

Increasing Schröder trees and restricted permutations

Permutation Patterns 2023

Mehdi Naima July, 4th 2023

RWTH Aachen University, Germany

Introduction

Evolutionary trees



- Represent the evolutionary relationship among species
- A Multifurcation means the descendant species have distinguished themselves



- Earliest illustrations made by Darwin in his book "On The Origin of Species" (1859)
- Represent divergence in characters



- Schröder published "Vier combinatorische Probleme" (1870)
- Asks for the number of ways a string of *n* identical letters, can be "bracketed"

Symbolic Method and Schröder trees

Ordinary generating function of a class C to be $C(z) = \sum_{n=0}^{\infty} C_n z^n$ where C_n is # of objects of size n.

Symbolic method systematized by Flajolet and Sedgewick is a grammar used to define (specify) combinatorial classes:

Operation	Notation	Description	OGF
Neutral class	ε	Class consisting of single object of size 0	1
Atomic class	Z	Class consisting of single object of size 1	z
Disjoint Union	$\mathcal{F}+\mathcal{G}$	Disjoint of objects from ${\mathcal F}$ and ${\mathcal G}$	F(z) + G(z)
Cartesian product	$\mathcal{F}\times \mathcal{G}$	Ordered pairs of objects from ${\mathcal F}$ and ${\mathcal G}$	$F(z) \cdot G(z)$
Sequence	Seq ${\cal F}$	Sequences of objects from ${\cal F}$	$\frac{1}{1-F(z)}$
Substitution	$\mathcal{F}\circ\mathcal{G}$	Substitute elements of ${\mathcal G}$ for atoms of ${\mathcal F}$	F(G(z))
Pointing	$\Theta \mathcal{F}$	Point an atom from ${\cal F}$	zF'(z)

Symbolic Method and Schröder trees

Ordinary generating function of a class C to be $C(z) = \sum_{n=0}^{\infty} C_n z^n$ where C_n is # of objects of size n.

Symbolic method systematized by Flajolet and Sedgewick is a grammar used to define (specify) combinatorial classes:

Operation	Notation	Description	OGF
Neutral class	ε	Class consisting of single object of size 0	1
Atomic class	Z	Class consisting of single object of size 1	z
Disjoint Union	$\mathcal{F}+\mathcal{G}$	Disjoint of objects from ${\mathcal F}$ and ${\mathcal G}$	F(z) + G(z)
Cartesian product	$\mathcal{F}\times \mathcal{G}$	Ordered pairs of objects from ${\mathcal F}$ and ${\mathcal G}$	$F(z) \cdot G(z)$
Sequence	Seq ${\cal F}$	Sequences of objects from $\boldsymbol{\mathcal{F}}$	$\frac{1}{1-F(z)}$
Substitution	$\mathcal{F}\circ\mathcal{G}$	Substitute elements of ${\mathcal G}$ for atoms of ${\mathcal F}$	F(G(z))
Pointing	$\Theta \mathcal{F}$	Point an atom from ${\cal F}$	zF'(z)



Schröder trees

- Size of the tree #Leaves
- Internal nodes arity ≥ 2

$$S = Z + SEQ_{\geq 2}(S) \implies$$

$$S(z) = z + \frac{S(z)^2}{1 - S(z)}$$

$$S(z) = \frac{1}{4}(1 + z - \sqrt{1 - 6z + z^2})$$

$$= z + z^2 + 3z^3 + 11z^4 + 45z^5 + \dots$$

Introduction

Increasing trees

Increasing trees have increasing labels on branches.

- Unique labels, analysis of permutations and data structures like BST using increasing binary trees. Drmota, Mahmoud, Flajolet, ...
- Repeated labellings, parallel executions of processes. Bodini, Gittenberger, Genitrini, Prodinger, Urbanek...



Introduction

Increasing trees

Increasing trees have increasing labels on branches.

