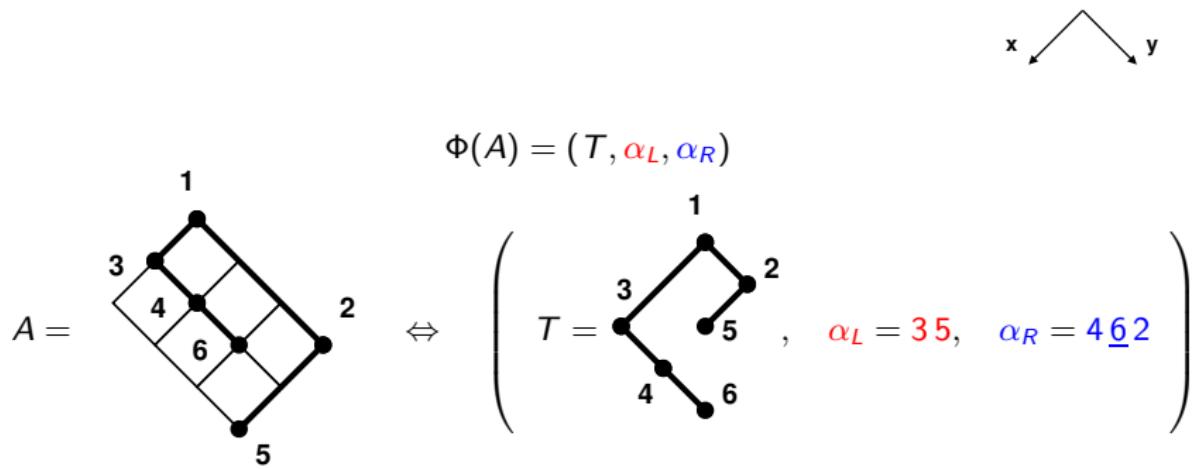
# A triplet representing a non-ambiguous tree



We can reconstruct  $A$  from  $\Phi(A)$ :  $X(1) = Y(1) = 0$  and

$$\begin{cases} X(i) = \alpha_L^{-1}(i) \text{ and } Y(i) = Y(\text{parent}(i)) & i \text{ left child} \\ Y(j) = \alpha_R^{-1}(j) \text{ and } X(j) = X(\text{parent}(j)) & j \text{ right child} \end{cases}$$

# A triplet representing a non-ambiguous tree



We can reconstruct  $A$  from  $\Phi(A)$ :  $X(1) = Y(1) = 0$  and

$$\begin{cases} X(i) = \alpha_L^{-1}(i) \text{ and } Y(i) = Y(\text{parent}(i)) & i \text{ left child} \\ Y(j) = \alpha_R^{-1}(j) \text{ and } X(j) = X(\text{parent}(j)) & j \text{ right child} \end{cases}$$

# A triplet representing a non ambiguous tree

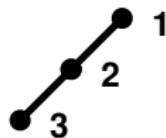
We can reconstruct  $A$  from  $\Phi(A)$ , hence  $\Phi$  is injective.

**Question:** what is the image of  $\Phi$ ? Are there some constraints on  $\alpha_L$  and  $\alpha_R$ ?

# A triplet representing a non ambiguous tree

We can reconstruct  $A$  from  $\Phi(A)$ , hence  $\Phi$  is injective.

**Question:** what is the image of  $\Phi$ ? Are there some constraints on  $\alpha_L$  and  $\alpha_R$ ?



Here we must have  $\alpha_L = 23$

# A triplet representing a non ambiguous tree

We can reconstruct  $A$  from  $\Phi(A)$ , hence  $\Phi$  is injective.

**Question:** what is the image of  $\Phi$ ? Are there some constraints on  $\alpha_L$  and  $\alpha_R$ ?



Here we must have  $\alpha_L = 23$

Actually, we have some constraints on  $\alpha_L$  and  $\alpha_R$

- $\alpha_L$  must be a linear extension of a particular poset defined on  $V_L$
- $\alpha_R$  must be a linear extension of a particular poset defined on  $V_R$

# The posets defined on $V_L$ and $V_R$

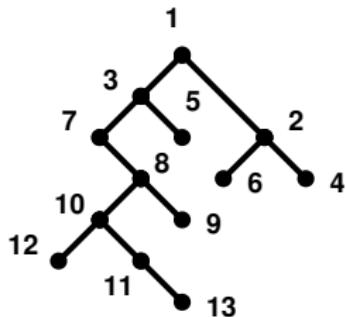
The order relation  $<_{V_L}$

Given  $i, k \in V_L$

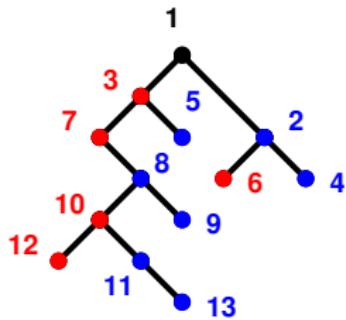


$i <_{V_L} k \iff i \text{ is an ancestor of } k.$

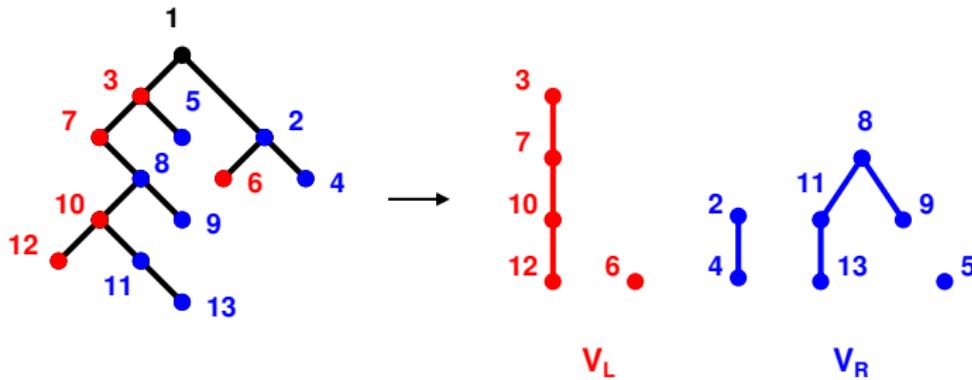
# The posets defined on $V_L$ and $V_R$



# The posets defined on $V_L$ and $V_R$



# The posets defined on $V_L$ and $V_R$



**Drawing convention:** In these Hasse diagrams, the minimal elements are the topmost ones.

## The constraints on $\alpha_L$ and $\alpha_R$

If  $i$  is an ancestor of  $k$ , and we must also have  $X(i) < X(k)$ . This means that  $i$  appears to the left of  $k$  in  $\alpha_L$ .

# The constraints on $\alpha_L$ and $\alpha_R$

If  $i$  is an ancestor of  $k$ , and we must also have  $X(i) < X(k)$ . This means that  $i$  appears to the left of  $k$  in  $\alpha_L$ .

## Constraints

$\alpha_L$  must be a linear extension of the poset  $(V_L, <_{V_L}) \rightarrow \alpha_L \in \mathcal{L}(V_L)$

$\alpha_R$  must be a linear extension of the poset  $(V_R, <_{V_R}) \rightarrow \alpha_R \in \mathcal{L}(V_R)$

# The constraints on $\alpha_L$ and $\alpha_R$

If  $i$  is an ancestor of  $k$ , and we must also have  $X(i) < X(k)$ . This means that  $i$  appears to the left of  $k$  in  $\alpha_L$ .

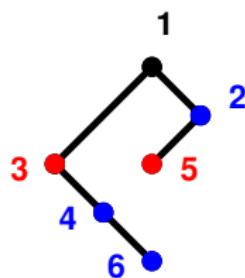
## Constraints

$\alpha_L$  must be a linear extension of the poset  $(V_L, <_{V_L}) \rightarrow \alpha_L \in \mathcal{L}(V_L)$

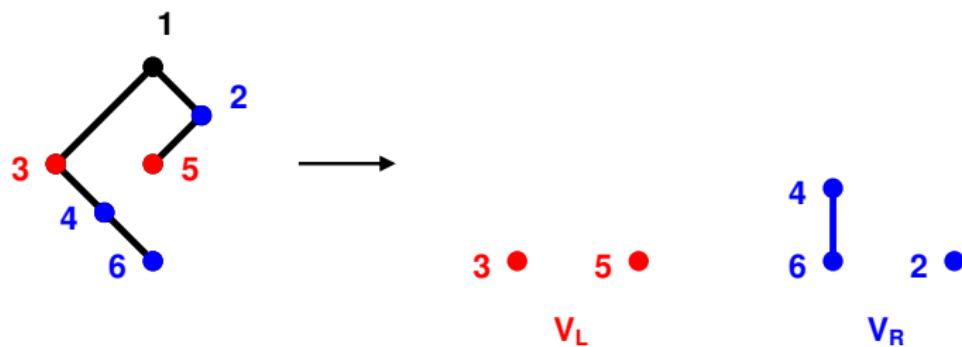
$\alpha_R$  must be a linear extension of the poset  $(V_R, <_{V_R}) \rightarrow \alpha_R \in \mathcal{L}(V_R)$

We can show that these conditions are also sufficient.

