Bijective counting of tree-rooted maps

Olivier Bernardi - LaBRI, Bordeaux

Combinatorics and Optimization seminar, March 2006, Waterloo University
Bijective counting of tree-rooted maps

Maps and trees.

Tree-rooted maps and parenthesis systems. (Mullin, Lehman & Walsh)

Bijection:

Tree-rooted maps $\iff$ Trees $\times$ Non-crossing partitions.

Isomorphism with a construction by Cori, Dulucq and Viennot.
Maps and trees
Planar maps

A map is a connected planar graph properly embedded in the oriented sphere. The map is considered up to deformation.
Planar maps

A **map** is a connected planar graph properly embedded in the oriented sphere. The map is considered up to deformation.

A map is **rooted** by adding a half-edge in a corner.
A tree is a map with only one face.
A tree is a map with only one face.

The size of a map, a tree, is the number of edges.
Tree-rooted maps

A submap is a spanning tree if it is a tree containing every vertex.
Tree-rooted maps

A submap is a spanning tree if it is a tree containing every vertex.

A tree-rooted map is a rooted map with a distinguished spanning tree.
Tree-rooted maps and Parenthesis systems
(Mullin, Lehman & Walsh)
Parenthesis systems

A parenthesis system is a word $w$ on $\{a, \bar{a}\}$ such that $|w|_a = |w'|_{\bar{a}}$ and for all prefix $w'$, $|w'|_a \geq |w'|_{\bar{a}}$.

Example: $w = aa\bar{a}a\bar{a}a\bar{a}$ is a parenthesis system.
Parenthesis shuffle

A parenthesis shuffle is a word \( w \) on \( \{a, \overline{a}, b, \overline{b}\} \) such that the subwords made of \( \{a, \overline{a}\} \) letters and \( \{b, \overline{b}\} \) letters are parenthesis systems.

Example: \( w = baab\overline{a}b\overline{a}baab\overline{a}a \) is a parenthesis shuffle.
Parenthesis shuffle

A parenthesis shuffle is a word \( w \) on \( \{ a, \overline{a}, b, \overline{b} \} \) such that the subwords made of \( \{ a, \overline{a} \} \) letters and \( \{ b, \overline{b} \} \) letters are parenthesis systems.

Example: \( w = baab\overline{abb}\overline{aba}\overline{ab}\overline{a}a \) is a parenthesis shuffle.

The size of a parenthesis system, shuffle is half its length.
Trees and parenthesis systems

Rooted trees of size $n$ are in bijection with parenthesis systems of size $n$. 
Trees and parenthesis systems

We turn around the tree and write:

\( a \) the first time we follow an edge,
\( \bar{a} \) the second time.
We turn around the tree and write:

\( a \) the first time we follow an edge,
\( \bar{a} \) the second time.
Tree-rooted maps and parenthesis shuffles

[Mullin 67, Lehman & Walsh 72] Tree-rooted maps of size $n$ are in bijection with parenthesis shuffles of size $n$. 
We turn around the tree and write:

- $a$ the first time we follow an internal edge,
- $\overline{a}$ the second time,
- $b$ the first time we cross an external edge,
- $\overline{b}$ the second time.
We turn around the tree and write:

- $a$ the first time we follow an internal edge,
- $\overline{a}$ the second time,
- $b$ the first time we cross an external edge,
- $\overline{b}$ the second time.
Counting results

There are $C_k = \frac{1}{k+1} \binom{2k}{k}$ parenthesis systems of size $k$. 
Counting results

There are \( C_k = \frac{1}{k+1} \binom{2k}{k} \) parenthesis systems of size \( k \).

There are \( \binom{2n}{2k} \) ways of shuffling a parenthesis system of size \( k \) (on \( \{a, \overline{a}\} \)) and a parenthesis system of size \( n-k \) (on \( \{b, \overline{b}\} \)).
Counting results

- There are $C_k = \frac{1}{k+1} \binom{2k}{k}$ parenthesis systems of size $k$.