- Unique labels, analysis of permutations and data structures like BST using increasing binary trees.
 Drmota, Mahmoud, Flajolet, ...
- Repeated labellings, parallel executions of processes. Bodini, Gittenberger, Genitrini, Prodinger, Urbanek...





$$eta(\sigma) = \langle \min(\sigma), eta(\sigma_L), eta(\sigma_R)
angle, \quad \sigma(\epsilon) = ullet$$

Increasing trees Schröder trees

Definition: increasingly labelled Schröder trees Schröder tree *T* whose internal nodes are labelled with integers such that:

- Root is labelled 1 and along each branch the sequence of labels is strictly increasing.
- If ℓ is the greatest label in the tree, then the set of labels of T is $[\ell]$.
- Label repetitions (allowed or not)



Introduction

Increasing trees Schröder trees

Definition: increasingly labelled Schröder trees Schröder tree *T* whose internal nodes are labelled with integers such that:

- Root is labelled 1 and along each branch the sequence of labels is strictly increasing.
- If ℓ is the greatest label in the tree, then the set of labels of *T* is $[\ell]$.
- Label repetitions (allowed or not)



Increasing trees Schröder trees

Definition: increasingly labelled Schröder trees Schröder tree *T* whose internal nodes are labelled with integers such that:

- Root is labelled 1 and along each branch the sequence of labels is strictly increasing.
- If ℓ is the greatest label in the tree, then the set of labels of *T* is $[\ell]$.
- Label repetitions (allowed or not)



Classical objects

Permutations

Set Partitions

Cycles

- - Cycle canonical order \implies smallest integer 1st
 - $\sigma^{-1}(k)$ returns (i, j), *i* the cycle number that contains *k* and *j* its index

Runs, Rises, Descents

- A rise (resp. descent) two successive increasing (resp. decreasing) elements
- A run is maximal increasing subsequence runs(3, 4, 6, 2, 5, 1) = 3
- # of *n*-permutations with *k* des. Eulerian numbers $\langle {n \atop k} \rangle$
- · des(p) = runs(p) 1
- \cdot ris(p) = n-1-des(p)

- A set partition is a partition of a set into a non-empty set of subsets
- ${n \\ k} \#$ of set partitions of a set *n* elements into *k* non-empty subsets
- Ordered set partition adds an order on the subsets, $k! {n \atop k}$
- (Ordered Bell numbers) $b_n = \sum_{k=1}^n k! {n \atop k}$

1 subset	2 subsets	3 subsets
({1, 2, 3})	({1,2}, {3})	({1}, {2}, {3})
	({3}, {1,2})	({2}, {1}, {3})
	({1,3}, {2})	({1}, {3}, {2})
	({2}, {1,3})	({2}, {3}, {1})
	({2,3}, {1})	({3}, {1}, {2})
	({1}, {2, 3})	({3}, {2}, {1})

Evolution process



Evolution process

Start at step 0 with a leaf; at each step $i \ge 1$ do:

1. Choose one leaf of the so-far built tree.

Increasing Schröder without repetitions

Evolution process

Start at step 0 with a leaf; at each step $i \ge 1$ do:

 $\frac{1}{2}$

- 1. Choose one leaf of the so-far built tree.
- 2. Choose an integer k > 1.

Evolution process



- 1. Choose one leaf of the so-far built tree.
- 2. Choose an integer k > 1.
- 3. Replace the leaf with an internal node labelled *i* and attach to it *k* new leaves.

Evolution process



- 1. Choose one leaf of the so-far built tree.
- 2. Choose an integer k > 1.
- 3. Replace the leaf with an internal node labelled *i* and attach to it *k* new leaves.

Evolution process



- 1. Choose one leaf of the so-far built tree.
- 2. Choose an integer k > 1.
- 3. Replace the leaf with an internal node labelled *i* and attach to it *k* new leaves.

Evolution process

- 1. Choose one leaf of the so-far built tree.
- 2. Choose an integer k > 1.
- 3. Replace the leaf with an internal node labelled *i* and attach to it *k* new leaves.

Evolution process

- 1. Choose one leaf of the so-far built tree.
- 2. Choose an integer k > 1.
- 3. Replace the leaf with an internal node labelled *i* and attach to it *k* new leaves.

Evolution process

- 1. Choose one leaf of the so-far built tree.
- 2. Choose an integer k > 1.
- 3. Replace the leaf with an internal node labelled *i* and attach to it *k* new leaves.

