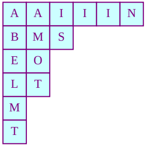
Combinatorics of non-ambiguous trees

Matteo Silimbani

(LaBRI - Bordeaux)

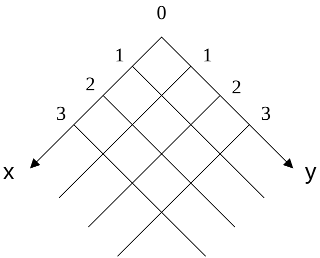


FPSAC 2013 - Université Sorbonne Nouvelle (Paris)

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

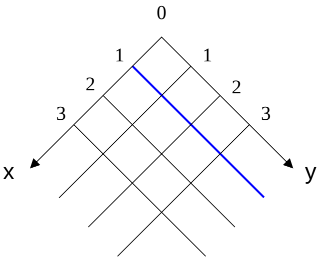
Some conventions

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



Some conventions

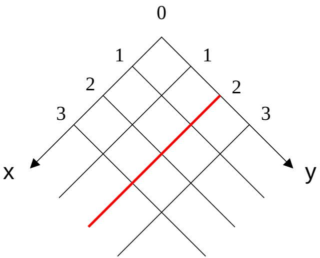
Objects will be drawn in a $\mathbb{N} \times \mathbb{N}$ grid.



y-oriented lines = rows

Some conventions

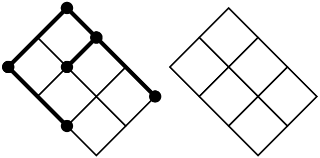
Objects will be drawn in a $\mathbb{N} \times \mathbb{N}$ grid.



y-oriented lines = rows
x-oriented lines = columns

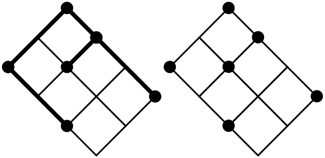
Intuitive definition of non ambiguous trees

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)



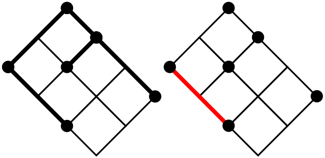
Intuitive definition of non ambiguous trees

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)



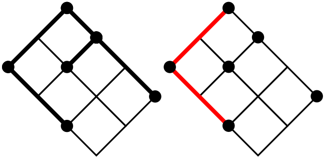
Intuitive definition of non ambiguous trees

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)



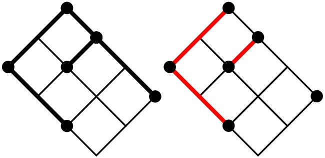
Intuitive definition of non ambiguous trees

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)



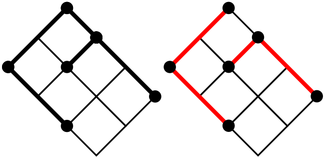
Intuitive definition of non ambiguous trees

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)



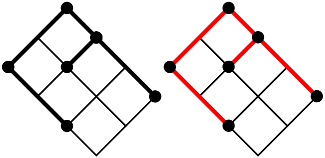
Intuitive definition of non ambiguous trees

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)



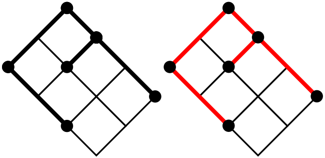
Intuitive definition of non ambiguous trees

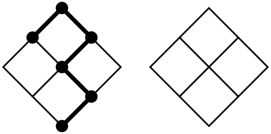
A non-ambiguous tree is a binary tree embedded in the grid in such a way that the embedding of its vertices in the grid determines the tree completely (i.e. determines its edges)



Intuitive definition of non ambiguous trees

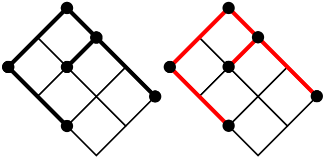
A non-ambiguous tree is a binary tree embedded in the grid in such a way that the embedding of its vertices in the grid determines the tree completely (i.e. determines its edges)

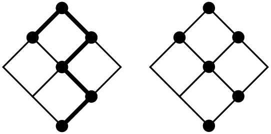




Intuitive definition of non ambiguous trees

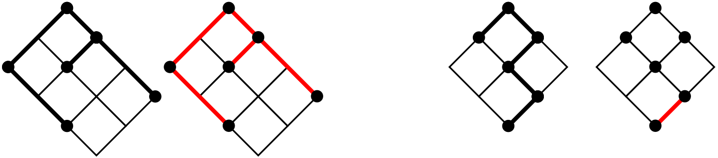
A non-ambiguous tree is a binary tree embedded in the grid in such a way that the embedding of its vertices in the grid determines the tree completely (i.e. determines its edges)





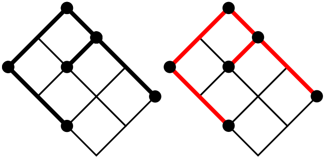
Intuitive definition of non ambiguous trees

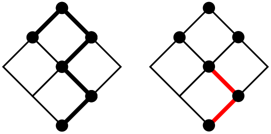
A non-ambiguous tree is a binary tree embedded in the grid in such a way that the embedding of its vertices in the grid determines the tree completely (i.e. determines its edges)



Intuitive definition of non ambiguous trees

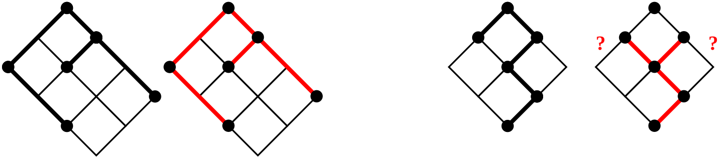
A non-ambiguous tree is a binary tree embedded in the grid in such a way that the embedding of its vertices in the grid determines the tree completely (i.e. determines its edges)





Intuitive definition of non ambiguous trees

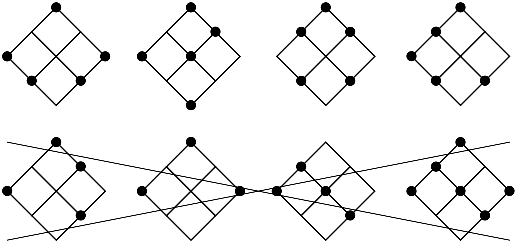
A non-ambiguous tree is a binary tree embedded in the grid in such a way that the embedding of its vertices in the grid determines the tree completely (i.e. determines its edges)



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:

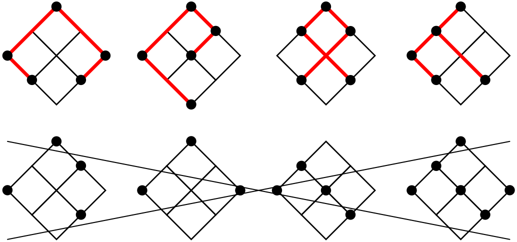
- $(0, 0) \in A$ (the root)
- every point (except the root) has a "parent"
- the pattern  is forbidden
- compactness: there is no empty row/column



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:

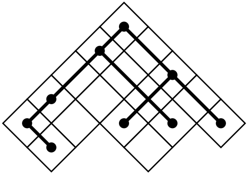
- the PASEP model in physics;
- $2-31$ pattern inside a permutation;
- excedences and cycles of a permutation;
- Laguerre polynomials;
- ...

