



**HAL**  
open science

## **Strict monotonic trees arising from evolutionary processes: combinatorial and probabilistic study**

Olivier Bodini, Antoine Genitrini, Cécile Mailler, Mehdi Naima

### ► **To cite this version:**

Olivier Bodini, Antoine Genitrini, Cécile Mailler, Mehdi Naima. Strict monotonic trees arising from evolutionary processes: combinatorial and probabilistic study. *Advances in Applied Mathematics*, 2022, 133, pp.102284. <10.1016/j.aam.2021.102284>. <hal-02865198v2>

**HAL Id: hal-02865198**

**<https://hal.sorbonne-universite.fr/hal-02865198v2>**

Submitted on 8 Oct 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

**STRICT MONOTONIC TREES ARISING FROM  
EVOLUTIONARY PROCESSES:  
COMBINATORIAL AND PROBABILISTIC STUDY**

OLIVIER BODINI, ANTOINE GENITRINI, CÉCILE MAILLER, AND MEHDI NAIMA

ABSTRACT. In this paper we study two models of labelled random trees that generalise the original unlabelled Schröder tree. Our new models can be seen as models for phylogenetic trees in which nodes represent species and labels encode the order of appearance of these species, and thus the chronology of evolution. One important feature of our trees is that they can be generated efficiently thanks to a dynamical, recursive construction. Our first model is an increasing tree in the classical sense (labels increase along each branch of the tree and each label appears only once). To better model phylogenetic trees, we relax the rules of labelling by allowing repetitions in the second model.

For each of the two models, we provide asymptotic theorems for different characteristics of the tree (e.g. degree of the root, degree distribution, height, etc), thus giving extensive information about the typical shapes of these trees. We also provide efficient algorithms to generate large trees efficiently in the two models. The proofs are based on a combination of analytic combinatorics, probabilistic methods, and bijective methods (we exhibit bijections between our models and well-known models of the literature such as permutations and Stirling numbers of both kinds).

It turns out that even though our models are labelled, they can be specified simply in the world of ordinary generating functions. However, the resulting generating functions will be formal. Then, by applying Borel transforms the models will be amenable to techniques of analytic combinatorics.

KEYWORDS. Evolution process; Increasing trees; Monotonic trees; Analytic Combinatorics; Uniform sampling.

---

*Date:* October 5, 2021.

This research is partially supported by the ANR project *MetACOnc*, ANR-15-CE40-0014. C. Mailler acknowledges EPSRC for support through the fellowship EP/R022186/1.

## CONTENTS

1. Introduction	3
2. Increasing Schröder trees	6
2.1. The model and its context	6
2.2. Overview of the main results	9
2.3. Exact enumeration and relationship with permutations	10
2.4. Analysis of typical parameters	11
2.5. Bijection with permutations	17
2.6. Uniform random sampling	18
2.7. Analysis of the height of a typical increasing Schröder tree	20
3. Strict monotonic Schröder trees	23
3.1. The model and its context	23
3.2. Overview of the main results	24
3.3. Enumeration and relationship with ordered Bell numbers	25
3.4. Bijection with ordered Bell numbers	27
3.5. Analysis of typical parameters	28
3.6. Uniform random sampling	32
References	37

## 1. INTRODUCTION

The aim of this paper is to study combinatorial models for phylogenetic trees: the main idea is to add node labels in order to encode chronology in the classical model introduced by Schröder trees in 1870 in the seminal paper *Vier Combinatorische Probleme* [Sch70]. In this paper (see the fourth problem), Schröder introduces a simple model of *phylogenetic tree model*, and enumerates this class of trees by their number of leaves.

In biology, a phylogenetic tree is a classical tool to represent the evolutionary relationship among species. At each branching node of the tree, the descendant species from distinct branches have distinguished themselves in some manner and are no more dependent: the past is shared but the futures are independent.

The main limitation of Schröder’s model of phylogenetic trees is that it does not take into account the chronology between the different branching nodes. Since then, probabilistic approaches have been developed to consider this chronology: in particular in the context of binary trees, one can mention, e.g., the stochastic model of Yule [Yul25] and its generalization by Aldous [Ald96].