Evolution process



- 1. Choose one leaf of the so-far built tree.
- 2. Choose an integer k > 1.
- 3. Replace the leaf with an internal node labelled *i* and attach to it *k* new leaves.

Evolution process



- 1. Choose one leaf of the so-far built tree.
- 2. Choose an integer k > 1.
- 3. Replace the leaf with an internal node labelled *i* and attach to it *k* new leaves.

Evolution process



- 1. Choose one leaf of the so-far built tree.
- 2. Choose an integer k > 1.
- 3. Replace the leaf with an internal node labelled *i* and attach to it *k* new leaves.

Evolution process



- 1. Choose one leaf of the so-far built tree.
- 2. Choose an integer k > 1.
- 3. Replace the leaf with an internal node labelled *i* and attach to it *k* new leaves.

Evolution process



- 1. Choose one leaf of the so-far built tree.
- 2. Choose an integer k > 1.
- 3. Replace the leaf with an internal node labelled *i* and attach to it *k* new leaves.

Evolution process



- 1. Choose one leaf of the so-far built tree.
- 2. Choose an integer k > 1.
- 3. Replace the leaf with an internal node labelled *i* and attach to it *k* new leaves.

Evolution process



- 1. Choose one leaf of the so-far built tree.
- 2. Choose an integer k > 1.
- 3. Replace the leaf with an internal node labelled *i* and attach to it *k* new leaves.

Enumeration and typical parameters

$$\mathcal{T} = \mathcal{Z} + (\Theta \mathcal{T} \times SEQ_{\geq 1} \mathcal{Z}) \implies T(z) = z + \frac{z^2}{1-z} T'(z).$$

$$t_n = \begin{cases} 1, & n = 1, 2\\ n \ t_{n-1}, & n \ge 3 \end{cases}$$

 $(t_n)_{n\geq 0} = 0, 1, 1, 3, 12, 60, 360, 2520,$ 20160, 181440, 1814400, ...

Theorem: The number of trees of size *n* for all $n \ge 2$ is

 $t_n=\frac{n!}{2}.$

	Mean	Variance	Limit law
Internal nodes	$n - \ln n$	In n	Normal
Number of binary nodes	n — 2 In n	4 ln <i>n</i>	Normal
Number of ternary nodes	ln n		
Depth of the leftmost leaf	ln n	In n	Normal
Height of the tree	$\Theta(\ln n)$		
Degree of the root	2 e - 3	$14 \mathrm{e} - 4 \mathrm{e}^2 - 8$	modified Poisson

Bijection with Permutations

Let \mathcal{PR} set of permutations with elements 1 and 2 in different cycles. Define $\mathcal{N}: \mathcal{PR} \to \mathcal{T}$

- If $\sigma = (1)(2)$ then $\mathcal{N}(\sigma)$ is a cherry labelled 1.
- Else, Let $(i, j) = \sigma^{-1}(n)$ where *n* largest element in σ .
 - If $|c_i| = 1$ then, we set $\mathcal{N}(\sigma)$ to be the tree $\mathcal{N}(\sigma \setminus n)$ and add a rightmost leaf to the last internal node.
 - Else, let $k = c_{i,j-1}$, we set $\mathcal{N}(\sigma)$ to be the tree $\mathcal{N}(\sigma \setminus n)$ and add a binary node labelled ν next integer with 2 leaves. Insert this binary node in $\mathcal{N}(\sigma \setminus n)$ by placing ν in the *k*-th leaf of $\mathcal{N}(\sigma \setminus n)$.

Bijection with Permutations

Let \mathcal{PR} set of permutations with elements 1 and 2 in different cycles. Define $\mathcal{N}: \mathcal{PR} \to \mathcal{T}$

- If $\sigma = (1)(2)$ then $\mathcal{N}(\sigma)$ is a cherry labelled 1.
- Else, Let $(i,j) = \sigma^{-1}(n)$ where *n* largest element in σ .
 - If $|c_i| = 1$ then, we set $\mathcal{N}(\sigma)$ to be the tree $\mathcal{N}(\sigma \setminus n)$ and add a rightmost leaf to the last internal node.
 - Else, let $k = c_{i,j-1}$, we set $\mathcal{N}(\sigma)$ to be the tree $\mathcal{N}(\sigma \setminus n)$ and add a binary node labelled ν next integer with 2 leaves. Insert this binary node in $\mathcal{N}(\sigma \setminus n)$ by placing ν in the *k*-th leaf of $\mathcal{N}(\sigma \setminus n)$.