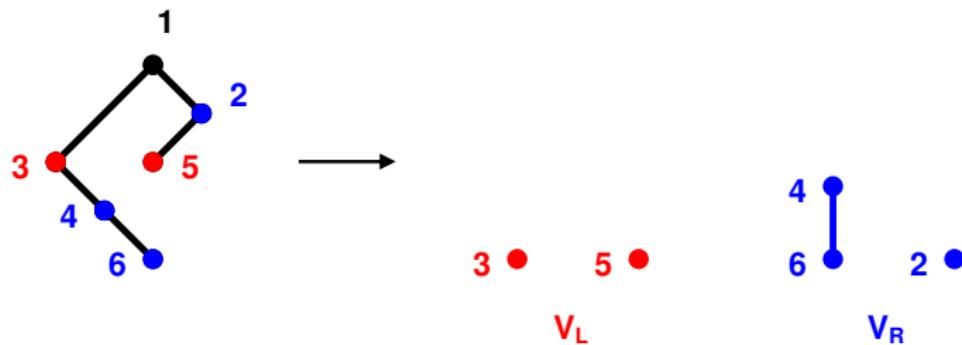
# The linear extensions of $V_L$ and $V_R$



# The linear extensions of $V_L$ and $V_R$

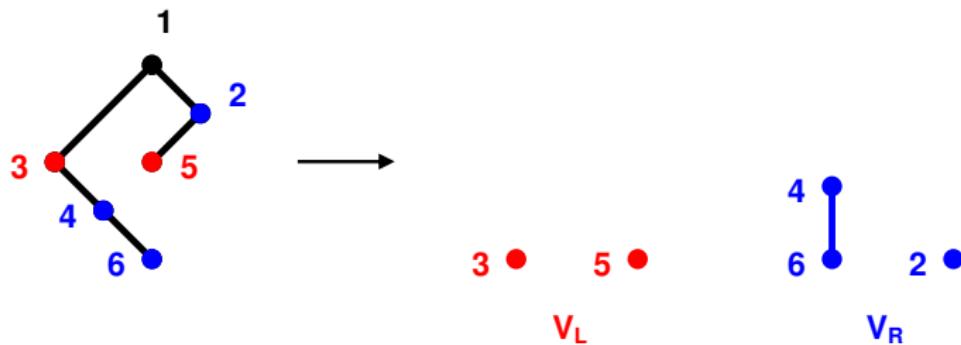


# The linear extensions of $V_L$ and $V_R$



$$\mathcal{L}(V_L) = \{35, 53\}$$

# The linear extensions of $V_L$ and $V_R$



$$\mathcal{L}(V_L) = \{35, 53\} \quad \mathcal{L}(V_R) = \{246, 426, 462\}$$

# The linear extensions of $V_L$ and $V_R$

$$\mathcal{L}(V_L) \times \mathcal{L}(V_R) = \{(35, 246), (35, 426), (35, 462), (53, 246), (53, 426), (53, 462)\}$$

# The linear extensions of $V_L$ and $V_R$

$$\mathcal{L}(V_L) \times \mathcal{L}(V_R) = \{(35, 246), (35, 426), (35, 462), (53, 246), (53, 426), (53, 462)\}$$

(35,246)

(35,426)

(35,462)

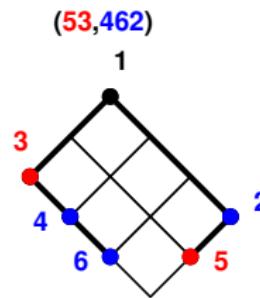
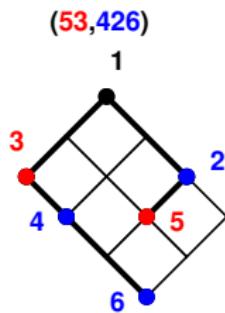
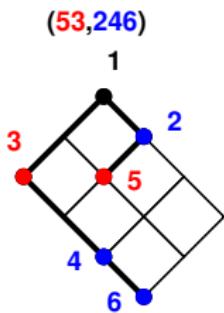
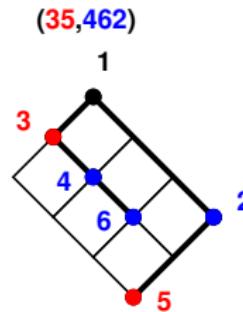
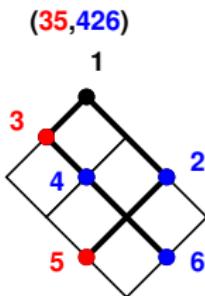
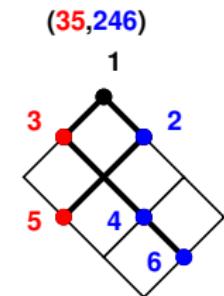
(53,246)

(53,426)

(53,462)

# The linear extensions of $V_L$ and $V_R$

$$\mathcal{L}(V_L) \times \mathcal{L}(V_R) = \{(35, 246), (35, 426), (35, 462), (53, 246), (53, 426), (53, 462)\}$$



# Enumerative consequences

## Theorem

$$NA(T) = |\mathcal{L}(V_L)| \times |\mathcal{L}(V_R)|$$

# Enumerative consequences

Theorem

$$NA(T) = |\mathcal{L}(V_L)| \times |\mathcal{L}(V_R)|$$

Theorem

The Hasse diagrams of  $(V_L, <_{V_L})$  and  $(V_R, <_{V_R})$  are forests.

# Enumerative consequences

## Theorem

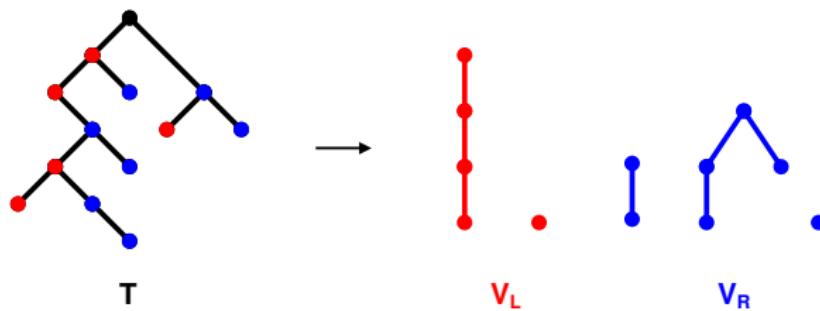
$$NA(T) = |\mathcal{L}(V_L)| \times |\mathcal{L}(V_R)|$$

## Theorem

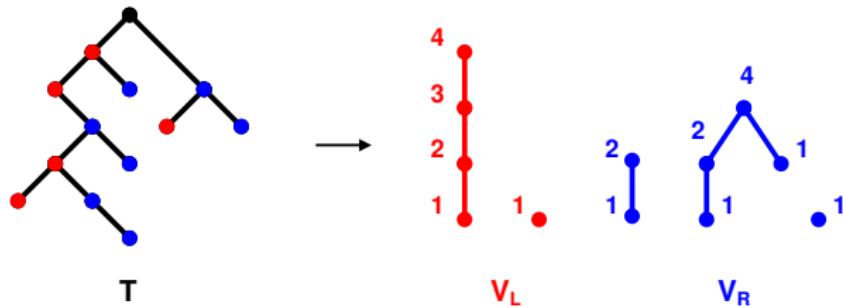
The Hasse diagrams of  $(V_L, <_{V_L})$  and  $(V_R, <_{V_R})$  are forests.

We can use Knuth's hook formula for forests to compute the value of  $NA(T)$ .

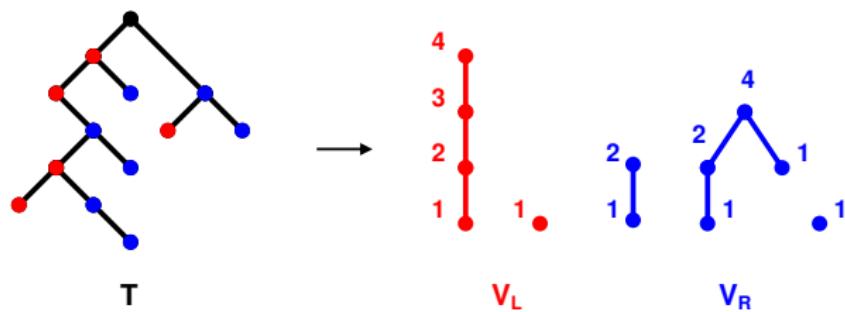
# A hook formula for $NA(T)$



# A hook formula for $NA(T)$

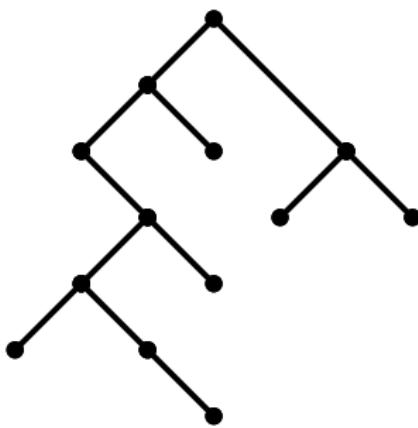


# A hook formula for $NA(T)$

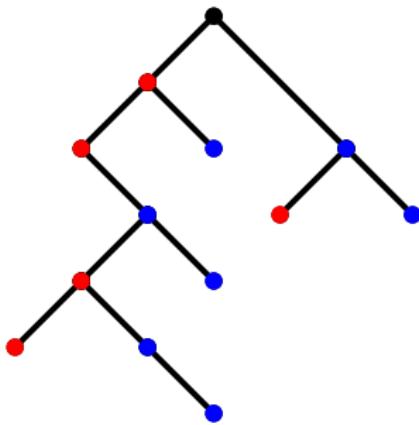


$$NA(T) = \frac{5!}{4 \cdot 3 \cdot 2 \cdot 1 \cdot 1} \cdot \frac{7!}{4 \cdot 2 \cdot 2 \cdot 1 \cdot 1 \cdot 1 \cdot 1} = 5 \cdot 315 = 1575$$