[Burstein, Corteel, Dasse-Hartaut, Hitczenko, Josuat-Vergès, Nadeau, Postnikov, Steingrímsson, Viennot, Williams, Kim, Novelli, Thibon 2005-2012]

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



Outline of the talk

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

Outline of the talk

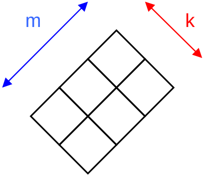
1. Enumeration of non-ambiguous trees inside a rectangular box
2. Enumeration of non-ambiguous trees with a given underlying binary tree
3. Complete non-ambiguous trees and the Bessel function
4. A bijection between parallelogram polyominoes and binary trees

Non-ambiguous trees inside a box

$A(m, k) = \#$ non-ambiguous trees inside a $m \times k$ box

Non-ambiguous trees inside a box

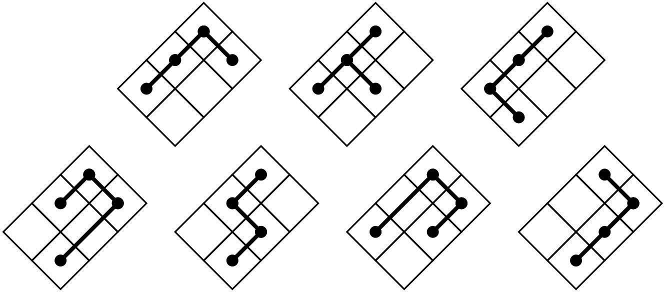
$A(m, k) = \#$ non-ambiguous trees inside a $m \times k$ box



$A(3, 2) =$

Non-ambiguous trees inside a box

$A(m, k) = \#$ non-ambiguous trees inside a $m \times k$ box



$A(3, 2) = 7$

Non-ambiguous trees inside a box

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

$$\sum_{k=1}^{n} s(n,k)\, A(m,k) = n^{m-1}\, n!\,,$$

where $s(n,k)$ are the Stirling numbers of first kind.

Non-ambiguous trees inside a box

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

$$\sum_{k=1}^{n} s(n,k)\, A(m,k) = n^{m-1}\, n!\,,$$

where $s(n,k)$ are the Stirling numbers of first kind.

Proposition

$A(m,k) = p(m+k-1,m) = \#$ permutations of size $m+k-1$ having m excedences at positions $1,2,\ldots,m$

Non-ambiguous trees inside a box

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

$$\sum_{k=1}^{n} s(n,k)\, A(m,k) = n^{m-1}\, n!\,,$$

where $s(n,k)$ are the Stirling numbers of first kind.

Proposition

$A(m,k) = p(m+k-1,m) = \#$ permutations of size $m+k-1$ having m excedences at positions $1,2,\ldots,m$

Theorem [Ehrenborg, Steingrímsson 2000]

$$p(m+k,m) = \sum_{i=1}^{k+1} (-1)^{k+1-i} S(k+1,i)\, i!\, i^m\,,$$

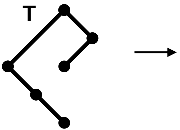
where $S(n,i)$ are the Stirling numbers of second kind.

Outline of the talk

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

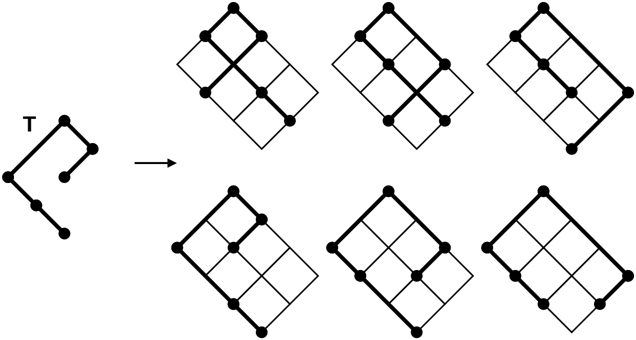
Non-ambiguous trees associated with a given binary tree

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



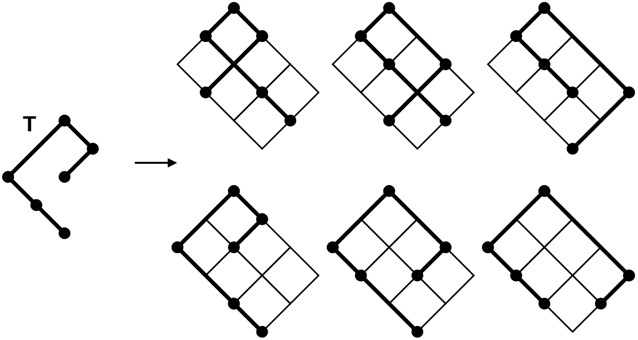
Non-ambiguous trees associated with a given binary tree

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



Non-ambiguous trees associated with a given binary tree

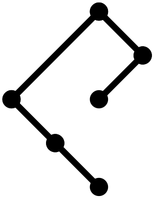
$NA(T) = \#$ non-ambiguous trees whose underlying binary tree is T.



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

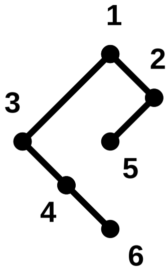
Non-ambiguous trees associated with a given binary tree

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



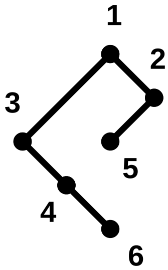
Non-ambiguous trees associated with a given binary tree

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

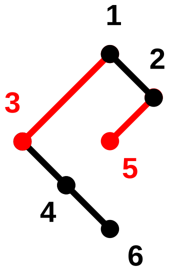
To distinguish the vertices of a binary tree T, we label them by integers to 1 to $|T|$:



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

Non-ambiguous trees associated with a given binary tree

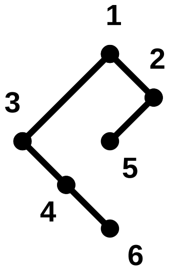
To distinguish the vertices of a binary tree T, we label them by integers to 1 to $|T|$:



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

Non-ambiguous trees associated with a given binary tree

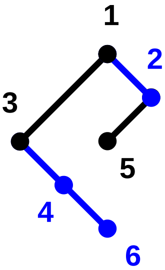
To distinguish the vertices of a binary tree T, we label them by integers to 1 to $|T|$:



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

Non-ambiguous trees associated with a given binary tree

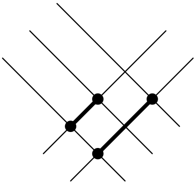
To distinguish the vertices of a binary tree T, we label them by integers to 1 to $|T|$:

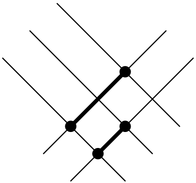


$V_L = \{$ left children $\}$ (here $V_L = \{3, 5\}$)
$V_R = \{$ right children $\}$ (here $V_R = \{2, 4, 6\}$)

The coordinates of left and right children

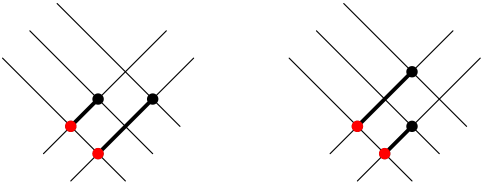
Two left children cannot belong to the same row





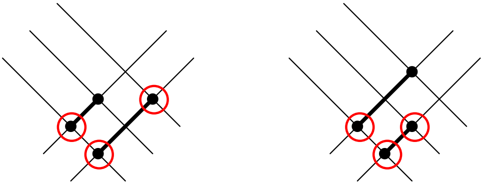
The coordinates of left and right children

Two left children cannot belong to the same row



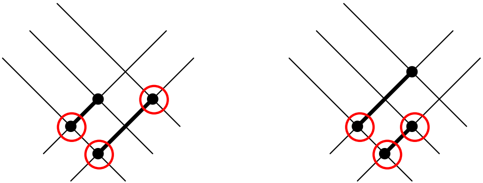
The coordinates of left and right children

Two left children cannot belong to the same row



The coordinates of left and right children

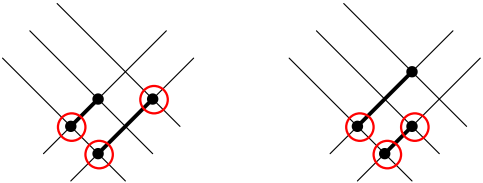
Two left children cannot belong to the same row



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

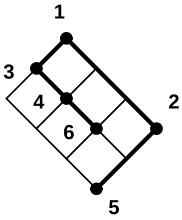
Two left children cannot belong to the same row



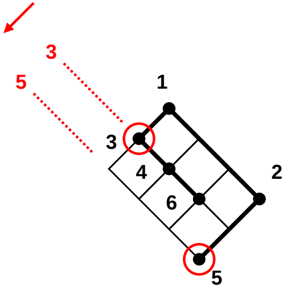
Hence, all the elements in V_L have different x-coordinates, and these coordinates form the interval $[1, |V_L|]$.

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 α_L and α_R

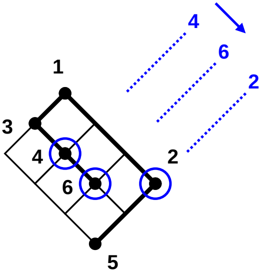


The words α_L and α_R



$\alpha_L = 3\,5$

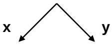
The words α_L and α_R



$$\alpha_L = 3\,5$$

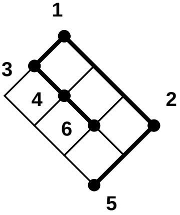
$$\alpha_R = 4\,6\,2$$

A triple representing a non-ambiguous tree

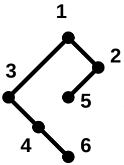


$$\Phi(A) = (T, \alpha_L, \alpha_R)$$

$A =$

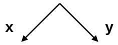


$\Leftrightarrow$ ($T =$

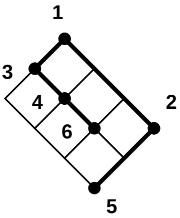


, $\alpha_L = 3\,5$, $\alpha_R = 4\,6\,2$)

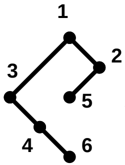
A triple representing a non-ambiguous tree



$$\Phi(A) = (T, \alpha_L, \alpha_R)$$

$A =$ 

$\Leftrightarrow$

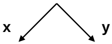
$\Bigg($ $T =$ 

$, \quad \alpha_L = 3\,5, \quad \alpha_R = 4\,6\,2 \Bigg)$

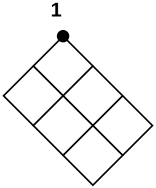
We can reconstruct A from $\Phi(A)$: $X(1) = Y(1) = 0$ and

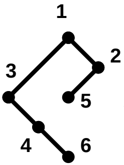
$$\begin{cases} X(i) = \alpha_L^{-1}(i) \text{ and } Y(i) = Y(\text{parent}(i)) & i \text{ left child} \\ Y(j) = \alpha_R^{-1}(j) \text{ and } X(j) = X(\text{parent}(j)) & j \text{ right child} \end{cases}$$

A triple representing a non-ambiguous tree



$$\Phi(A) = (T, \color{red}{\alpha_L}, \color{blue}{\alpha_R})$$

$A =$ 

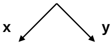
$\Leftrightarrow$ $\left(T = \right.$ 

$, \quad \color{red}{\alpha_L} = \color{red}{3\,5}, \quad \color{blue}{\alpha_R} = \color{blue}{4\,6\,2} \left. \right)$

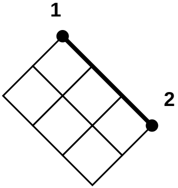
We can reconstruct A from $\Phi(A)$: $X(1) = Y(1) = 0$ and

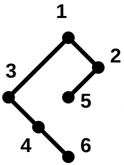
$$\begin{cases} X(i) = \alpha_L^{-1}(i) \text{ and } Y(i) = Y(\text{parent}(i)) & i \text{ left child} \\ Y(j) = \alpha_R^{-1}(j) \text{ and } X(j) = X(\text{parent}(j)) & j \text{ right child} \end{cases}$$

A triple representing a non-ambiguous tree



$$\Phi(A) = (T, \alpha_L, \alpha_R)$$

$A =$ 

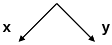
$\Leftrightarrow$ $\Bigg($ $T =$ 

$, \quad \alpha_L = 3\,5, \quad \alpha_R = 4\,6\,\underline{2}$ $\Bigg)$

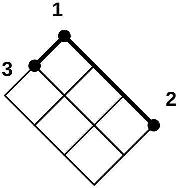
We can reconstruct A from $\Phi(A)$: $X(1) = Y(1) = 0$ and

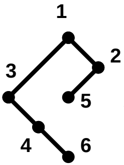
$$\begin{cases} X(i) = \alpha_L^{-1}(i) \text{ and } Y(i) = Y(\text{parent}(i)) & i \text{ left child} \\ Y(j) = \alpha_R^{-1}(j) \text{ and } X(j) = X(\text{parent}(j)) & j \text{ right child} \end{cases}$$

A triple representing a non-ambiguous tree



$$\Phi(A) = (T, \color{red}{\alpha_L}, \color{blue}{\alpha_R})$$

$A =$ 

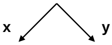
$\Leftrightarrow$ $\Bigg($ $T =$ 

$,\quad \color{red}{\alpha_L = \underline{3}\,5}, \quad \color{blue}{\alpha_R = 4\,6\,2}$ $\Bigg)$

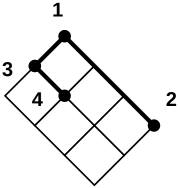
We can reconstruct A from $\Phi(A)$: $X(1) = Y(1) = 0$ and