Bijection with Permutations

Let \mathcal{PR} set of permutations with elements 1 and 2 in different cycles. Define $\mathcal{N}: \mathcal{PR} \to \mathcal{T}$

- If $\sigma = (1)(2)$ then $\mathcal{N}(\sigma)$ is a cherry labelled 1.
- Else, Let $(i, j) = \sigma^{-1}(n)$ where *n* largest element in σ .
 - If $|c_i| = 1$ then, we set $\mathcal{N}(\sigma)$ to be the tree $\mathcal{N}(\sigma \setminus n)$ and add a rightmost leaf to the last internal node.
 - Else, let $k = c_{i,j-1}$, we set $\mathcal{N}(\sigma)$ to be the tree $\mathcal{N}(\sigma \setminus n)$ and add a binary node labelled ν next integer with 2 leaves. Insert this binary node in $\mathcal{N}(\sigma \setminus n)$ by placing ν in the *k*-th leaf of $\mathcal{N}(\sigma \setminus n)$.



Theorem: Let *t* be an increasing Schröder tree with *n* leaves and *k* internal nodes, and let $p = \mathcal{N}^{-1}(t)$ be its corresponding permutation. Let *i* be the number of cycles of *p*, then

k = n + 1 - i.

Evolution process



Evolution process

Start at step 0 with a leaf; at each step $i \ge 1$ do:

1. Choose a non-empty subset *L* of leaves of the so-far built tree.

Evolution process



- 1. Choose a non-empty subset *L* of leaves of the so-far built tree.
- 2. For each $\ell \in L$ choose an integer k > 1.

Evolution process



- 1. Choose a non-empty subset *L* of leaves of the so-far built tree.
- 2. For each $\ell \in L$ choose an integer k > 1.
- 3. Replace ℓ with an internal node labelled *i* and attach to it *k* new leaves.

Evolution process



- 1. Choose a non-empty subset *L* of leaves of the so-far built tree.
- 2. For each $\ell \in L$ choose an integer k > 1.
- 3. Replace ℓ with an internal node labelled *i* and attach to it *k* new leaves.

Evolution process



- 1. Choose a non-empty subset *L* of leaves of the so-far built tree.
- 2. For each $\ell \in L$ choose an integer k > 1.
- Replace ℓ with an internal node labelled i and attach to it k new leaves.

Evolution process



- 1. Choose a non-empty subset *L* of leaves of the so-far built tree.
- 2. For each $\ell \in L$ choose an integer k > 1.
- Replace ℓ with an internal node labelled i and attach to it k new leaves.

Evolution process



- 1. Choose a non-empty subset *L* of leaves of the so-far built tree.
- 2. For each $\ell \in L$ choose an integer k > 1.
- 3. Replace ℓ with an internal node labelled *i* and attach to it *k* new leaves.

Evolution process



- 1. Choose a non-empty subset *L* of leaves of the so-far built tree.
- 2. For each $\ell \in L$ choose an integer k > 1.
- 3. Replace ℓ with an internal node labelled *i* and attach to it *k* new leaves.

Evolution process



- 1. Choose a non-empty subset *L* of leaves of the so-far built tree.
- 2. For each $\ell \in L$ choose an integer k > 1.
- Replace ℓ with an internal node labelled i and attach to it k new leaves.

Evolution process



- 1. Choose a non-empty subset *L* of leaves of the so-far built tree.
- 2. For each $\ell \in L$ choose an integer k > 1.
- 3. Replace ℓ with an internal node labelled *i* and attach to it *k* new leaves.