However, to the best of our knowledge, there seems to have been no attempt to combinatorially enrich Schröder’s original model in order to encode the chronology of evolution (except the short version [BGN19] of this paper). Motivated by the fact that models with restricted arities appear in the biology literature under the name of “ranked phylogenetic trees” (see, e.g., [SDH<sup>+</sup>04]), we aim at relaxing these restrictions and study models with unrestricted arities. To do so, we consider labelled versions of Schröder’s tree, where the labels represent the order at which branchings occur. In Figure 1 we have represented on the left hand-side a classical Schröder trees of size 50 (i.e. with 50 leaves), and, on the right hand-side, a labelled version of the same tree: time is on the vertical axis, from top to bottom, and a node of label  $x$  is placed at time  $x$  (the horizontal placement is irrelevant).

The two models of trees we study are enumerated by their number of leaves since leaves represent the number of current species. In this paper we do not distinguish between the leaves and therefore they are unlabelled. The justification for this is that we consider species to have the same evolutionary power and are thus undistinguishable. Additionally, the trees are ordered which is not the case in usual phylogenetic tree models. However, this gives a first approach to the study of such trees since the equivalent non-plane models (i.e. where the order of siblings is irrelevant) of these trees require more involved technical tools. We aim in the near future to extend this study to such non-plane models.

**Discussion of related models:** Increasing trees are classical in the literature: for example, Bergeron, Flajolet and Salvy [BFS92] study several families of increasingly-labelled trees, and, to do so, they develop some tools that are now classical in *analytic combinatorics*. As an example, one of these classical tools is the integration of the Greene operators. We refer the reader to [Drm09] where more recent results on various families of increasing trees and the analytic combinatorics methods to quantitatively study them are surveyed.

Beyond their combinatorial description, increasing trees can often be described as the result of a dynamic construction: the nodes are added one by one at successive integer-times in the tree (their labels being the time at which they are added). This description sometimes allows to apply probabilistic method to prove theorems

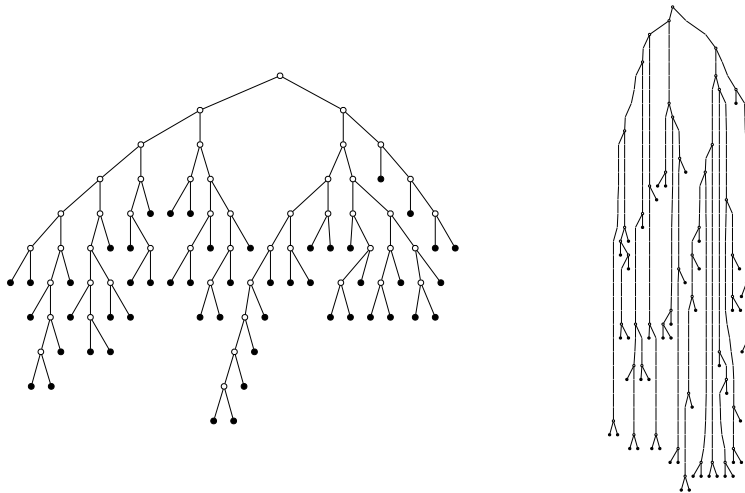


FIGURE 1. A Schröder tree: without chronological evolution (on the left-hand side), and with chronological evolution (on the right-hand side): the label of a node is represented as the distance from this node to the root.

about some characteristics such as the height of the tree, and it also often gives a very efficient way to generate large trees from the considered class using simple, iterative and local rules.

We illustrate this dynamical evolution on the simple case of *recursive trees*. This simple model was originally designed as a simple model for the spread of epidemics [Moo74]. Combinatorially, a recursive tree is a rooted non-plane (i.e. the order of siblings is irrelevant) tree whose nodes are labelled from 1 to the number of nodes in such a way that each label appears exactly once, and the labels increase along all branches. We denote by  $\mathcal{R}_n$  the class of all  $n$ -node recursive trees. Now, consider a sequence of random trees  $(t_n)_{n \geq 1}$  built recursively as follows:  $t_0$  has only one node, labelled by 1. Given  $t_{n-1}$ , attach a new child labelled by  $n$  to a node picked uniformly at random among the  $n-1$  nodes of  $t_{n-1}$ . Then, it is known that for all  $n \geq 1$ ,  $t_n$  is a tree taken uniformly at random in  $\mathcal{R}_n$ , the set of all  $n$ -node recursive trees. Both analytic combinatorics and probabilistic methods, as well as a bijection with permutations, have been used to understand the average typical shape of a large recursive tree: it is known that the degree of the root grows on average as  $\ln n$  (see [Drm09, Sec. 6.1]), the height as  $c \ln n$  (for an explicit constant  $c$  – see [Pit94]), the proportion of nodes of arity  $k \geq 0$  converges to  $2^{-k}$  (see [Drm09, Th. 6.8]). Although our two models of increasing Schröder trees are more involved, our proofs rely on the same three methods used in the recursive tree literature: analytic combinatorics, a dynamical evolution, probabilistic methods, and bijections with classes of permutations.