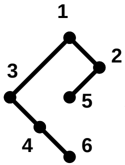
$$\begin{cases} X(i) = \alpha_L^{-1}(i) \text{ and } Y(i) = Y(\text{parent}(i)) & i \text{ left child} \\ Y(j) = \alpha_R^{-1}(j) \text{ and } X(j) = X(\text{parent}(j)) & j \text{ right child} \end{cases}$$

A triple representing a non-ambiguous tree



$$\Phi(A) = (T, \alpha_L, \alpha_R)$$

$A =$ 

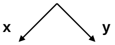
$\Leftrightarrow$ $\Bigg($ $T =$ 

$, \quad \alpha_L = 3\,5, \quad \alpha_R = \underline{4}\,6\,2 \Bigg)$

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

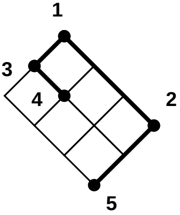
$$\begin{cases} X(i) = \alpha_L^{-1}(i) \text{ and } Y(i) = Y(\text{parent}(i)) & i \text{ left child} \\ Y(j) = \alpha_R^{-1}(j) \text{ and } X(j) = X(\text{parent}(j)) & j \text{ right child} \end{cases}$$

A triple representing a non-ambiguous tree

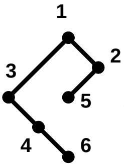


$$\Phi(A) = (T, {\color{red}\alpha_L}, {\color{blue}\alpha_R})$$

$A =$



$\Leftrightarrow$ ($T =$

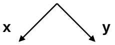


, ${\color{red}\alpha_L = 3\,\underline{5}}$, ${\color{blue}\alpha_R = 4\,6\,2}$)

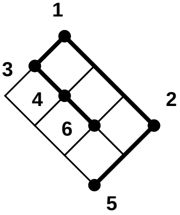
We can reconstruct A from $\Phi(A)$: $X(1) = Y(1) = 0$ and

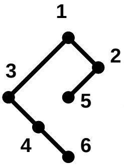
$$\begin{cases} X(i) = \alpha_L^{-1}(i) \text{ and } Y(i) = Y(\text{parent}(i)) & i \text{ left child} \\ Y(j) = \alpha_R^{-1}(j) \text{ and } X(j) = X(\text{parent}(j)) & j \text{ right child} \end{cases}$$

A triple representing a non-ambiguous tree



$$\Phi(A) = (T, \color{red}{\alpha_L}, \color{blue}{\alpha_R})$$

$A =$ 

$\Leftrightarrow \left(T = \right.$ 

$\left. , \quad \color{red}{\alpha_L = 3\,5}, \quad \color{blue}{\alpha_R = 4\,\underline{6}\,2} \right)$

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 triple representing a non ambiguous tree

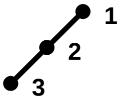
We can reconstruct A from $\Phi(A)$, hence Φ is injective.

Question: what is the image of Φ? Are there some constraints on α_L and α_R?

A triple representing a non ambiguous tree

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

Question: what is the image of Φ? Are there some constraints on α_L and α_R?

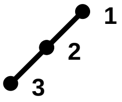


Here we must have $\alpha_L = 2\,3$

A triple representing a non ambiguous tree

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

Question: what is the image of Φ? Are there some constraints on α_L and α_R?



Here we must have $\alpha_L = 2\,3$

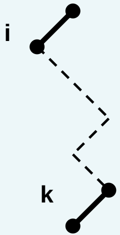
Actually, we have some constraints on α_L and α_R

- α_L must be a linear extension of a particular poset defined on V_L
- α_R must be a linear extension of a particular poset defined on V_R

The posets defined on V_L and V_R

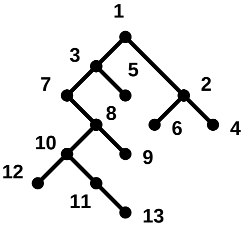
The order relation $<_{V_L}$

Given $i, k \in V_L$

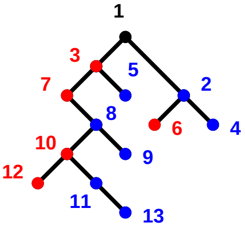


$$i <_{V_L} k \quad \Longleftrightarrow \quad i \text{ is an ancestor of } k.$$

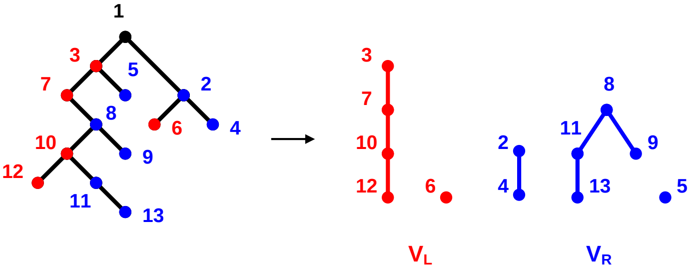
The posets defined on V_L and V_R



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 α_L and α_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 α_L.

The constraints on α_L and α_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 α_L.

Constraints

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

The constraints on α_L and α_R

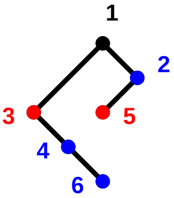
If i is an ancestor of k, and we must also have $X(i) < X(k)$. This means that i appears to the left of k in α_L.

Constraints

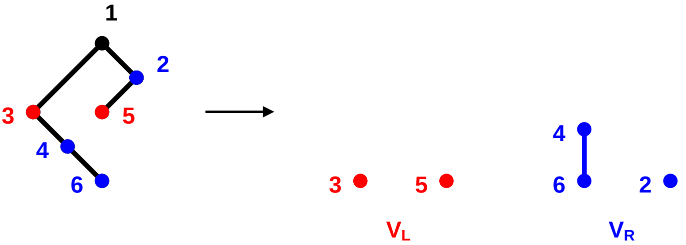
α_L must be a linear extension of the poset $(V_L, <_{V_L}) \rightarrow \alpha_L \in \mathcal{L}(V_L)$
α_R must be a linear extension of the poset $(V_R, <_{V_R}) \rightarrow \alpha_R \in \mathcal{L}(V_R)$

We can show that these coniditions are also sufficient.