Evolution process



- 1. Choose a non-empty subset *L* of leaves of the so-far built tree.
- 2. For each $\ell \in L$ choose an integer k > 1.
- 3. Replace ℓ with an internal node labelled *i* and attach to it *k* new leaves.

Evolution process



- 1. Choose a non-empty subset *L* of leaves of the so-far built tree.
- 2. For each $\ell \in L$ choose an integer k > 1.
- Replace ℓ with an internal node labelled i and attach to it k new leaves.

Enumeration and typical parameters

$$G(z) = z + G\left(\frac{z^2}{1-z} + z\right) - G(z).$$
$$g_n = \begin{cases} 1 & \text{if } n = 1\\ \sum_{k=1}^{n-1} {n-1 \choose k-1} g_k. & \text{otherwise.} \end{cases}$$

 $(g_n)_{n\geq 0} = 0, 1, 1, 3, 13, 75, 541, 4683, \ldots$

Theorem: The number of trees of g_n :

$$g_n = \mathop{\sim}_{n \to \infty} \frac{(n-1)!}{2(\ln 2)^n}.$$



	Mean	Variance	Limit law
Internal nodes	$n - \ln 2 \ln n$		
Distinct labels	$\frac{1}{2 \ln 2} n$	$\frac{(1-\ln 2)}{(2\ln 2)^2}n$	Normal
Degree of the root	2 ln 2 + 1	−2 ln 2 (ln 2 − 1)	(shifted) zero-truncated Poisson
Depth of the leftmost leaf	In n	In n	Normal

Bijection with Ordered set partitions

An ordered partition is denoted $p = (p_1, \ldots, p_i)$. We define M'

- If *p* contains one subset, associate a tree with one internal node labelled by 1 to which are attached $|p_1| + 1$ leaves.
- Else if *p* has *i* subsets, construct the tree *t* on (p_1, \ldots, p_{i-1}) . Denote by p'_1, \ldots, p'_i the ordered subsets of the renormalization of (p_1, \ldots, p_i) , i.e. $\operatorname{norm}((p_1, \ldots, p_i)) = (p'_1, \ldots, p'_i)$. We denote by r_1, \ldots, r_j the runs of p'_i , i.e. $\operatorname{runs}(p'_i) = (r_1, \ldots, r_j)$, then for *k* from 1 to *j*, take the leaf whose index is the first element of r_k and replace it with an internal node with label *i* attached to $|r_k| + 1$ leaves.

Bijection with Ordered set partitions

An ordered partition is denoted $p = (p_1, \ldots, p_i)$. We define M'

- If *p* contains one subset, associate a tree with one internal node labelled by 1 to which are attached $|p_1| + 1$ leaves.
- Else if *p* has *i* subsets, construct the tree *t* on (p_1, \ldots, p_{i-1}) . Denote by p'_1, \ldots, p'_i the ordered subsets of the renormalization of (p_1, \ldots, p_i) , i.e. $\operatorname{norm}((p_1, \ldots, p_i)) = (p'_1, \ldots, p'_i)$. We denote by r_1, \ldots, r_j the runs of p'_i i.e. $\operatorname{runs}(p'_i) = (r_1, \ldots, r_j)$, then for *k* from 1 to *j*, take the leaf whose index is the first element of r_k and replace it with an internal node with label *i* attached to $|r_k| + 1$ leaves.



Bijection with Ordered set partitions

An ordered partition is denoted $p = (p_1, \ldots, p_i)$. We define M'

- If *p* contains one subset, associate a tree with one internal node labelled by 1 to which are attached $|p_1| + 1$ leaves.
- Else if *p* has *i* subsets, construct the tree *t* on (p_1, \ldots, p_{i-1}) . Denote by p'_1, \ldots, p'_i the ordered subsets of the renormalization of (p_1, \ldots, p_i) , i.e. $\operatorname{norm}((p_1, \ldots, p_i)) = (p'_1, \ldots, p'_i)$. We denote by r_1, \ldots, r_j the runs of p'_i , i.e. $\operatorname{runs}(p'_i) = (r_1, \ldots, r_j)$, then for *k* from 1 to *j*, take the leaf whose index is the first element of r_k and replace it with an internal node with label *i* attached to $|r_k| + 1$ leaves.