**Our main contributions:** Although, as mentioned above, many variations of the recursive tree have been studied, this paper (together with its short version [BGN19]) contains the first study of increasing versions of the classical model of Schröder.

Since the classical tools for specifying tree models in analytic combinatorics are not giving here usable generating functions, we aim at defining an evolution process that associates, to each Schröder tree, an evolution represented by an increasing labelling of its internal nodes. Furthermore, in our second model, we relax the labelling constraints by allowing repetitions of labels. In the dynamical construction of the trees, allowing repetition of labels means allowing the addition of several nodes at once in the tree. Our generalisations can be seen as natural discrete-time versions of the classical probabilistic model of Yule trees (see, e.g., [SM01]); recall that, in the Yule tree, the time between two branchings are exponentially distributed.

This work is part of a long-term over-arching project, in which we aim at relaxing the classical rules of increasing labelling (described in, e.g., [BFS92]), by, for example, allowing labels to appear more than once in the tree. The following papers are part of this strand of research: [BGGW20, BGNS20] introduce and study models of labelled trees with less-constrained increasing labelling rules; [BDF<sup>+</sup>16] focuses on increasingly-labelled “diamonds”; and [GGKW20] analyses at a compacted structure that specifies classes of directed acyclic graphs.

In this paper, we introduce two new different models of Schröder trees with chronological evolution: the *increasing Schröder trees* and the *strict monotonic Schröder trees*. One important feature of these models is that they can both be simulated efficiently. The models are based on some *increasing labelling* of Schröder trees, and the repetition of labels is allowed in the second model. For the two models, we prove asymptotic results about important characteristics of a typical large tree of this class (e.g. root distribution, number of nodes of arity 2, 3, etc, height of the tree, etc – see Table 1 where our main results are summarised), and design an algorithm that generates a large tree taken uniformly at random among all trees of a given size in the class. The quantitative analysis of the models and the design of the random samplers rely on a combination of analytic combinatorics methods (see [FS09] for a survey), probabilistic methods (in particular methods developed by Devroye [Dev90] to study the height of *split trees*), and bijective methods (we exhibit bijections between our classes of trees and classes of permutations, these are then useful for the analysis of different characteristics and for the design of the generation algorithms). In particular, we exhibit interesting relations between Stirling numbers and parameters on trees such as the number of the labellings of nodes, the number of internal nodes, and the depth of a leaf.

**Generic approach highlighted in the paper:** Similarly to the recursive tree, the two models have a *generic constrained evolution process*. The specificity of each model is induced by small changes of the evolution process: we give here a generic, non precise description of the evolution process, details specific to each family of trees will be given in each section:

- *Start with a single (unlabelled) leaf;*
- *Iterate the following process: at step  $\ell$  (for  $\ell \geq 1$ ), select a subset of leaves and replace each selected leaf by an internal node with label  $\ell$  attached to an arbitrary sequence of leaves.*

Note that the increasing labelling corresponds to the chronology of the construction of the tree: internal nodes labelled by an integer  $\ell$  were added at time  $\ell$ . Our models differ from each other by different constraints on the set of selected leaves: in our first model, this subset is always of size 1, while it can be bigger in the second

model. Importantly, our models can be seen as phylogenetic trees of  $n$  species ( $n$  being the number of leaves): the labels of internal nodes stand for the times at which different branches of the phylogenetic trees split.

	Number of trees	Distinct labels	Internal nodes	Height
Increasing Schröder trees	$n!/2$	$n - \ln n$	$n - \ln n$	$\Theta(\ln n)$
Strict monotonic Schröder trees	$(n-1)/(2(\ln 2)^n)$	$\frac{1}{2\ln 2} n$	$n - \ln 2 \ln n$	

TABLE 1. Main analytic results of the paper: behaviour of the characteristics of a large typical tree of each of the two classes of labelled Schröder trees. The parameter  $n$  stands for the size of the trees (i.e. their number of leaves) and the results represent the average values of the parameters when  $n \rightarrow +\infty$ .