- There are $\binom{2n}{2k}$ ways of shuffling a parenthesis system of size $k$ (on $\{a, \overline{a}\}$) and a parenthesis system of size $n - k$ (on $\{b, \overline{b}\}$).

$\implies$ There are $M_n = \sum_{k=0}^{n} \binom{2n}{2k} C_k C_{n-k}$ parenthesis shuffles of size $n$. 
Counting results

\[ M_n = \sum_{k=0}^{n} \binom{2n}{2k} C_k C_{n-k} \]

\[ = \frac{(2n)!}{(n+1)!^2} \sum_{k=0}^{n} \binom{n+1}{k} \binom{n+1}{n-k} \]

\[ = \frac{(2n)!}{(n+1)!^2} \binom{2n+2}{n} \]
Counting results

\[ M_n = \sum_{k=0}^{n} \binom{2n}{2k} C_k C_{n-k} \]

\[ = \frac{(2n)!}{(n+1)!^2} \sum_{k=0}^{n} \binom{n+1}{k} \binom{n+1}{n-k} \]

\[ = \frac{(2n)!}{(n+1)!^2} \binom{2n+2}{n} \]

**Theorem**: The number of parenthesis shuffles of size \( n \) is

\[ M_n = C_n C_{n+1}. \]
Counting results

$$\mathcal{M}_n = \sum_{k=0}^{n} \binom{2n}{2k} C_k C_{n-k}$$

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

$$= \frac{(2n)!}{(n+1)!^2} \binom{2n+2}{n}$$

**Theorem [Mullin 67]**: The number of tree-rooted maps of size $n$ is

$$\mathcal{M}_n = C_n C_{n+1}.$$
A pair of trees?

Theorem [Mullin 67]: The number of tree-rooted maps of size $n$ is

$$\mathcal{M}_n = C_n C_{n+1}.$$ 

Is there a pair of trees hiding somewhere?
A pair of trees?

**Theorem [Mullin 67]**: The number of tree-rooted maps of size $n$ is

$$M_n = C_n C_{n+1}.$$ 

Is there a pair of trees hiding somewhere?

**Theorem [Cori, Dulucq, Viennot 86]**: There is a (recursive) bijection between parenthesis shuffles of size $n$ and pairs of trees.
A pair of trees?

Theorem [Mullin 67]: The number of tree-rooted maps of size $n$ is

$$\mathcal{M}_n = C_n C_{n+1}.$$  

Is there a pair of trees hiding somewhere?

Theorem [Cori, Dulucq, Viennot 86]: There is a (recursive) bijection between parenthesis shuffles of size $n$ and pairs of trees.

Is there a good interpretation on maps?
Tree-rooted maps

Trees $\times$ Non-crossing partitions
Orientations of tree-rooted maps
Orientations of tree-rooted maps

Internal edges are oriented from the root to the leaves.
Orientations of tree-rooted maps

- Internal edges are oriented from the root to the leaves.
- External edges are oriented in such a way their heads appear before their tails around the tree.
Orientations of tree-rooted maps

Proposition:

The orientation is *root-connected*:
there is an oriented path from the root to any vertex.
Orientations of tree-rooted maps

Proposition:
- The orientation is root-connected: there is an oriented path from the root to any vertex.
- The orientation is minimal: every directed cycle is oriented clockwise.
Orientations of tree-rooted maps

Proposition:
- The orientation is root-connected: there is an oriented path from the root to any vertex.
- The orientation is minimal: every directed cycle is oriented clockwise.