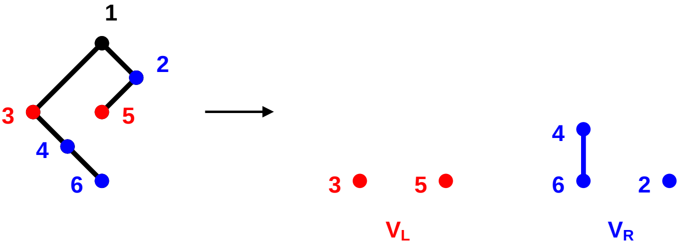
The linear extensions of V_L and V_R



The linear extensions of V_L and V_R

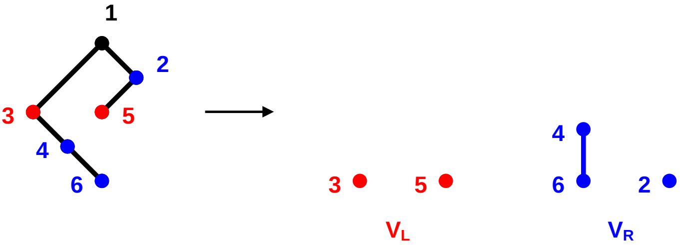


The linear extensions of V_L and V_R



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

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

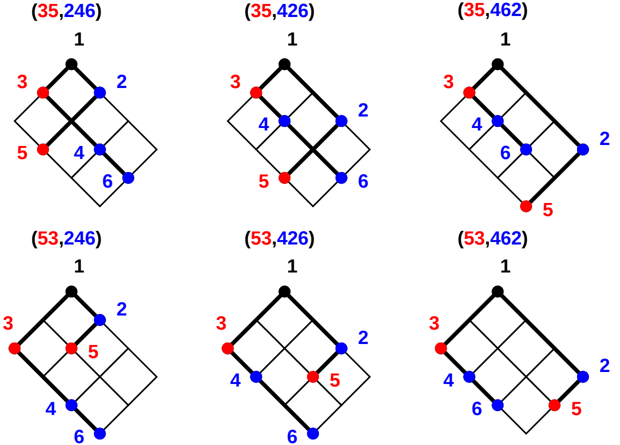
$$\mathcal{L}(V_L) \times \mathcal{L}(V_R) = \{(35, 246), (35, 426), (35, 462), (53, 246), (53, 426), (53, 462)\}$$

(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 [ABBS 2012]

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

Enumerative consequences

Theorem [ABBS 2012]

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

Theorem [ABBS 2012]

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

Enumerative consequences

Theorem [ABBS 2012]

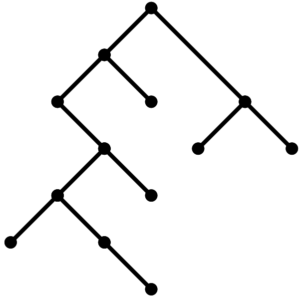
$$NA(T) = |\mathcal{L}(V_L)| \times |\mathcal{L}(V_R)|$$

Theorem [ABBS 2012]

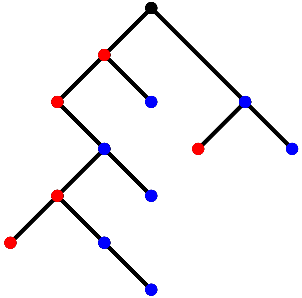
The Hasse diagrams of $(V_L, <_{V_L})$ and $(V_R, <_{V_R})$ are forests.

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

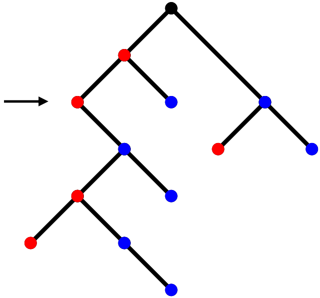
A hook formula for $NA(T)$



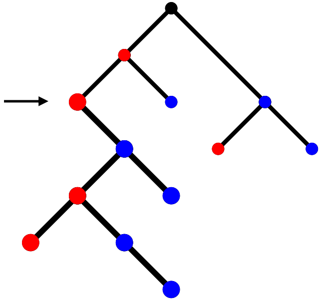
A hook formula for $NA(T)$



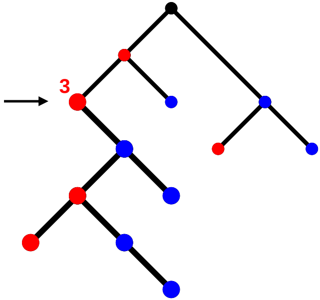
A hook formula for $NA(T)$



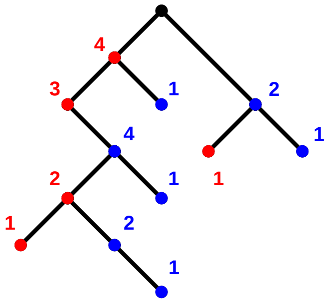
A hook formula for $NA(T)$



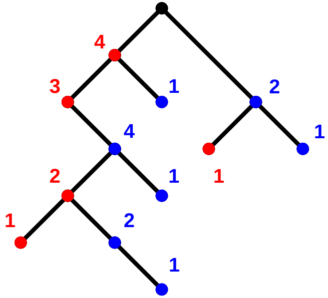
A hook formula for $NA(T)$



A hook formula for $NA(T)$



A hook formula for $NA(T)$



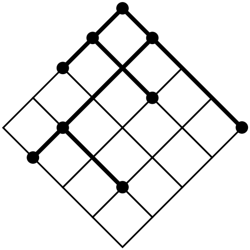
$$NA(T) = \frac{|V_L|!}{\prod_{e \in V_L} \lambda(e)} \cdot \frac{|V_R|!}{\prod_{e \in V_R} \lambda(e)} = \frac{5!}{4 \cdot 3 \cdot 2 \cdot 1 \cdot 1} \cdot \frac{7!}{4 \cdot 2 \cdot 2 \cdot 1 \cdot 1 \cdot 1 \cdot 1} = 1575$$

Outline of the talk

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

Complete non-ambiguous trees

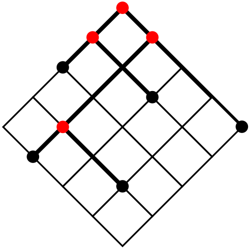
A complete non-ambiguous tree is a non-ambiguous tree whose underlying binary tree is complete (i.e. every vertex has 0 or 2 children)



size =

Complete non-ambiguous trees

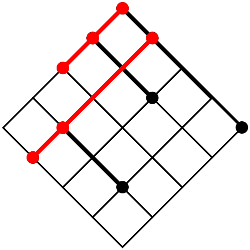
A complete non-ambiguous tree is a non-ambiguous tree whose underlying binary tree is complete (i.e. every vertex has 0 or 2 children)



size $=$ $\#$ internal vertices

Complete non-ambiguous trees

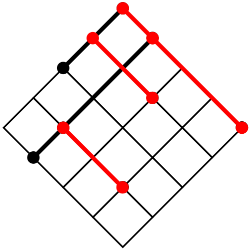
A complete non-ambiguous tree is a non-ambiguous tree whose underlying binary tree is complete (i.e. every vertex has 0 or 2 children)



size = # internal vertices = # 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)



size = # internal vertices = # left edges = # 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:

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

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:

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

with no combinatorial interpretation.