**Plan of the paper:** Each of the two main sections (Sections 2 and 3) is dedicated to one of our models of labelled Schröder trees. The organisation inside each section is similar: after defining the model we show theorems about different characteristics of the trees using analytic combinatorics and bijective methods. We then exhibit the associated dynamical evolution that generates the considered class of trees, and use this evolution process to (a) design an efficient random sampler for this class of trees and (b) in the case of our first model, prove some probabilistic results about the height of a typical large tree from this class.

## 2. INCREASING SCHRÖDER TREES

The first model we are interested in is a generalisation of the Schröder tree, a classical combinatorial structure that was originally introduced in the context of phylogenetics [Sch70]. Our generalisation consists in labelling the internal nodes of a Schröder tree – denote by  $\ell$  their number – with the integers  $\{1, \dots, \ell\}$  with the constraints that each label appears exactly once and a node’s label is larger than the label of its parent; such a labelling of a tree is called “increasing”, we call such a constrained-labelled Schröder tree an *increasing Schröder tree*. In the tree seen as an evolutionary process, the labels can be interpreted as the order of appearance of the different nodes (which, for example, stand for different species). Several classes of increasingly-labelled trees have already been studied in the literature using *analytic combinatorics* [FS09] methods, but these methods applied to the Schröder tree would raise new technical problems. The novelty of our approach is to use a dynamical description of the increasing Schröder tree inspired by its evolutionary interpretation; this allows us to give the first analytical results about this combinatorial structure.

**2.1. The model and its context.** In this paper, we define rooted trees as genealogical structures: the root is the unique common ancestor of all nodes of the tree, each node except the root has exactly one parent (the root has no parent), nodes that have no children are called *leaves*, nodes that have at least one child are called *internal nodes*. The *arity* of a node is its number of children. We say that a tree is *plane* if siblings (nodes that have the same parent) are ordered.

**Definition 2.1.1** (see [FS09, p. 69]). *A Schröder tree is a rooted plane tree whose internal nodes all have arity at least 2. The size of a Schröder tree is its number of leaves.*

Note that a Schröder tree is an unlabelled combinatorial structure (neither the leaves nor the internal nodes are labelled). In the context of analytic combinatorics the combinatorial class  $\mathcal{S}$  of Schröder trees is thus specified as

$$\mathcal{S} = \mathcal{Z} \cup \text{SEQ}_{\geq 2} \mathcal{S}. \tag{1}$$

Its combinatorial meaning is direct in the context of decomposable objects (see Flajolet and Sedgewick [FS09] for a detailed introduction for the method of combinatorial specification): An object from  $\mathcal{S}$  is either a leaf (represented by the single atom  $\mathcal{Z}$ , of size 1), or it is composed of an internal node, parent of a sequence of at least two elements from  $\mathcal{S}$ . Note that, in the specification, the internal nodes are omitted (because they are of size 0): the expression  $\text{SEQ}_{\geq 2} \mathcal{S}$  is a abbreviation of  $\mathcal{E} \times \text{SEQ}_{\geq 2} \mathcal{S}$  where  $\mathcal{E}$  stands for an atom of size 0 and  $\text{SEQ}_{\geq 2} \mathcal{S}$  is a sequence of at least two elements from  $\mathcal{S}$ .

Once the combinatorial specification is given, the *symbolic method* (see [FS09] for details), translates automatically the equation specifying the objects into a functional equation satisfied by the (ordinary) generating functions associated to the structures. The generating function of  $\mathcal{S}$  is defined as the formal series  $S(z) = \sum_{n \geq 1} s_n z^n$  where  $s_n$  is the number of Schröder trees of size  $n$  (i.e. with  $n$  leaves). Using the symbolic method on Equation (1), we get that

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

An elementary iteration allows us to extract the first coefficients of the sequence  $(s_n)_{n \in \mathbb{N}}$ :

$$(0, 1, 1, 3, 11, 45, 197, 903, 4279, 20793, 103049, 518859, 2646723, 13648869, \dots).$$

Equation (2) implies that the generating function  $S$  is algebraic and in fact

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

This is sufficient to get the following asymptotic equivalent of  $s_n$  when  $n$  tends to infinity:

$$s_n = \frac{\sqrt{3\sqrt{2} - 4}}{4\sqrt{\pi}} n^{-3/2} (3 - 2\sqrt{2})^{-n} (1 + \mathcal{O}(1/n)).$$

We refer the reader to [FS09, page 69] for a more detailed analysis of this generating function  $S$ .