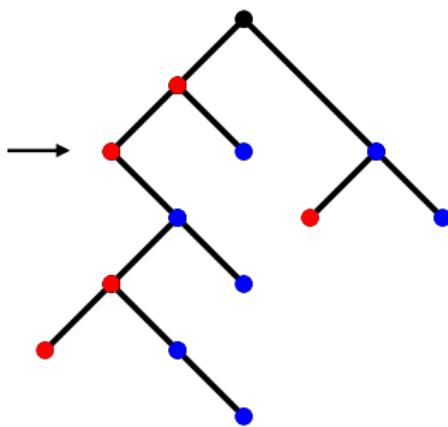
# A hook formula for $NA(T)$



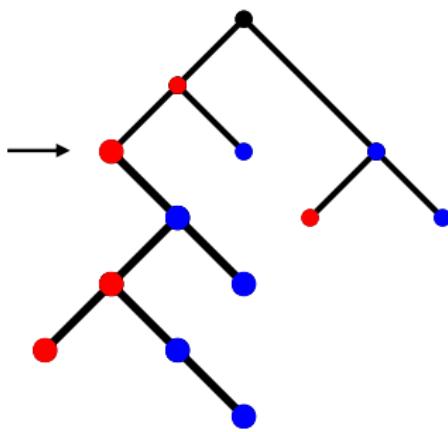
# A hook formula for $NA(T)$



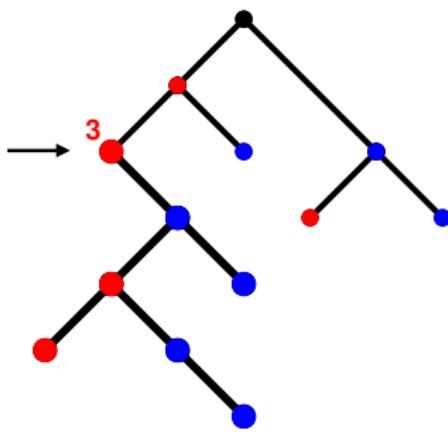
# A hook formula for $NA(T)$



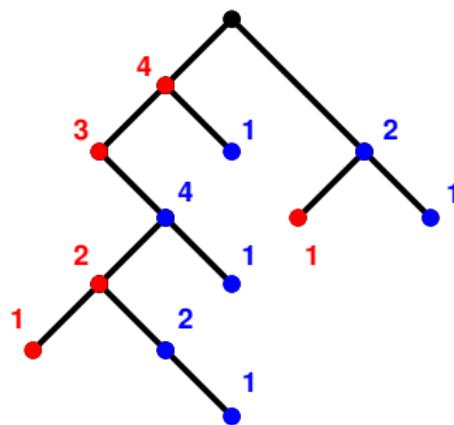
# A hook formula for $NA(T)$



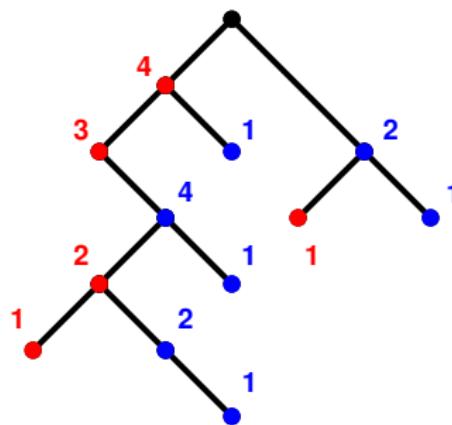
# A hook formula for $NA(T)$



# A hook formula for $NA(T)$



# A hook formula for $NA(T)$



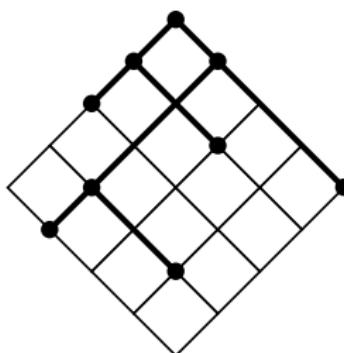
$$NA(T) = \frac{|V_L|!}{\prod_{e \in V_L} \lambda(e)} \cdot \frac{|V_R|!}{\prod_{e \in V_R} \lambda(e)} = \frac{5!}{4 \cdot 3 \cdot 2 \cdot 1 \cdot 1} \cdot \frac{7!}{4 \cdot 2 \cdot 2 \cdot 1 \cdot 1 \cdot 1 \cdot 1} = 1575$$

# Outline of the talk

- 1 Enumeration of non-ambiguous trees with a given underlying binary tree
- 2 **Complete non-ambiguous trees and the Bessel function**
- 3 A bijection between parallelogram polyominoes and binary trees

# Complete non-ambiguous trees

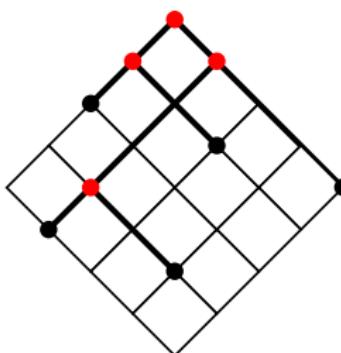
A complete non-ambiguous tree is a non-ambiguous tree whose underlying binary tree is complete (i.e. every vertex has 0 or 2 children)



size =

# Complete non-ambiguous trees

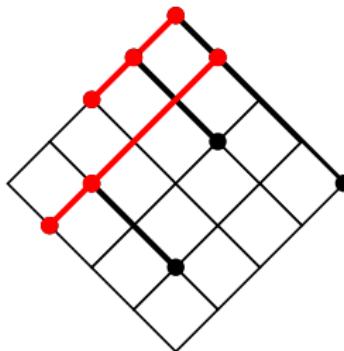
A complete non-ambiguous tree is a non-ambiguous tree whose underlying binary tree is complete (i.e. every vertex has 0 or 2 children)



$\text{size} = \# \text{ internal vertices}$

# Complete non-ambiguous trees

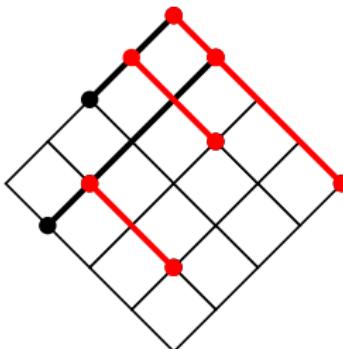
A complete non-ambiguous tree is a non-ambiguous tree whose underlying binary tree is complete (i.e. every vertex has 0 or 2 children)



$$\text{size} = \# \text{ internal vertices} = \# \text{ left edges}$$

# Complete non-ambiguous trees

A complete non-ambiguous tree is a non-ambiguous tree whose underlying binary tree is complete (i.e. every vertex has 0 or 2 children)



$$\text{size} = \# \text{ internal vertices} = \# \text{ left edges} = \# \text{ right edges}$$

# The number of complete non-ambiguous trees

$b_n = \#$  complete non-ambiguous trees of size  $n$

The sequence  $(b_k)_{k \geq 0}$  appears on the OEIS:

$$\text{A002190} = (b_n)_{n \geq 0} = (1, 1, 4, 33, 456, 9460, \dots)$$

with no combinatorial interpretation.

# The number of complete non-ambiguous trees

$b_n = \#$  complete non-ambiguous trees of size  $n$

The sequence  $(b_k)_{k \geq 0}$  appears on the OEIS:

$$\text{A002190} = (b_n)_{n \geq 0} = (1, 1, 4, 33, 456, 9460, \dots)$$

with no combinatorial interpretation.

## Theorem

$$-\ln(J_0(x)) = \sum_{k \geq 0} b_k \frac{x^{2(k+1)}}{((k+1)!2^{k+1})^2}$$

where  $J_0$  is the Bessel function of order 0

$$J_0(x) = \sum_{i \geq 0} \frac{(-1)^i}{(i!)^2} \left(\frac{x}{2}\right)^{2i}$$

# The number of complete non-ambiguous trees

**Proof** The Bessel function of order 0 is the solution of the differential equation:

$$\frac{d^2y}{dx^2} + \frac{1}{x} \frac{dy}{dx} + y = 0,$$

such that  $y(0) = 1$  and  $y'(0) = 0$ .

# The number of complete non-ambiguous trees

**Proof** The Bessel function of order 0 is the solution of the differential equation:

$$\frac{d^2y}{dx^2} + \frac{1}{x} \frac{dy}{dx} + y = 0,$$

such that  $y(0) = 1$  and  $y'(0) = 0$ . We want to check that the function

$$y(x) = \exp \left( - \sum_{k \geq 0} b_k \frac{x^{2(k+1)}}{((k+1)!2^{k+1})^2} \right).$$

solves this differential equation.

# The number of complete non-ambiguous trees

**Proof** The Bessel function of order 0 is the solution of the differential equation:

$$\frac{d^2y}{dx^2} + \frac{1}{x} \frac{dy}{dx} + y = 0,$$

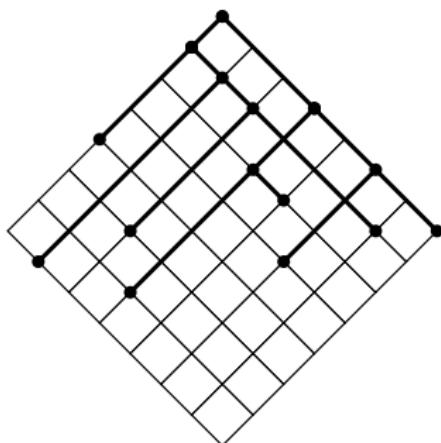
such that  $y(0) = 1$  and  $y'(0) = 0$ . We want to check that the function

$$y(x) = \exp \left( - \sum_{k \geq 0} b_k \frac{x^{2(k+1)}}{((k+1)!2^{k+1})^2} \right).$$

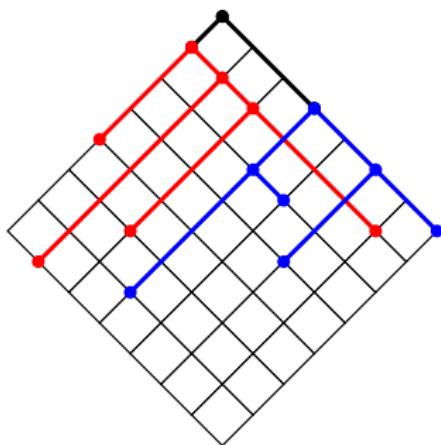
solves this differential equation. This is equivalent to show that the sequence  $b_k$  satisfies the Bessel differential equation if the following recurrence for the sequence  $b_n$

$$b_{n+1} = \sum_{u+v=n} \binom{n+1}{u} \binom{n+1}{v} b_u b_v$$

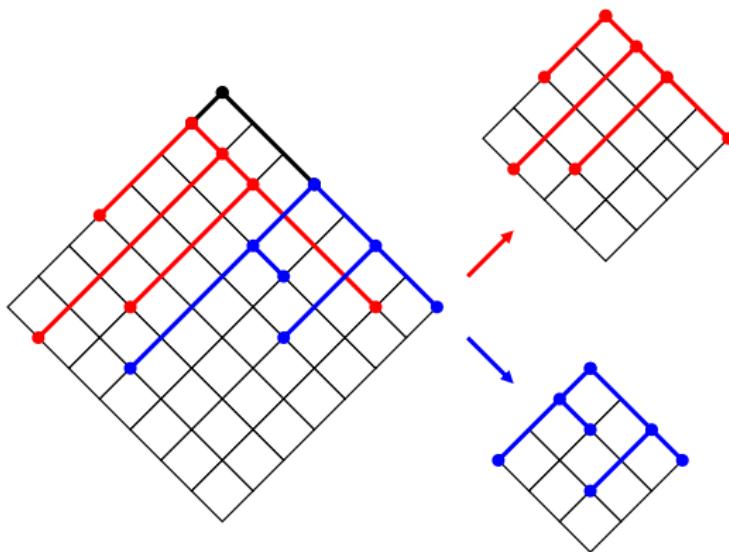
# Recursive enumeration of complete non-ambiguous trees



# Recursive enumeration of complete non-ambiguous trees

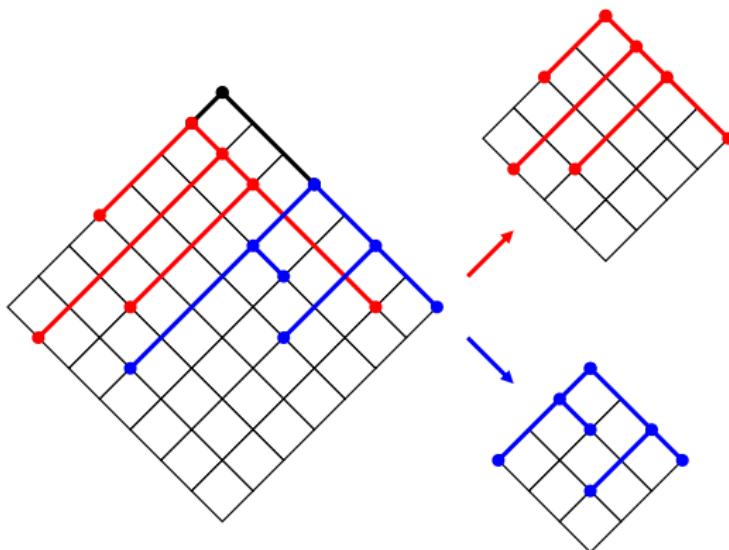


# Recursive enumeration of complete non-ambiguous trees



$$b_{n+1} = \sum_{u+v=n} \binom{n+1}{u} \binom{n+1}{v} b_u b_v$$

# Recursive enumeration of complete non-ambiguous trees



$$b_{n+1} = \sum_{u+v=n} \binom{n+1}{u} \binom{n+1}{v} b_u b_v \quad \blacksquare$$

# Another identity involving complete non-ambiguous trees

In 1963, Carlitz gave the following identity satisfied by the sequence  $b_n$  (without proof):

$$\sum_{k=0}^{n-1} (-1)^k \binom{n}{k+1} \binom{n-1}{k} b_k = 1$$

# Another identity involving complete non-ambiguous trees

In 1963, Carlitz gave the following identity satisfied by the sequence  $b_n$  (without proof):

$$\sum_{k=0}^{n-1} (-1)^k \binom{n}{k+1} \binom{n-1}{k} b_k = 1$$

We use a slightly modified version of complete non-ambiguous trees to prove it.