Theorem [ABBS 2012]

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

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:

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

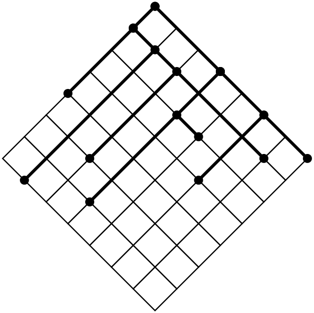
such that $y(0) = 1$ and $y'(0) = 0$. The function

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

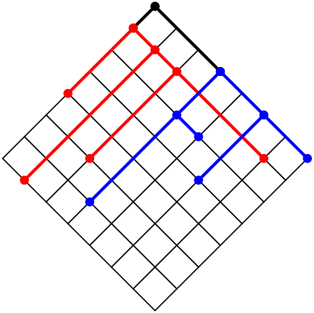
solves this differential equation if and only if the sequence b_n satisfies the following recurrence

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

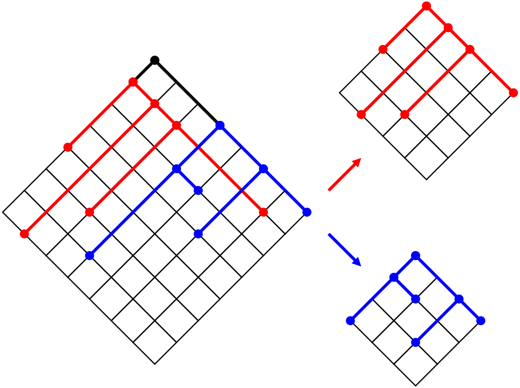
Recursive enumeration of complete non-ambiguous trees



Recursive enumeration of complete non-ambiguous trees

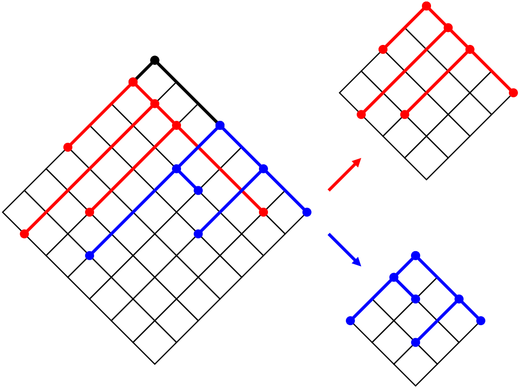


Recursive enumeration of complete non-ambiguous trees



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

Recursive enumeration of complete non-ambiguous trees



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

Another identity involving complete non-ambiguous trees

Theorem [Carlitz 1963, combinatorial proof ABBS 2012]

For every $n \geq 1$

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

Another identity involving complete non-ambiguous trees

Theorem [Carlitz 1963, combinatorial proof ABBS 2012]

For every $n \geq 1$

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

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

Another identity involving complete non-ambiguous trees

Theorem [Carlitz 1963, combinatorial proof ABBS 2012]

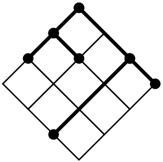
For every $n \geq 1$

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

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

Gridded trees

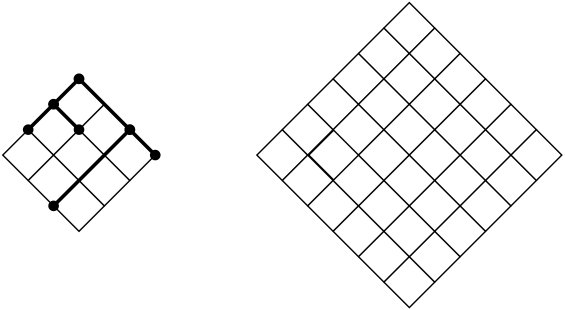
A gridded tree of size (k, n) is a complete non-ambiguous tree of size k embedded in a $n \times n$ grid:



A complete non-ambiguous tree of size 3

Gridded trees

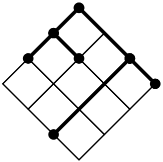
A gridded tree of size (k, n) is a complete non-ambiguous tree of size k embedded in a $n \times n$ grid:

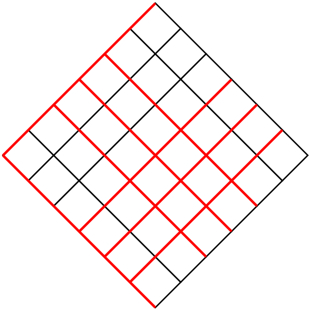


A complete non-ambiguous tree of size 3

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:

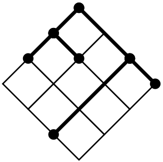


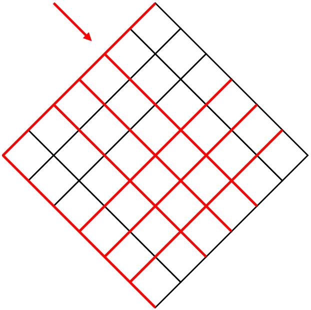


A complete non-ambiguous tree of size 3

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:

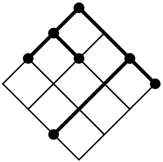




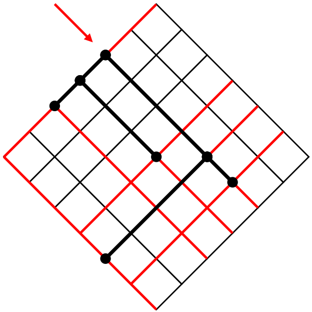
A complete non-ambiguous tree of size 3

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

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

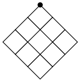
Gridded trees

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

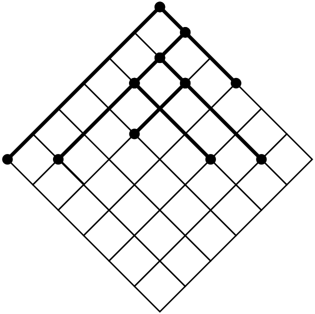
$$g_{k,n} = \binom{n}{k+1}\binom{n-1}{k} b_k$$

The identity that we want to prove becomes

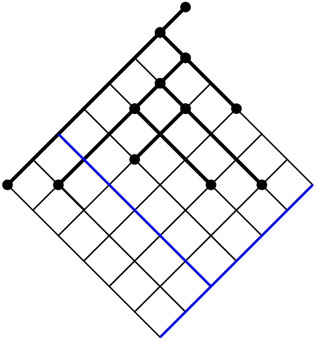
$$\sum_{k=0}^{n-1} (-1)^k \binom{n}{k+1}\binom{n-1}{k} b_k = \sum_{k=0}^{n-1} (-1)^k g_{k,n} = 1 \equiv$$



An involution on gridded trees

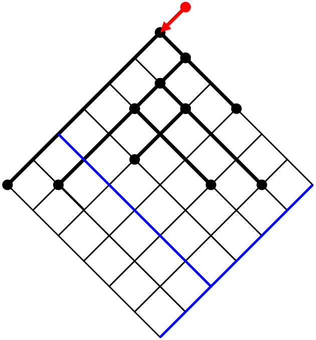


An involution on gridded trees



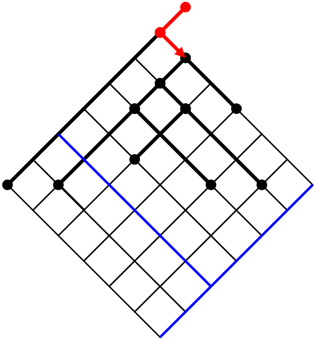
Case 1: the zig-zag path doesn't cross an empty row/column