$$\frac{\operatorname{norm}((\{3,4\})) = (\{1,2\}) \xrightarrow{\texttt{M}'} \overset{(1)}{\bigwedge}}{\operatorname{norm}((\{3,4\},\{1,5,7\})) = (\{2,3\},\{1,4,5\}) \xrightarrow{\texttt{M}'} \overset{(1)}{\bigwedge} \overset{(1)}{\overset{(1)}{} \overset{(1)}{\overset{(1)}{} \overset{(1)}{\overset{(1)}{} \overset{(1)}{\overset{(1)}{} \overset{(1)}{\overset{(1)}{} \overset{(1)}{\overset{(1)}{} \overset{(1)}{\overset{(1)}{} \overset{(1)}{\overset{(1)}{} \overset{(1)}{\overset{(1)}{} \overset{(1)}{\overset{(1)}{\overset{(1)}{} \overset{(1)}{\overset{(1)}{} \overset{(1)}{\overset$$

Theorem: The number of trees of size *n* with *k* distinct labels:

$$g_{n,k} = k! \begin{Bmatrix} n-1 \\ k \end{Bmatrix} \implies g_n = \sum_{k=0}^{n-1} k! \begin{Bmatrix} n-1 \\ k \end{Bmatrix}.$$

- ris((4,6,1,2,5,3)) = 3
- Distinguish rises
- 4 6 1 2 5 3
 - Selected consec. rises merge in a set

- ris((4,6,1,2,5,3)) = 3
- Distinguish rises
- 4 6 1 2 5 3
 - Selected consec. rises merge in a set ({4}, {6}, {1, 2, 5}, {3})

- ris((4,6,1,2,5,3)) = 3
- Distinguish rises
- 4 6 1 2 5 3
 - Selected consec. rises merge in a set ({4,6}, {1,2}, {5}, {3})

- ris((4, 6, 1, 2, 5, 3)) = 3
- Distinguish rises
- 4 6 1 2 5 3
 - Selected consec. rises merge in a set ({4,6}, {1,2}, {5}, {3})

$(\{4\},\{6\},\{1\},\{2\},\{5\},\{3\})$	$(\{4,6\},\{1\},\{2\},\{5\},\{3\})$
$({4}, {6}, {1, 2}, {5}, {3})$	$(\{4\},\{6\},\{1\},\{2,5\},\{3\})$
$(\{4,6\},\{1,2\},\{5\},\{3\})$	({4,6}, {1}, {2,5}, {3})
$({4}, {6}, {1, 2, 5}, {3})$	({4, 6}, {1, 2, 5}, {3})

 \cdot (Velleman and Call 95) showed that

$$b_n = \sum_{k=0}^{n-1} 2^k \left< {n \atop k} \right>.$$

- ris((4, 6, 1, 2, 5, 3)) = 3
- Distinguish rises
- 4 6 1 2 5 3
 - Selected consec. rises merge in a set ({4,6}, {1,2}, {5}, {3})

$({4}, {6}, {1}, {2}, {5}, {3})$	$(\{4,6\},\{1\},\{2\},\{5\},\{3\})$
$({4}, {6}, {1, 2}, {5}, {3})$	$(\{4\},\{6\},\{1\},\{2,5\},\{3\})$
$(\{4,6\},\{1,2\},\{5\},\{3\})$	$({4,6}, {1}, {2,5}, {3})$
({4}, {6}, {1, 2, 5}, {3})	({4,6}, {1,2,5}, {3})

 \cdot (Velleman and Call 95) showed that

$$b_n = \sum_{k=0}^{n-1} 2^k \left< {n \atop k} \right>.$$

• A tree *t* (without repetitions) of size *n* with *k* descents, makes

$$2^{n-k-1}+2^k$$
,

trees with repetitions of size n-1

• Open question. Give an interpretation on the tree level

Increasing Schröder with repetitions

Conclusion

Increasing versions of classical Schröder trees



- Bijection with classes of Permutations
- Cycles relates to internal nodes

- Bijection with Set partitions
- Subset number relates to the max label
- Step toward a more general framework of increasing label repetitions