# Another identity involving complete non-ambiguous trees

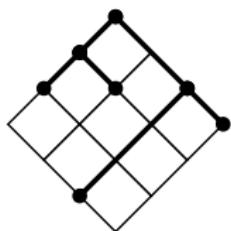
In 1963, Carlitz gave the following identity satisfied by the sequence  $b_n$  (without proof):

$$\sum_{k=0}^{n-1} (-1)^k \binom{n}{k+1} \binom{n-1}{k} b_k = 1$$

We use a slightly modified version of complete non-ambiguous trees to prove it.

# Gridded trees

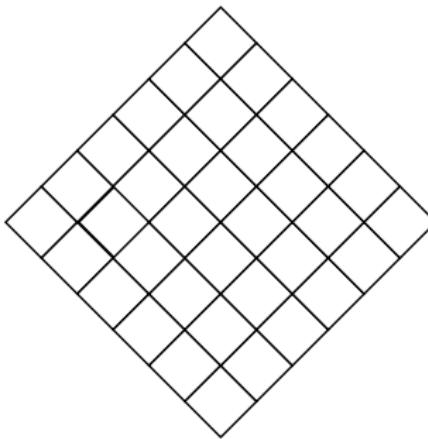
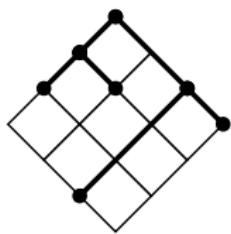
A gridded tree of size  $(k, n)$  is a complete non-ambiguous tree of size  $k$  embedded in a  $n \times n$  grid:



A complete non-ambiguous tree of size 3

# Gridded trees

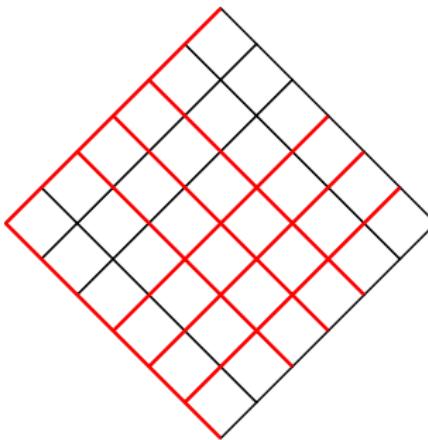
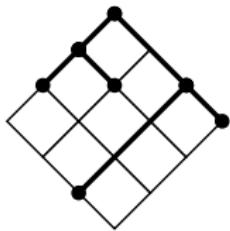
A gridded tree of size  $(k, n)$  is a complete non-ambiguous tree of size  $k$  embedded in a  $n \times n$  grid:



A complete non-ambiguous tree of size 3

# Gridded trees

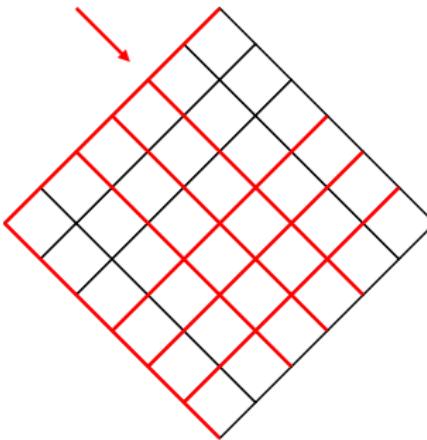
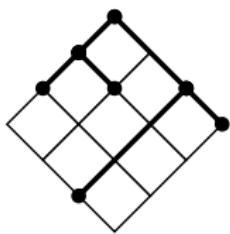
A gridded tree of size  $(k, n)$  is a complete non-ambiguous tree of size  $k$  embedded in a  $n \times n$  grid:



A complete non-ambiguous tree of size 3

# Gridded trees

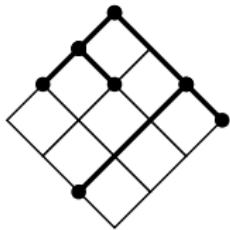
A gridded tree of size  $(k, n)$  is a complete non-ambiguous tree of size  $k$  embedded in a  $n \times n$  grid:



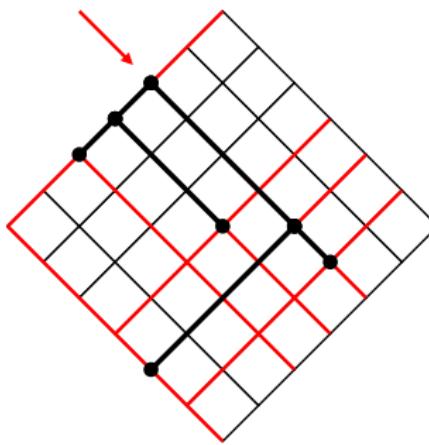
A complete non-ambiguous tree of size 3

# Gridded trees

A gridded tree of size  $(k, n)$  is a complete non-ambiguous tree of size  $k$  embedded in a  $n \times n$  grid:



A complete non-ambiguous tree of size 3



A gridded tree of size  $(3, 7)$

# Gridded trees

The number of gridded trees of size  $(k, n)$  is

$$g_{k,n} = \binom{n}{k+1} \binom{n-1}{k} b_k$$

The identity that we want to prove becomes

# Gridded trees

The number of gridded trees of size  $(k, n)$  is

$$g_{k,n} = \binom{n}{k+1} \binom{n-1}{k} b_k$$

The identity that we want to prove becomes

$$\sum_{k=0}^{n-1} (-1)^k \binom{n}{k+1} \binom{n-1}{k} b_k = \sum_{k=0}^{n-1} (-1)^k g_{k,n} = 1$$

# Gridded trees

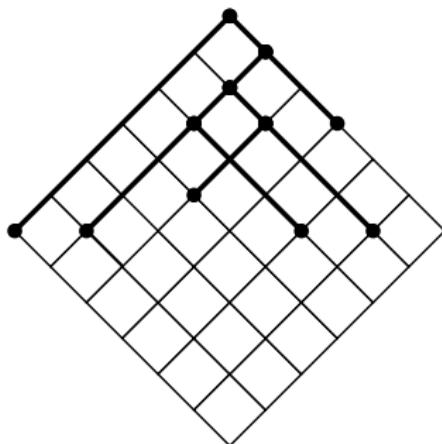
The number of gridded trees of size  $(k, n)$  is

$$g_{k,n} = \binom{n}{k+1} \binom{n-1}{k} b_k$$

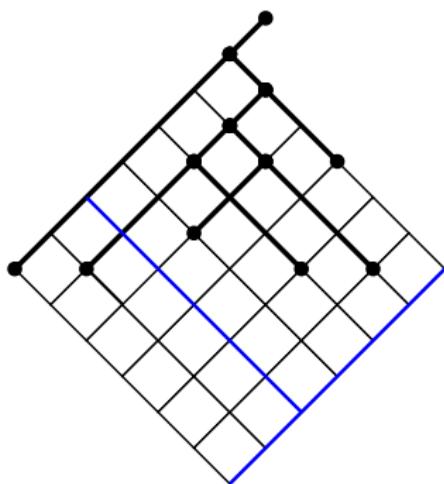
The identity that we want to prove becomes

$$\sum_{k=0}^{n-1} (-1)^k \binom{n}{k+1} \binom{n-1}{k} b_k = \sum_{k=0}^{n-1} (-1)^k g_{k,n} = 1 \equiv \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array}$$

# An involution on gridded trees

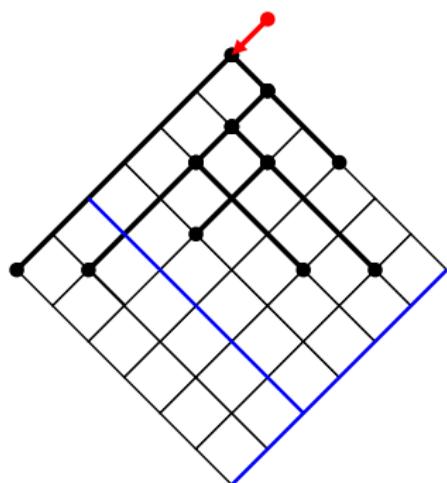


# An involution on gridded trees



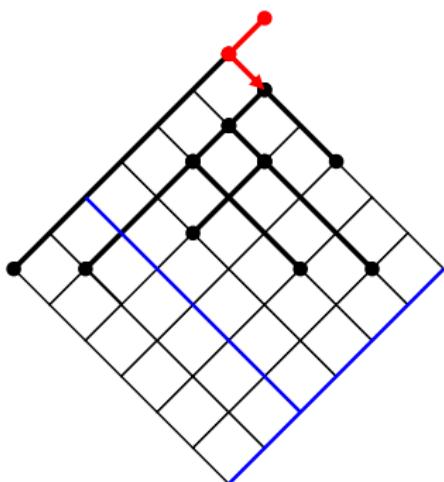
**Case 1:** the zig-zag path doesn't cross an empty row/column