In the rest of the section we are interested in an increasingly-labelled variation of Schröder trees.

**Definition 2.1.2.** *An increasing Schröder tree has a Schröder tree structure and its internal nodes are labelled with the integers between 1 and  $\ell$  (where  $\ell$  is the number of internal nodes) in such a way that each label appears exactly once and each sequence of labels in the paths from the root to any leaf is (strictly) increasing.*

Increasing trees can, to a certain extent, be specified using the Greene operator  $\square_\star$  (see, for example, [FS09, page 139]), and the specification can then be translated into an equation satisfied by the exponential generating function of the increasing

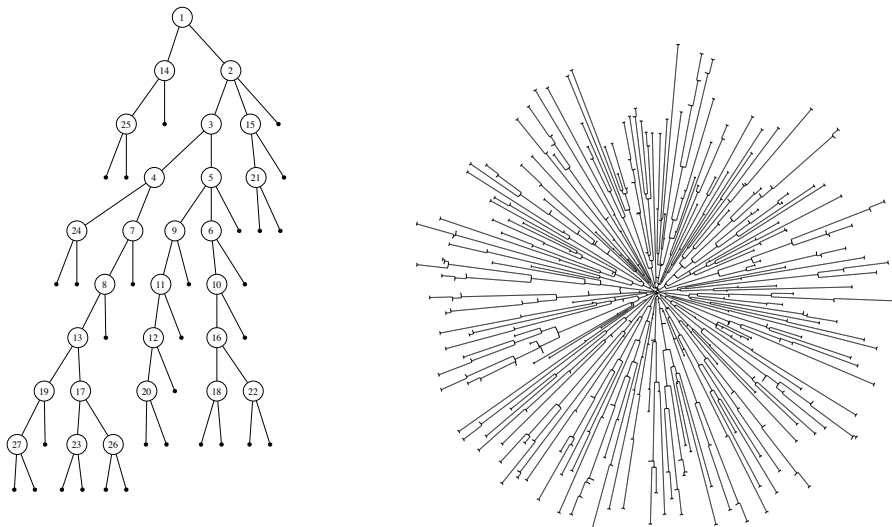


FIGURE 2. Two increasing Schröder trees

tree class. Since in our context the size of a tree is the number of its leaves while only internal nodes are labelled, we need to introduce a second variable  $u$  to mark the internal nodes. We let  $s_{n,\ell}$  denote the number of increasing Schröder trees with  $n$  leaves and  $\ell$  internal nodes. Following standard methods in combinatorics we define a generating function that is ordinary for the leaf marks and exponential for the internal node marks: we set  $S^*(z, u) = \sum_{n,\ell} s_{n,\ell} z^n u^\ell / \ell!$ . The specification of this combinatorial class is

$$\mathcal{S}^* = \mathcal{Z} \cup \mathcal{U}^\square \star \text{SEQ}_{\geq 2} \mathcal{S}^*.$$

Using the symbolic method [FS09], we obtain the following equation satisfied by  $S^*(z, u)$ :

$$S^*(z, u) = z + \int_{v=0}^u \frac{S^*(z, v)^2}{1 - S^*(z, v)} dv.$$

Although this integral equation could be analysed further in order to get information about increasing Schröder trees, this analysis would be very cumbersome; a better approach is to see the Schröder tree as the result of an evolutionary process. Another advantage of this new approach is that it extends to other families of labelled Schröder trees for which there seems to be no (classical) specification, even using the Greene operator: one such example is the family of *strict monotonic Schröder trees* studied in Section 3.

In Figure 2 we have represented two increasing Schröder trees: both are generated uniformly at random among all increasing Schröder trees of the same size: size 30 on the left, size 500 on the right. The left-hand-side tree has 27 internal nodes (and 30 leaves). It is the same tree as the one represented in Figure 1, where its chronological evolution is represented on the right-hand side: the internal node labelled by  $\ell$  is displayed on level  $\ell - 1$  (i.e. at distance  $\ell - 1$  from the root on the vertical axis), for all  $\ell \in \{1, \dots, 27\}$ . The right-hand-side one is drawn using

a circular representation, which is often used for phylogenetic trees: the labels are omitted but as in Figure 1, the length of an edge is proportional to the difference of the labels of the two nodes it connects. This right-hand-side tree has 492 internal nodes (and 500 leaves).