An involution on gridded trees



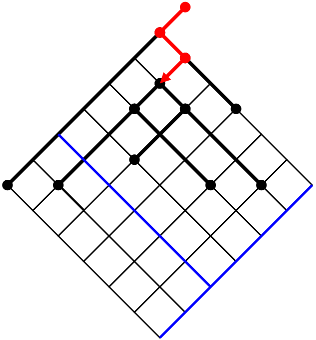
Case 1: the zig-zag path doesn't cross an empty row/column

An involution on gridded trees



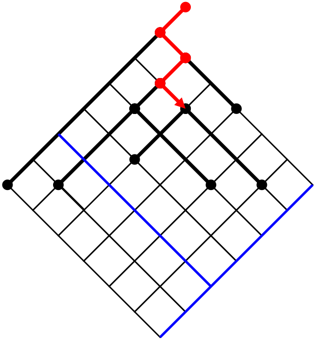
Case 1: the zig-zag path doesn't cross an empty row/column

An involution on gridded trees



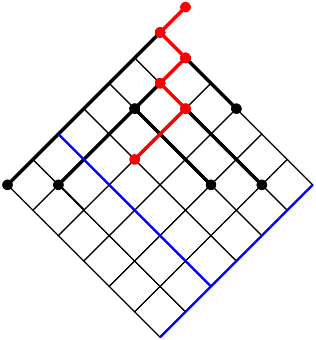
Case 1: the zig-zag path doesn't cross an empty row/column

An involution on gridded trees



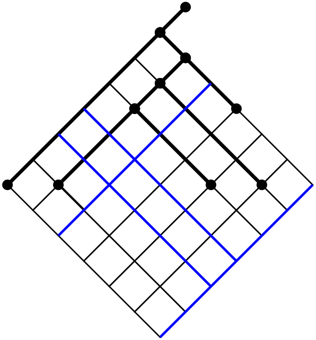
Case 1: the zig-zag path doesn't cross an empty row/column

An involution on gridded trees



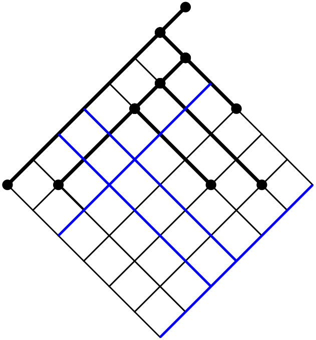
Case 1: the zig-zag path doesn't cross an empty row/column

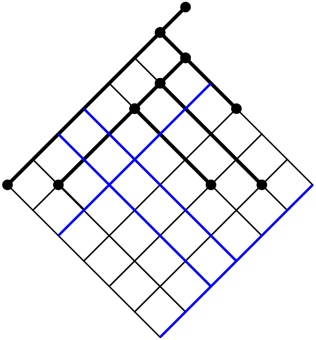
An involution on gridded trees



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

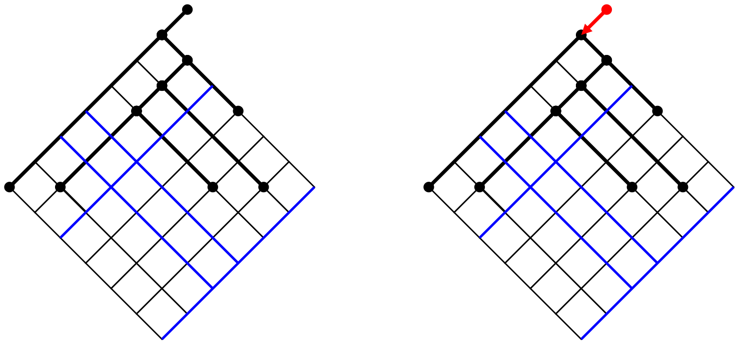
An involution on gridded trees





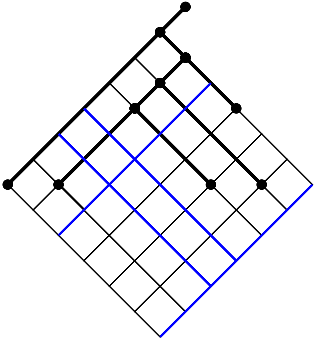
Case 2: the zig-zag path crosses an empty row/column

An involution on gridded trees



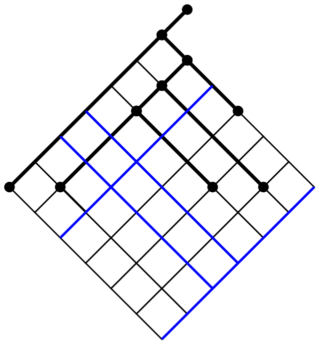
Case 2: the zig-zag path crosses an empty row/column

An involution on gridded trees



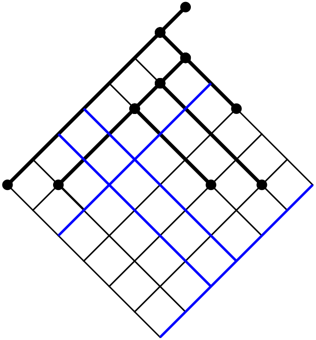
Case 2: the zig-zag path crosses an empty row/column