# An involution on gridded trees



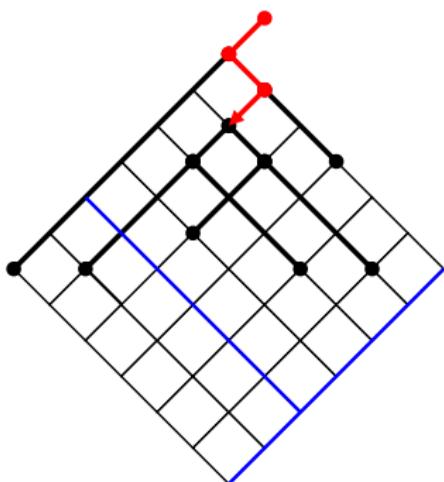
**Case 1:** the zig-zag path doesn't cross an empty row/column

# An involution on gridded trees



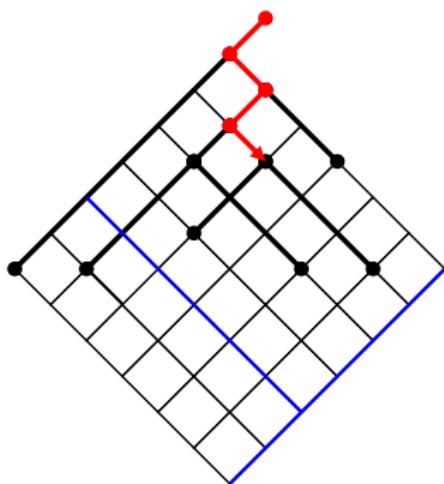
**Case 1:** the zig-zag path doesn't cross an empty row/column

# An involution on gridded trees



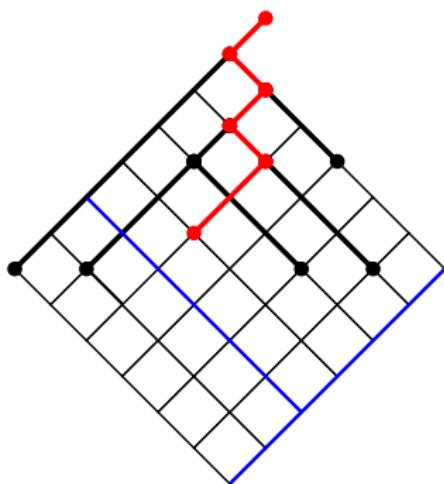
**Case 1:** the zig-zag path doesn't cross an empty row/column

# An involution on gridded trees



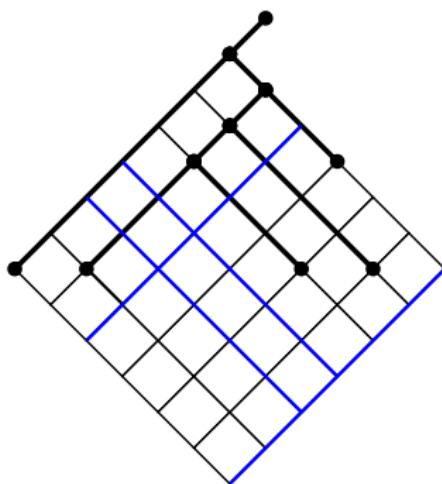
**Case 1:** the zig-zag path doesn't cross an empty row/column

# An involution on gridded trees



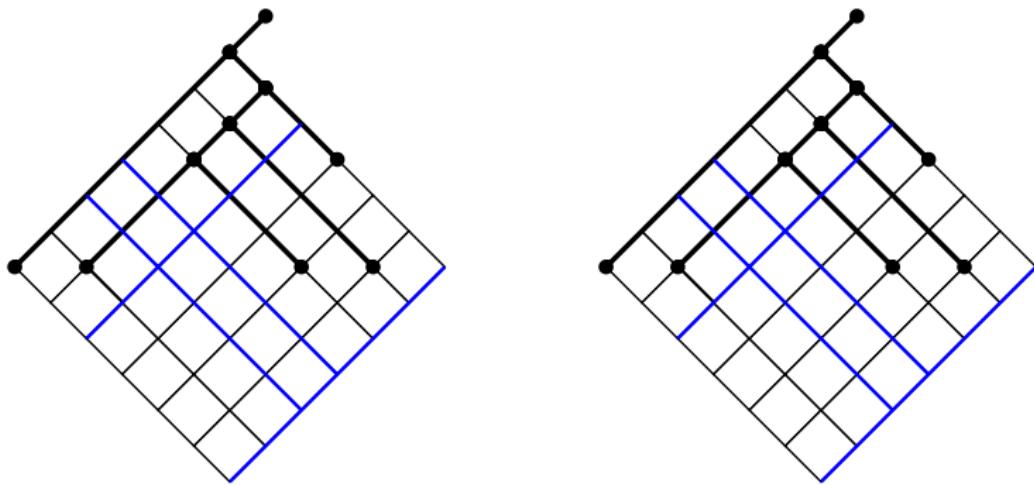
**Case 1:** the zig-zag path doesn't cross an empty row/column

# An involution on gridded trees



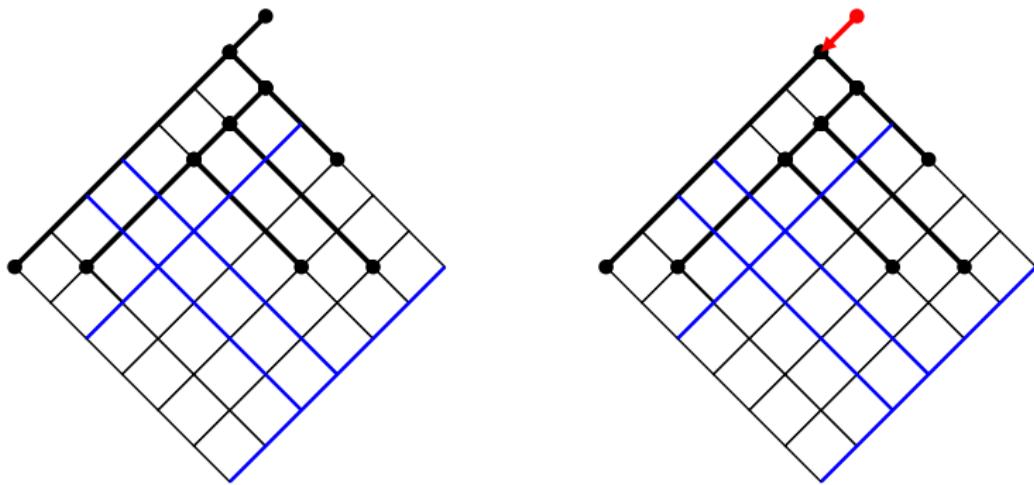
**Case 1:** the zig-zag path doesn't cross an empty row/column

# An involution on gridded trees



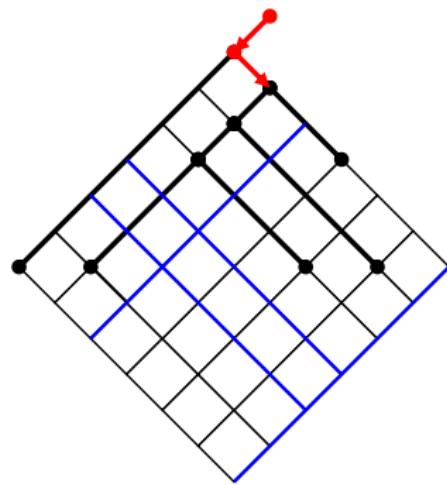
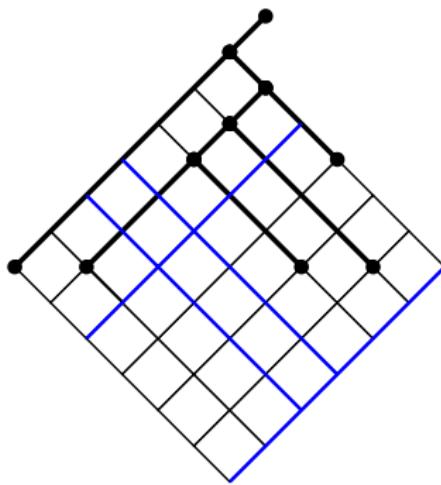
Case 2: the zig-zag path crosses an empty row/column

# An involution on gridded trees



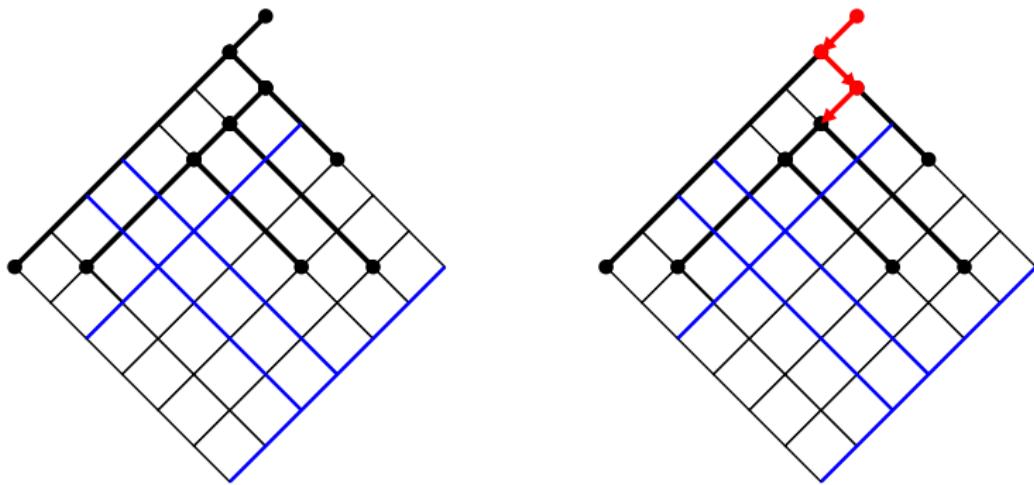
Case 2: the zig-zag path crosses an empty row/column

# An involution on gridded trees



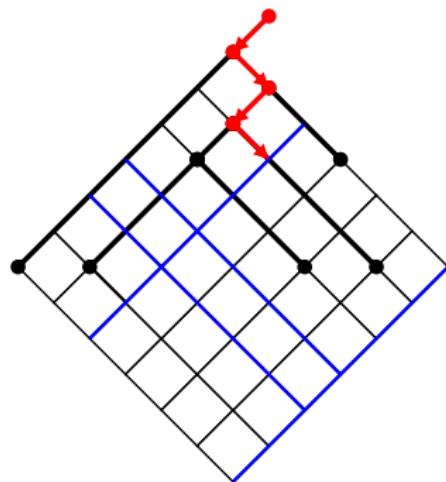
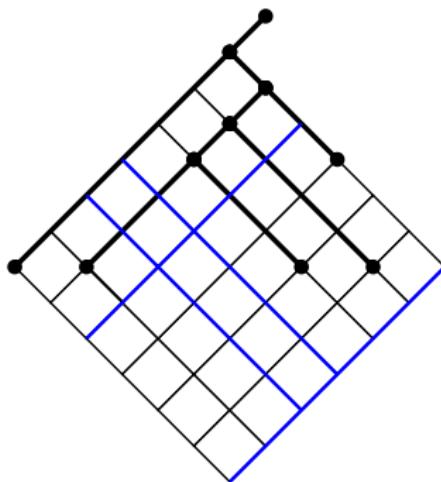
Case 2: the zig-zag path crosses an empty row/column