We introduce an evolution process generating increasing Schröder trees:

- Start with a single (unlabelled) leaf;
- Iterate the following process: at step  $\ell$  (for  $\ell \geq 1$ ), select one leaf and replace it by an internal node with label  $\ell$  attached to an arbitrary sequence of new leaves.

By taking all trees of the same size together, we obtain the following recurrence, enumerating increasing Schröder trees by size: if, for all  $n \geq 1$ ,  $t_n$  is the number of  $n$ -leaf Schröder trees, then  $t_1 = 1$  and, for all  $n \geq 2$ ,

$$t_n = \sum_{\ell=1}^{n-1} \ell t_\ell. \quad (3)$$

To make sure that this algorithm generates all increasing Schröder trees and generates each tree exactly “once”; we define this evolution process more rigorously as follows: The process takes as input two sequences of integers  $(d_\ell)_{\ell \geq 1}$  and  $(u_\ell)_{\ell \geq 1}$  such that  $u_1 = 1$  and for all  $\ell \geq 1$ ,  $d_\ell \geq 2$  and  $1 \leq u_{\ell+1} \leq \sum_{i=1}^{\ell} d_i - (\ell - 1)$  and gives as an output a sequence  $(\tau_\ell)_{\ell \geq 0}$  of  $\ell$ -internal-node increasing Schröder trees. The process is defined inductively as follows:

- Tree  $\tau_0$  is the 1-node tree, without any internal node.
- Given  $\tau_\ell$ , we define  $\tau_{\ell+1}$  as follows: we number the leaves of  $\tau_\ell$  in the depth-first order (the choice of the ordering does not matter) from 1 to  $\sum_{i=1}^{\ell} d_i - (\ell - 1)$ , and replace leaf number  $u_{\ell+1}$  by an internal node labelled by  $\ell + 1$  to which  $d_{\ell+1}$  leaves are attached.

Note that, by construction,  $\tau_\ell$  is an increasing Schröder tree with  $\ell$  internal nodes for all  $\ell \geq 0$ , and the label of a node corresponds to the time in the evolution process when this node became an internal node. In other words, the increasing labelling corresponds to the chronology of the evolution process. Finally, note that the evolution process indeed defines a bijection between the set of increasing Schröder trees having  $p$  internal nodes and the set of all sequences  $(d_\ell, u_\ell)_{1 \leq \ell \leq p}$  such that  $u_1 = 1$ , and for all  $1 \leq \ell \leq p$ ,  $d_\ell \geq 2$  and  $1 \leq u_{\ell+1} \leq \sum_{i=1}^{\ell} d_i - (\ell - 1)$ .

Recall that we define the size of a Schröder tree to be its number of leaves. It is important to note that, because a Schröder tree with  $n - 1$  internal nodes has at least  $n$  leaves, the evolution process defines a bijection between the set of all  $n$ -leaf Schröder trees and the set of all sequences  $(d_\ell^{(n)}, u_\ell^{(n)})_{1 \leq \ell < n}$  such that for all  $1 \leq \ell < n$ ,  $u_1^{(n)} = 1$ ,  $d_\ell^{(n)} \geq 2$ ,  $1 \leq u_{\ell+1}^{(n)} \leq \sum_{i=1}^{\ell} d_i^{(n)} - (\ell - 1)$ , and  $\sum_{i=1}^{\ell} d_i^{(n)} - (\ell - 1) = n$ .

**2.2. Overview of the main results.** We start by solving the enumeration problem in Section 2.3. Then we turn to the analysis of typical parameters in Section 2.4 with the results summarised in Table 2. From the enumeration a relationship with permutations is unveiled. Therefore, in Section 2.5 we will characterise this relationship through a bijection. The bijection will serve as a basis to construct an efficient uniform sampler for this classes that we give in Section 2.6. We conclude with Section 2.7, where we study the average height of these trees; this result is dealt

with in an independent subsection because its proof is different from the proofs of all analytical results of Section 2.4 and relies on the uniform sampling method of Section 2.6.