We call tree-orientation a minimal root-connected orientation.
Orientations of tree-rooted maps

**Theorem** : The orientation of edges in tree-rooted maps gives a bijection between *tree-rooted maps* and *tree-oriented maps*. 
Orientations of tree-rooted maps

**Theorem**: The orientation of edges in tree-rooted maps gives a bijection between tree-rooted maps and tree-oriented maps.
From the orientation to the tree

We turn around the tree we are constructing.
From the orientation to the tree

We turn around the tree we are constructing.
From the orientation to the tree

We turn around the tree we are constructing.
From the orientation to the tree

We turn around the tree we are constructing.
From the orientation to the tree

We turn around the tree we are constructing.
From the orientation to the tree

We turn around the tree we are constructing.
From the orientation to the tree

We turn around the tree we are constructing.
Vertex explosion
Vertex explosion

We explode the vertex and obtain a vertex per ingoing edge + a (gluing) cell.
Example
Example
Example

A tree!
Bijection

Proposition:
- The map obtained by exploding the vertices is a tree.
Bijection

Proposition:
- The map obtained by exploding the vertices is a tree.
- The gluing cells are incident to the first corner of each vertex. They define a non-crossing partition of the vertices of the tree.
**Bijection**

**Theorem**: The orientation of tree-rooted maps and the explosion of vertices gives a bijection between tree-rooted maps of size $n$ and trees of size $n \times$ non-crossing partitions of size $n + 1$. 
Bijection

**Theorem**: The orientation of tree-rooted maps and the explosion of vertices gives a bijection between tree-rooted maps of size $n$ and trees of size $n$ non-crossing partitions of size $n + 1$.

**Corollary**: $M_n = C_n C_{n+1}$. 
Example
Example
Example
Example
Example
Isomorphism with a bijection by Cori, Dulucq and Viennot
**Tree code** \( \Phi \)

**Definition:**

- \( \Phi(\epsilon) = u \cdot v. \)
- \( \Phi_a : \) Replace last occurrence of \( u \) by \( u \cdot v. \)
- \( \Phi_b : \) Replace first occurrence of \( v \) by \( u \cdot v. \)
- \( \Phi_{\bar{a}} : \) Replace first occurrence of \( v \) by \( a v T_2 T_1. \)
- \( \Phi_{\bar{b}} : \) Replace last occurrence of \( u \) by \( b u T_2 T_1. \)
Tree code $\Phi$

Example: $ba\bar{a}a\bar{b}\bar{a}$

$$\begin{array}{c}
u \quad v \quad b \\ u \quad u \quad v \quad a \\ u \quad u \quad v \quad \bar{a} \\ u \quad u \quad v \quad a \\ u \quad u \quad v \quad \bar{b} \\ u \quad v \quad v \quad \bar{a} \\ u \quad v \quad v \quad \bar{a} \\ u \quad v \quad v \quad u \\
\end{array}$$
Tree code $\Phi$

Example: $b\bar{a}\bar{a}ab\bar{a}$
Partition code $\Psi$

Definition:

- $\Psi(\epsilon)$:
  - $\Psi_a$: Replace last active left leaf.
  - $\Psi_b$: Replace first active right leaf.
  - $\Psi_a^-$: Inactivate first active right leaf.
  - $\Psi_b^-$: Inactivate last active left leaf.
Partition code \( \Psi \)

Example: \( ba\overline{a}a\overline{b}\overline{a} \)
Partition code $\Psi$

Example: $ba\overline{a}a\overline{b}a$

Diagram of partition code: 

\[
\begin{align*}
\begin{array}{c}
\bullet & \rightarrow & \bullet & \rightarrow & \overline{a} & \rightarrow & a & \rightarrow & \overline{b} & \rightarrow & \overline{a} \\
\end{array}
\end{align*}
\]

Rooted tree representation of $ba\overline{a}a\overline{b}a$ under $\Psi$: 

\[
\begin{array}{c}
\bullet & \rightarrow & \bullet & \rightarrow & \overline{a} & \rightarrow & a & \rightarrow & \overline{b} & \rightarrow & \overline{a} \\
\end{array}
\]

Waterloo, March 2006
Isomorphism

\[ \text{baabaaba} \]
Isomorphism tree code:
Isomorphism tree code:
Isomorphism tree code:
Isomorphism tree code:
Isomorphism tree code:
Isomorphism tree code:
Isomorphism tree code:
Isomorphism tree code:
Isomorphism tree code:
Isomorphism tree code:
Isomorphism tree code:
Thanks.