# An involution on gridded trees



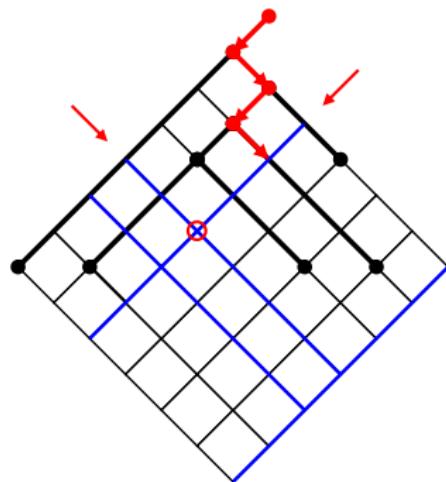
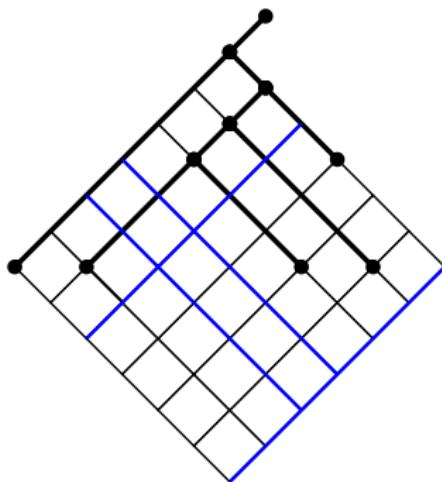
Case 2: the zig-zag path crosses an empty row/column

# An involution on gridded trees



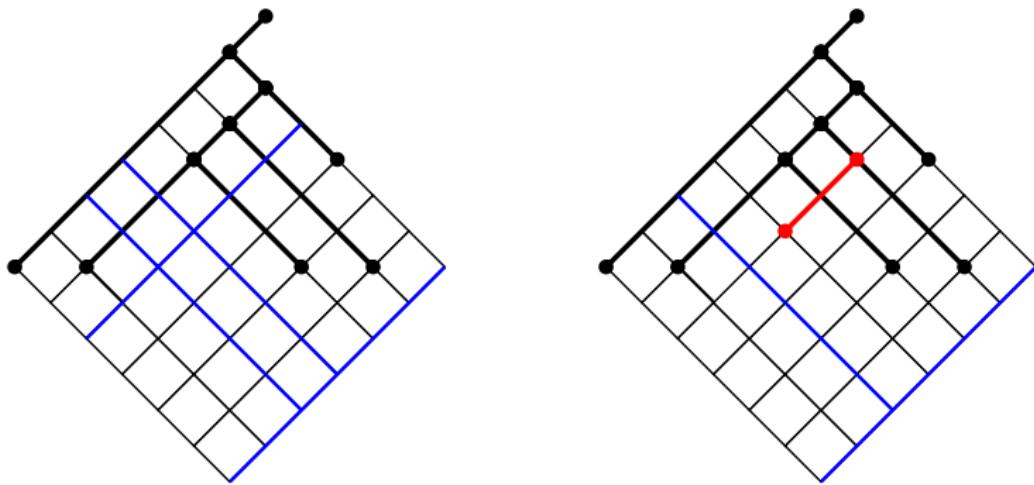
Case 2: the zig-zag path crosses an empty row/column

# An involution on gridded trees



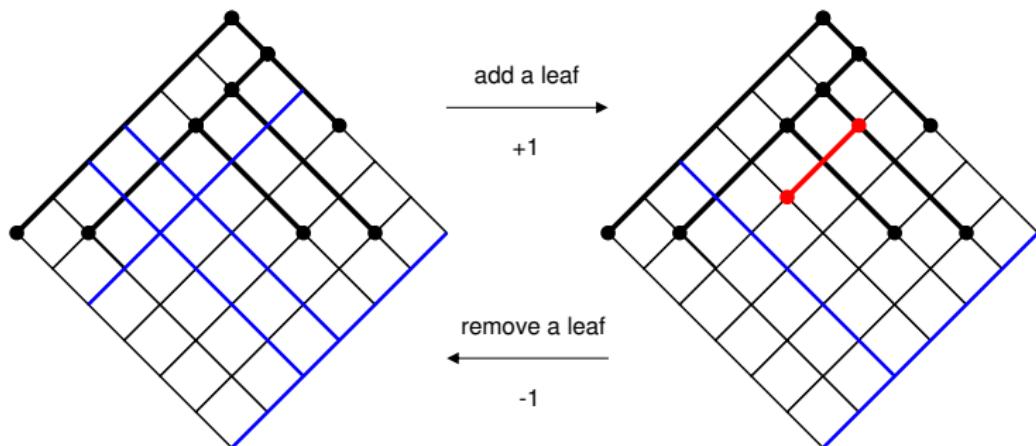
Case 2: the zig-zag path crosses an empty row/column

# An involution on gridded trees



Case 2: the zig-zag path crosses an empty row/column

# An involution on gridded trees



# A conjecture involving the sequence $b_k$

Conjecture [Paul Hanna, OEIS 2010]

$$\sum_{k=0}^n (-1)^k b_k C_k \binom{n+k}{n-k}^2 = 0$$

where  $C_k$  is the  $k$ -th Catalan number.

Complete non-ambiguous trees could be the right objects in order to prove this conjecture.

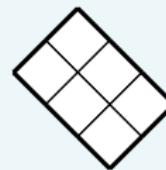
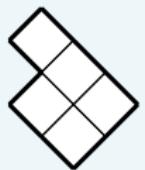
# Outline of the talk

- 1 Enumeration of non-ambiguous trees with a given underlying binary tree
- 2 Complete non-ambiguous trees and the Bessel function
- 3 A bijection between parallelogram polyominoes and binary trees

# Parallelogram polyominoes

A parallelogram polyomino of size  $n$  is a pair of lattice paths of lengths  $n + 1$  with south-west and south-east steps starting at the same point, ending at the same point and never meeting each other.

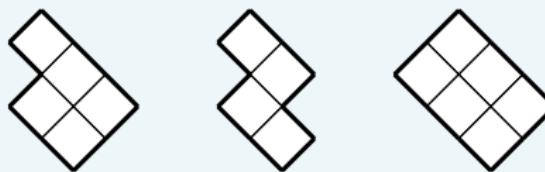
Some examples for  $n = 4$



# Parallelogram polyominoes

A parallelogram polyomino of size  $n$  is a pair of lattice paths of lengths  $n + 1$  with south-west and south-east steps starting at the same point, ending at the same point and never meeting each other.

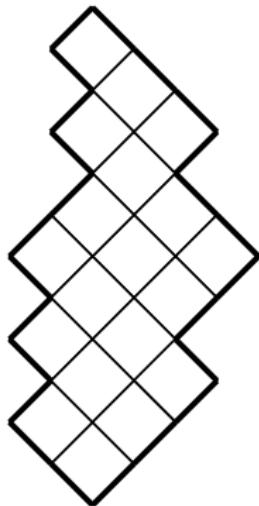
Some examples for  $n = 4$



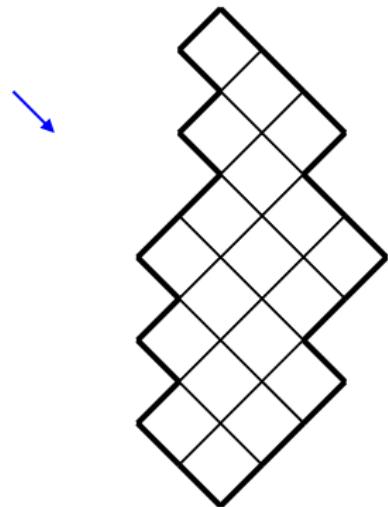
Theorem [Delest, Viennot, ...]

Parallelogram polyominoes of size  $n$  are in bijection with binary trees with  $n$  vertices.

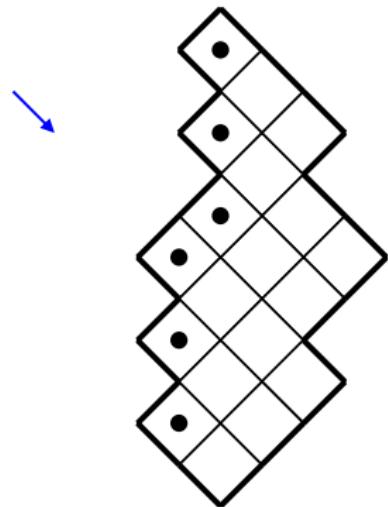
# A new bijection $S$ between PPs and binary trees



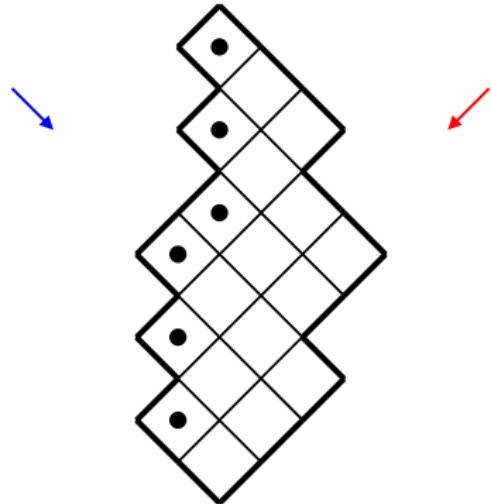
# A new bijection $S$ between PPs and binary trees



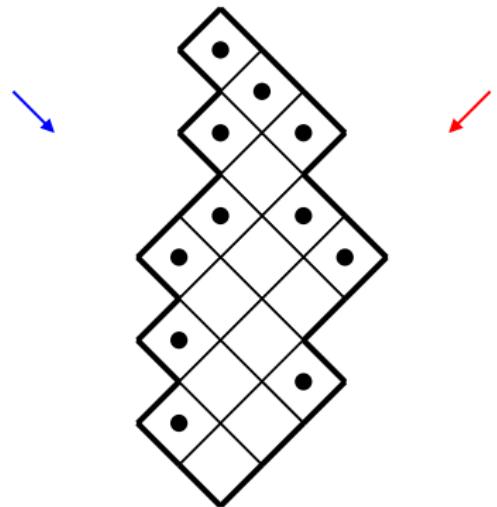
# A new bijection $S$ between PPs and binary trees