	Mean	Variance	Limit law
Internal nodes	$n - \ln n$	$\ln n$	Normal
Degree of the root	$2e - 3$	0.4997	modified Poisson
Number of binary nodes	$n - 2 \ln 2$	$4 \ln n$	Normal
Number of ternary nodes	$\ln n$		
Height of the tree	$\Theta(\ln n)$		

TABLE 2. Summary of the main results on parameters of *Increasing Schröder trees*

**2.3. Exact enumeration and relationship with permutations.** Let  $\mathcal{T}$  denote the class of increasing Schröder trees. Using the evolution process, we get the following specification for  $\mathcal{T}$ :

$$\mathcal{T} = \mathcal{Z} \cup (\Theta\mathcal{T} \times \text{SEQ}_{\geq 1} \mathcal{Z}). \quad (4)$$

In this specification,  $\mathcal{Z}$  stands for the leaves, and the operator  $\Theta$  is the classical pointing operator (see [FS09, page 86] for details). The specification is a direct rewriting of the evolution process: a tree is either of size 1 ( $\mathcal{Z}$ ), or it has been built by pointing at a leaf in a smaller tree ( $\Theta\mathcal{T}$ ) and replacing it by a sequence of at least two leaves. Although the latter sequence is of length at least 2, we use the operator  $\text{SEQ}_{\geq 1}(\mathcal{Z})$  instead of  $\text{SEQ}_{\geq 2}(\mathcal{Z})$  because the leaf that was pointed at is reused as the leftmost child of the new internal node.

The symbolic method translates this specification into a functional equation satisfied by the generating function associated to the combinatorial class of increasing Schröder trees. Note that although the increasing Schröder trees are labelled, this labelling does not appear in the specification, i.e. it is possible to work with ordinary generating functions (as opposed to exponential generating functions). This is because the size of an increasing Schröder tree is its number of leaves, and the leaves are not labelled. We define the *ordinary generating function* associated to  $\mathcal{T}$  by  $T(z) = \sum_{n \geq 1} t_n z^n$ , where  $t_n$  is the number of increasing Schröder trees of size  $n$ . Using the symbolic method (in particular, pointing at a leaf translates into a differential operator), we get

$$T(z) = z + \frac{z^2}{1-z} T'(z). \quad (5)$$

From this equation we get that for all  $n \geq 3$ ,  $t_n = n t_{n-1}$ ; which is also a consequence of Equation (3). Using the fact that  $t_1 = t_2 = 1$ , we get that  $t_n = n!/2$  for all  $n \geq 2$  (this sequence  $(t_n)_n$  appears under the reference [OEIS A001710](#)<sup>1</sup>). Note that the radius of convergence of the ordinary generating series  $T(z)$  is 0; this series is thus purely formal.

<sup>1</sup>Throughout this paper, a reference [OEIS A...](#) points to Sloane's Online Encyclopedia of Integer Sequences [www.oeis.org](http://www.oeis.org).

**2.4. Analysis of typical parameters.** In this section, our aim is to describe the *shape* of a *typical* increasing Schröder tree, i.e. a tree taken uniformly at random among all increasing Schröder trees of a fixed size. To get information about this *shape*, we focus on four characteristics of the tree: the number of internal nodes, the arity of the root, the number of leaves that are children of the root, and the number of binary nodes (node of arity 2). We show asymptotic theorems for these characteristics in a typical increasing Schröder tree when the size goes to infinity.

**2.4.1. Quantitative analysis of the number of iteration steps.** In this section, we show that although an increasing Schröder tree of size  $n$  can have between 1 and  $n - 1$  internal nodes, it typically has of order  $n - \ln n$  internal nodes. This result is particularly interesting to analyse the complexity of the evolutionary process: this means that, on average, this evolutionary process takes of order  $n - \ln n$  iteration steps to generate a typical increasing Schröder tree of size  $n$ . In fact, our result is stronger than just finding an equivalent for the average number of iterations since we prove a central limit theorem for this quantity. To complete the picture we also quantify the average number of nodes of a fixed degree. We will show that the average number of binary nodes in a typical tree is  $n - 2 \ln n$ , the number of ternary nodes is  $\ln n$  and higher arity nodes have a constant mean.