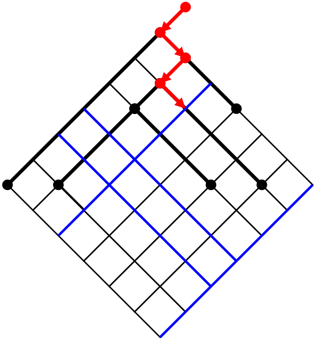
An involution on gridded trees



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

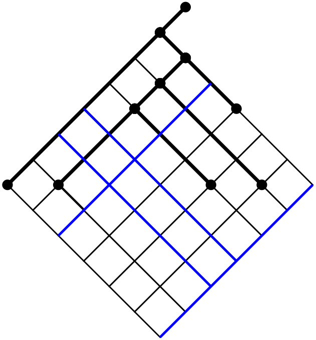
An involution on gridded trees

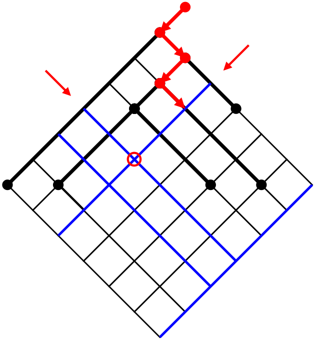




Case 2: the zig-zag path crosses an empty row/column

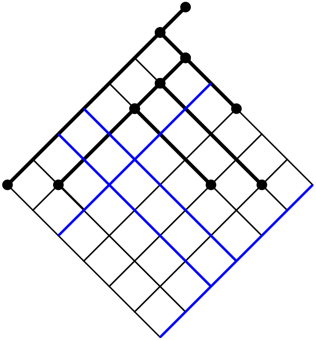
An involution on gridded trees

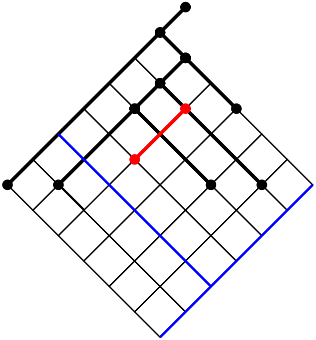




Case 2: the zig-zag path crosses an empty row/column

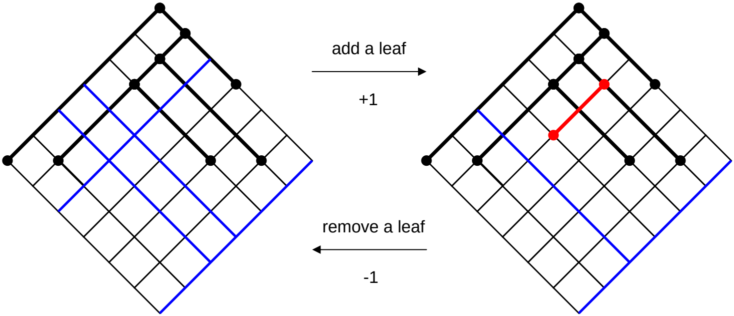
An involution on gridded trees





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

An involution on gridded trees



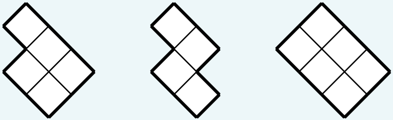
Outline of the talk

1. Enumeration of non-ambiguous trees inside a rectangular box
2. Enumeration of non-ambiguous trees with a given underlying binary tree
3. Complete non-ambiguous trees and the Bessel function
4. 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.

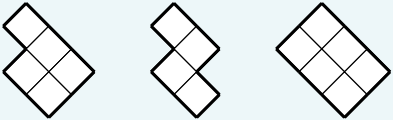
Some examples for $n = 4$



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.

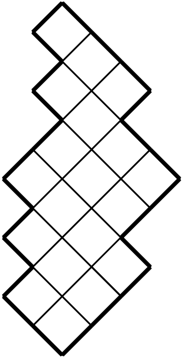
Some examples for $n = 4$



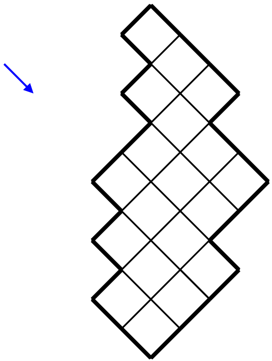
Theorem [Delest, Viennot, . . .]

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

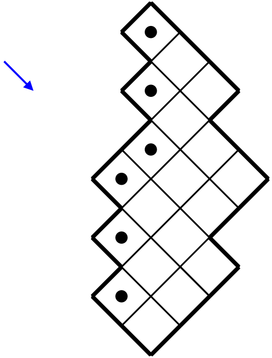
A new bijection S between PPs and binary trees



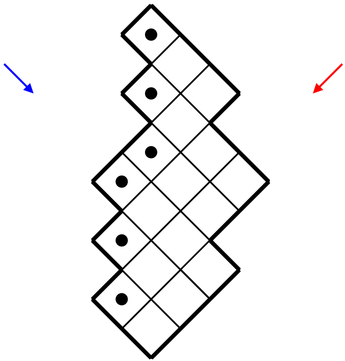
A new bijection S between PPs and binary trees



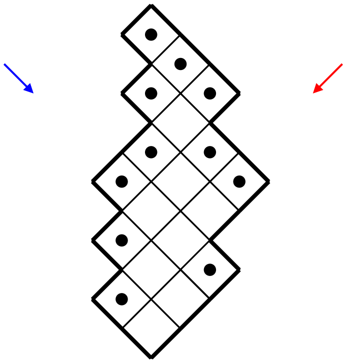
A new bijection S between PPs and binary trees



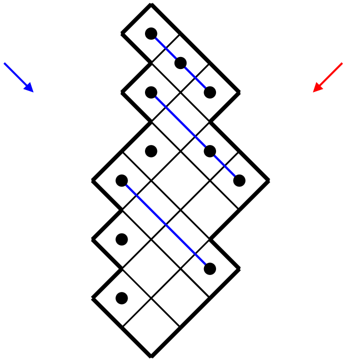
A new bijection S between PPs and binary trees



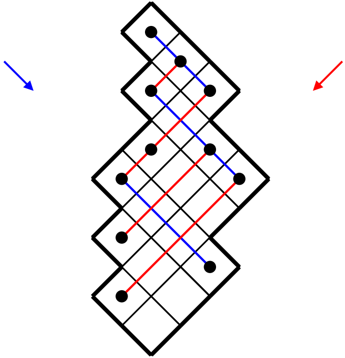
A new bijection S between PPs and binary trees



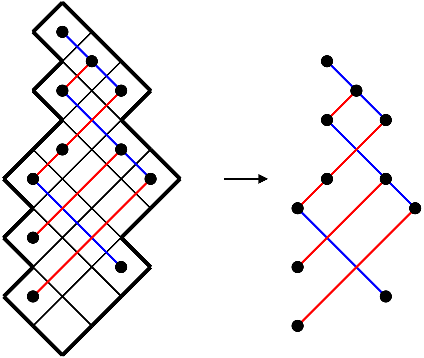
A new bijection S between PPs and binary trees



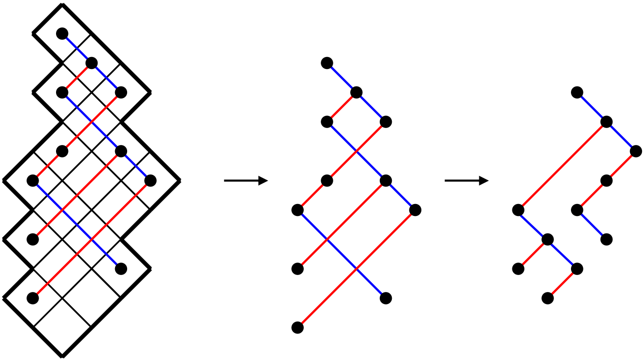
A new bijection S between PPs and binary trees



A new bijection S between PPs and binary trees



A new bijection S between PPs and binary trees



Some perspectives

- Find the OGF/EGF of non-ambiguous trees and of complete non-ambiguous trees
- 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
- Study the relationship between non-ambiguous trees and a family of tilings of a rectangle, the *floorplans*. Floorplans are in bijection with Baxter permutations [Ackerman et al. 2006]

Some perspectives

- Find the OGF/EGF of non-ambiguous trees and of complete non-ambiguous trees
- Find an analogue of complete non-ambiguous trees enumerated by the coefficients in the expansion of $-\ln\left(J_k(x)\right)$ for other values of k
- Define an analogue of non-ambiguous trees in higher dimensions
- Study the relationship between non-ambiguous trees and a family of tilings of a rectangle, the *floorplans*. Floorplans are in bijection with Baxter permutations [Ackerman et al. 2006]

Thank you for your attention!