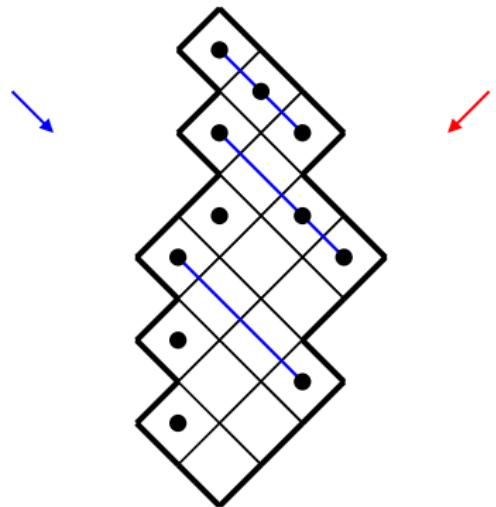
# A new bijection $S$ between PPs and binary trees



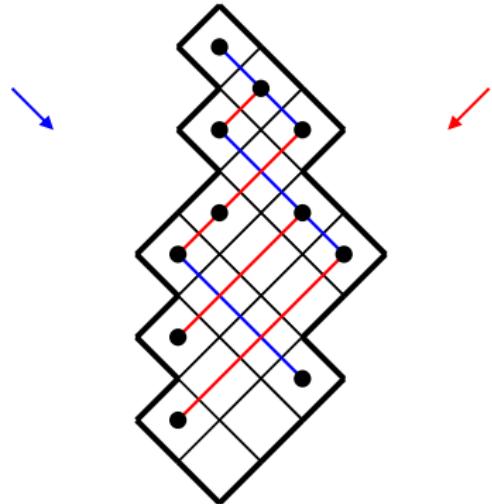
# A new bijection $S$ between PPs and binary trees



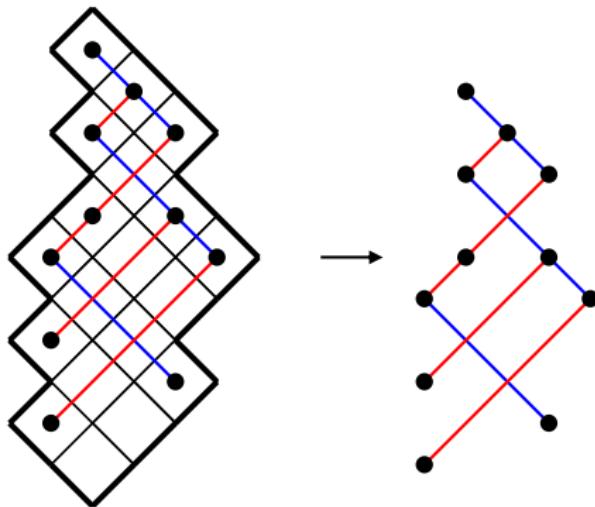
# A new bijection $S$ between PPs and binary trees



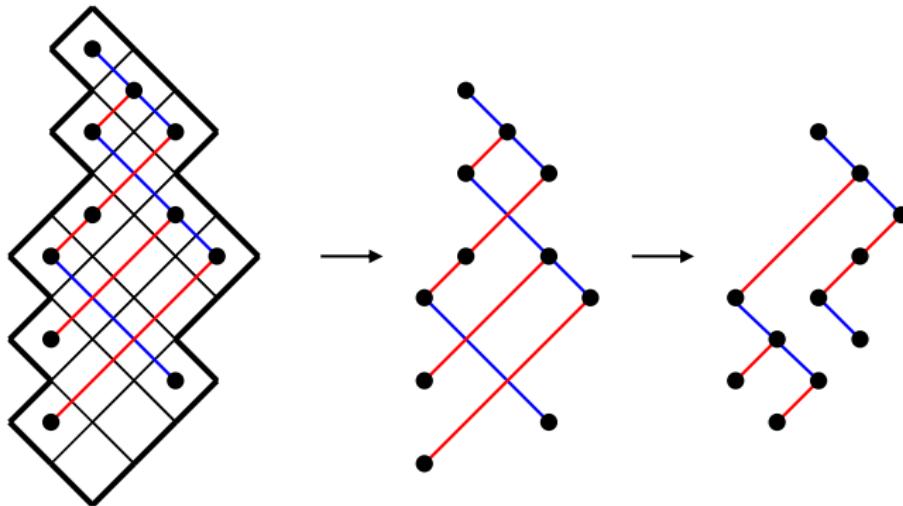
# A new bijection $S$ between PPs and binary trees



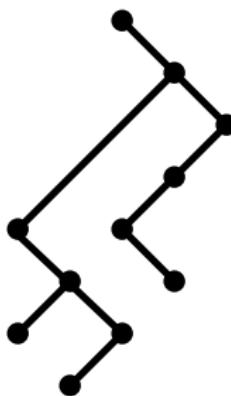
# A new bijection $S$ between PPs and binary trees



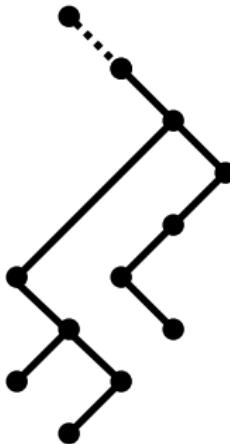
# A new bijection $S$ between PPs and binary trees



# The inverse of S: the turn labelling

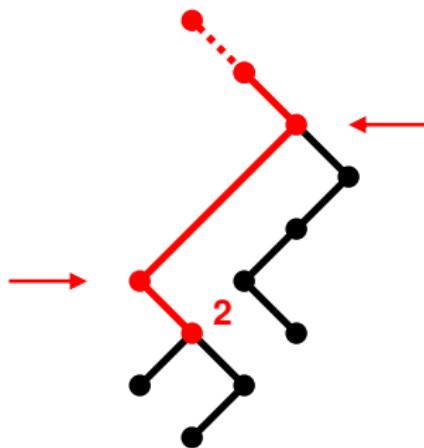


# The inverse of S: the turn labelling



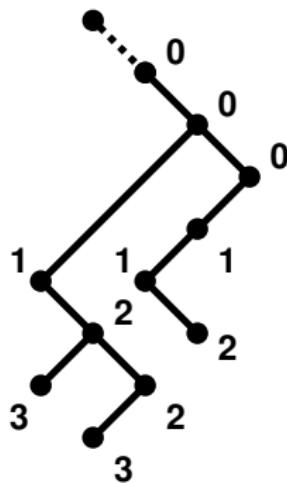
# The inverse of S: the turn labelling

$t(s) =$  the number of turns in the path from the virtual root to  $s$ .

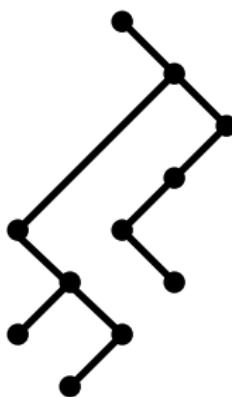


# The inverse of S: the turn labelling

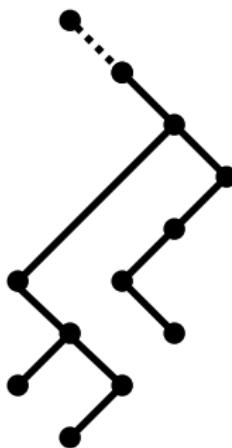
$t(s) = \text{the number of turns in the path from the virtual root to } s.$



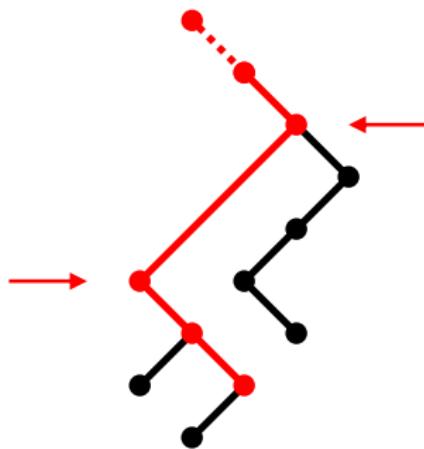
# The inverse of S: the zig-zag labelling



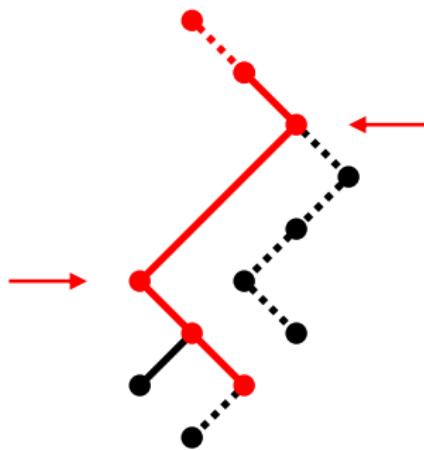
# The inverse of S: the zig-zag labelling



# The inverse of S: the zig-zag labelling

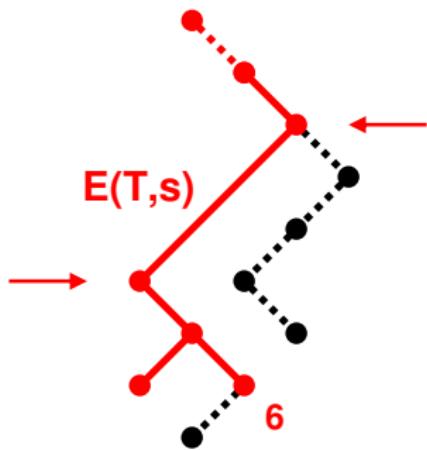


# The inverse of S: the zig-zag labelling



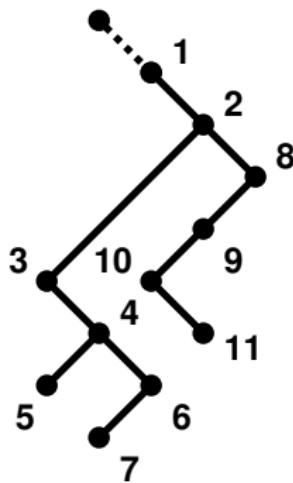
# The inverse of S: the zig-zag labelling

$$z(s) = |E(T, s)|.$$

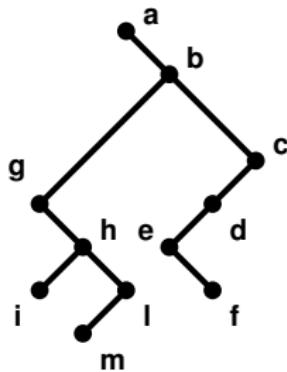


# The inverse of S: the zig-zag labelling

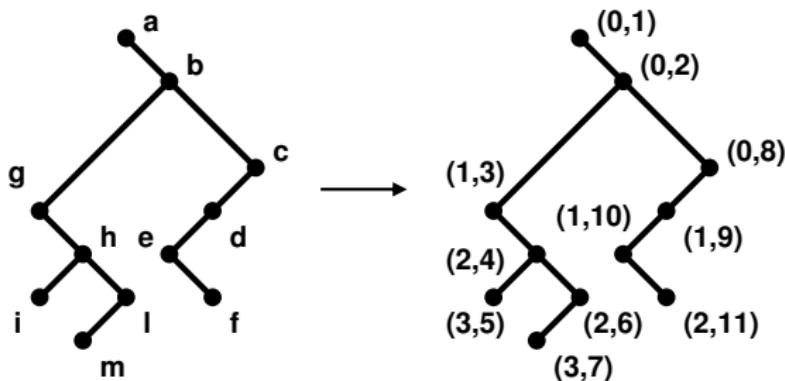
$$z(s) = |E(T, s)|.$$



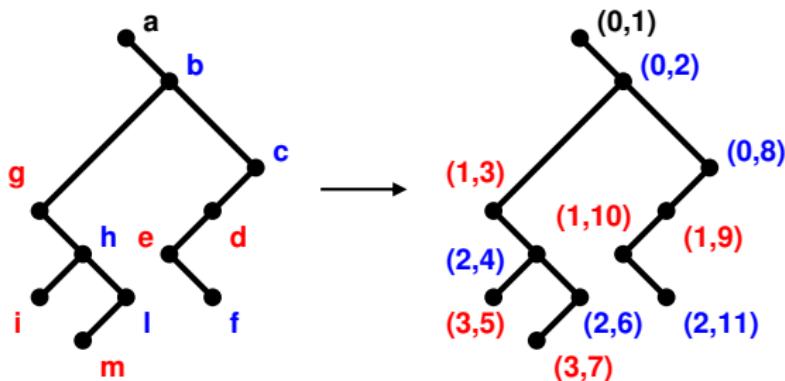
# The inverse of S: two labellings defined on a binary tree



# The inverse of S: two labellings defined on a binary tree



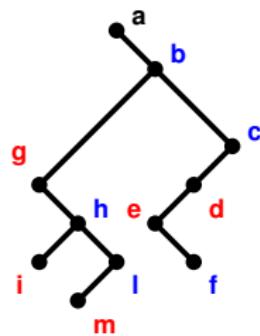
# The inverse of S: two labellings defined on a binary tree



$$\alpha_L = g \ d \ e \ i \ m$$

$$\alpha_R = b \ c \ h \ l \ f$$

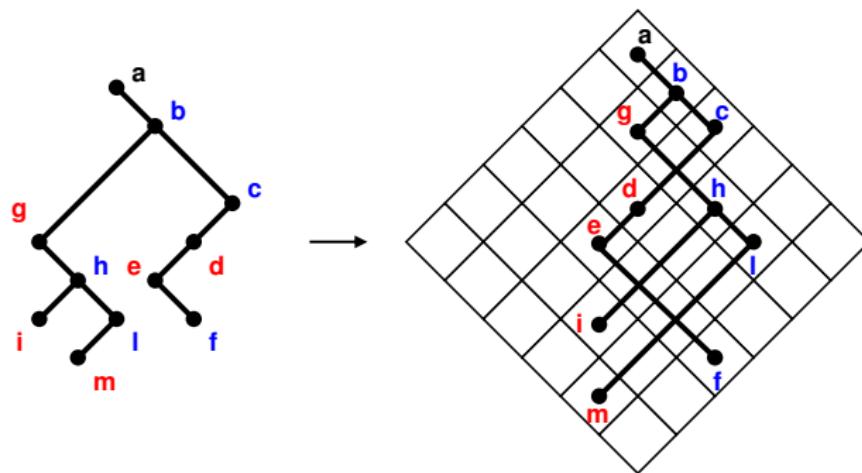
# From binary trees to PPs



$$\alpha_L = g \, d \, e \, i \, m$$

$$\alpha_R = b \, c \, h \, l \, f$$

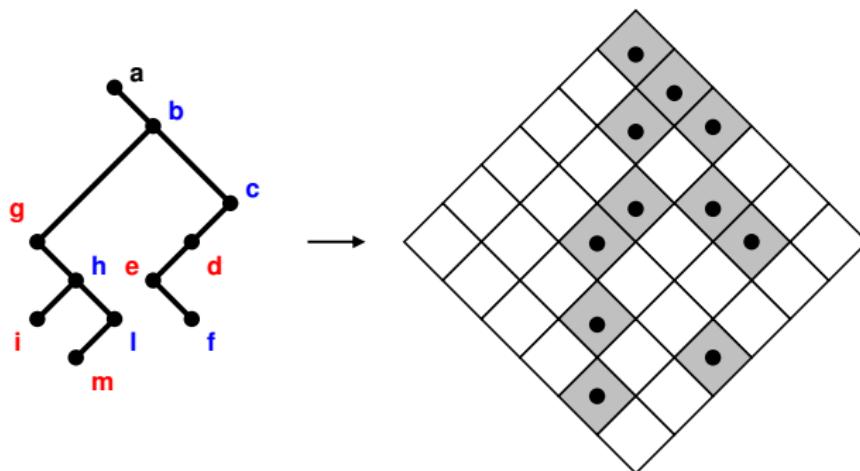
# From binary trees to PPs



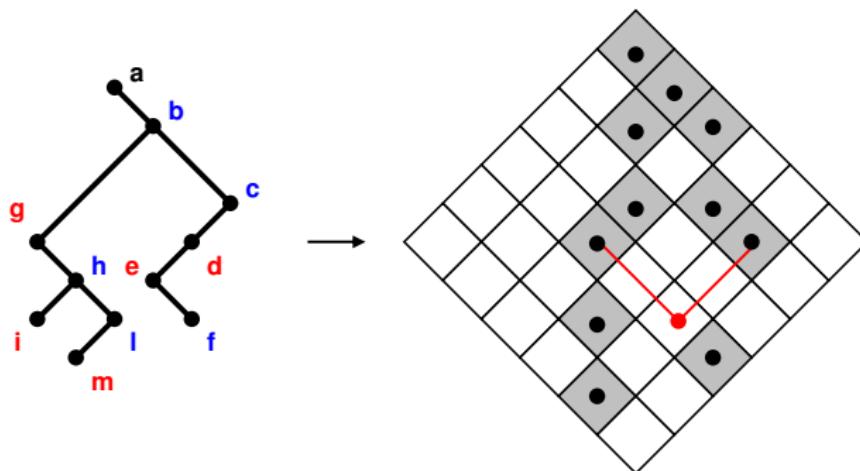
$$\alpha_L = g \, d \, e \, i \, m$$

$$\alpha_R = b \, c \, h \, l \, f$$

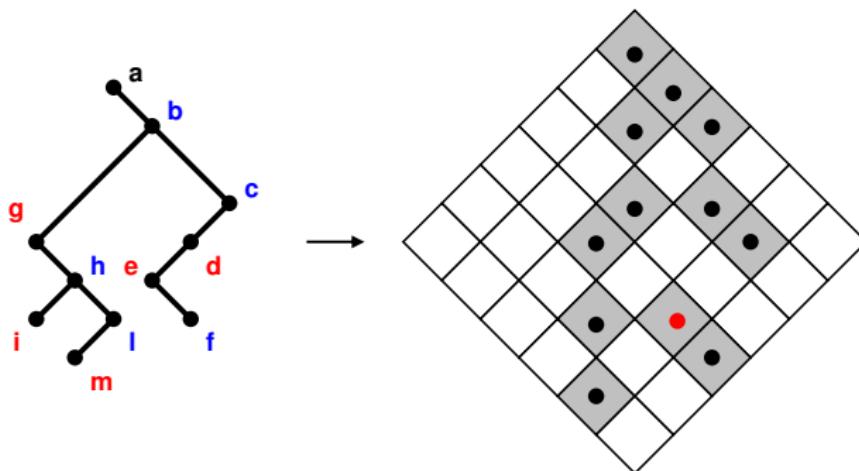
# From binary trees to PPs



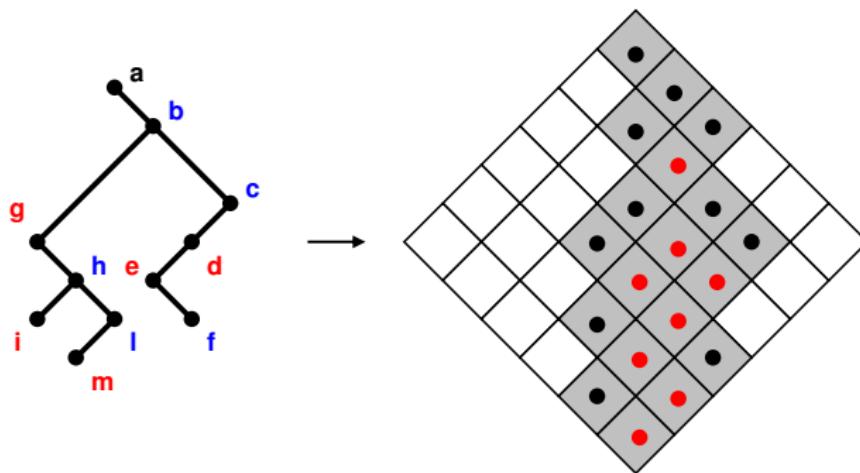
# From binary trees to PPs



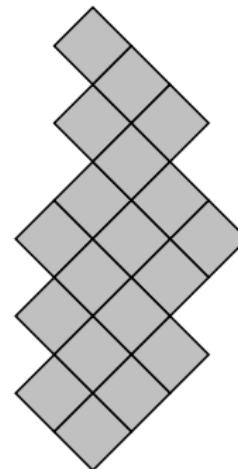
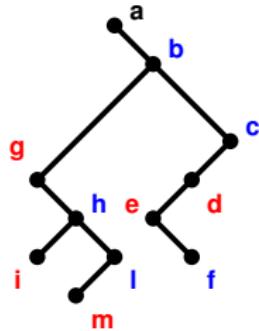
# From binary trees to PPs



# From binary trees to PPs



# From binary trees to PPs



# Some perspectives

- Find a combinatorial proof of Hanna's identity

$$\sum_{k=0}^n (-1)^k b_k C_k \binom{n+k}{n-k}^2 = 0$$

- Find an analogue of complete non-ambiguous trees enumerated by the coefficients in the expansion of  $-\ln(J_k(x))$  for other values of  $k$
- Define an analogue of non-ambiguous trees in higher dimensions

Merci beaucoup

Merci de votre attention !