**Theorem 2.4.1.** *For all  $n \geq 1$ , we denote by  $X_n$  the number of internal nodes in a tree taken uniformly at random among all increasing Schröder trees of size  $n$ . Then, asymptotically when  $n$  tends to infinity,  $\mathbb{E}_{\mathcal{T}_n}[X_n] \sim n - \ln n$ ,  $\mathbb{V}_{\mathcal{T}_n}[X_n] \sim \ln n$ , and*

$$\frac{X_n - (n - \ln n)}{\sqrt{\ln n}} \xrightarrow{d} \mathcal{N}(0, 1) \quad \text{in distribution.}$$

To prove this theorem, we enrich the specification (4) with an additional parameter  $\mathcal{U}$  marking the internal nodes:

$$\mathcal{T} = \mathcal{Z} \cup (\mathcal{U} \times \Theta_{\mathcal{Z}}\mathcal{T} \times \text{SEQ}_{\geq 1} \mathcal{Z}),$$

where the operator  $\Theta_{\mathcal{Z}}$  consists in pointing at an element marked by  $\mathcal{Z}$ . Note that here we do not use the Greene operator: the increasing labelling is a consequence of the iterative process. Using the symbolic method, this implies that, if  $t_{n,k}$  is the number of increasing Schröder trees with  $n$  leaves and  $k$  internal nodes,  $t_n(u) = \sum_{k=0}^{n-1} t_{n,k} u^k$ , and  $T(z, u) = \sum_{n \geq 1} t_n(u) z^n$ , then

$$T(z, u) = z + \frac{uz^2}{1-z} \partial_z T(z, u), \quad (6)$$

where  $\partial_z$  denotes the partial differentiation according to  $z$ . Once again, we write  $(1-z)T(z, u) = z(1-z) + uz^2 \partial_z T(z, u)$ , and extract the coefficient of  $z^n$  on both sides, we thus get that  $t_1(u) = 1$ ,  $t_2(u) = u$  and, for all  $n > 2$ ,

$$t_n(u) = (1 + (n-1)u) t_{n-1}(u). \quad (7)$$

Extracting the coefficient of  $u^k$  on both sides of this last equation gives:  $t_{1,0} = 1$ ,  $t_{n,1} = 1$  and for all  $n > 1$ ,

$$t_{n,k} = t_{n-1,k} + (n-1) t_{n-1,k-1} \quad \text{for all } 0 < k < n,$$

and  $t_{n,k} = 0$  otherwise. The first values of  $t_{n,k}$  are listed in Table 3. Note that, for all  $n \geq 1$ ,  $t_{n,n-1}$  is the number of increasing binary trees (see [FS09, page 143] for details).

1
0, 1
0, 1, 2
0, 1, 5, 6
0, 1, 9, 26, 24
0, 1, 14, 71, 154, 120
0, 1, 20, 155, 580, 1044, 720

TABLE 3. Values of  $t_{n,k}$  (the number of increasing Schröder trees with  $n$  leaves and  $k$  internal nodes) for  $n \in \{1, 2, \dots, 7\}$ , and  $k \in \{0, 1, \dots, n-1\}$ .

From Equation (7), we easily deduce a closed form for  $t_n(u)$ : for all  $n \geq 2$ , we have

$$t_n(u) = u \prod_{\ell=2}^{n-1} (1 + \ell u). \quad (8)$$

This is a shifted version of the sequence [OEIS A145324](#), which is related to Stirling cycle numbers. Our proof of Theorem 2.4.1 relies on the following lemma, which is a straightforward consequence of Equation (8). From now on  $t_n(u)$  will always refer to the generating function of  $t_{n,k}$  with respect to  $k$ , that is  $t_n(u) = \sum_{k=0}^{n-1} t_{n,k} u^k$ .

**Lemma 2.4.2.** *Let  $SC_n(u) = \prod_{i=0}^{n-1} (u + i)$  be the generating functions of the respective rows of the Stirling Cycle numbers (see [FS09, page 735]), which enumerate all permutations of a set of size  $n$  that decompose into  $k$  cycles (i.e. Stirling numbers of the first kind). If we set  $\hat{t}_n(u) = \sum_{k=0}^{n-1} t_{n,k} u^{n-k}$ , which is the row-reversed generating function, then*

$$\hat{t}_n(u) = \frac{SC_n(u)}{1+u} = u \prod_{\ell=2}^{n-1} (u + \ell).$$

*Proof of Theorem 2.4.1.* One could apply Hwang's quasi-powers theorem [Hwa98], but since we have an explicit formula for  $t_n(u)$ , we decided instead to apply Lévy's continuity theorem directly. By Lemma 2.4.2, we have that, if  $\bar{X}_n = n - X_n$ , for all  $\xi \in \mathbb{R}$ ,

$$\begin{aligned} \mathbb{E} \left[ e^{i\xi \cdot \frac{\bar{X}_n - \ln n}{\sqrt{\ln n}}} \right] &= \frac{1}{t_n} e^{-i\xi \sqrt{\ln n}} \hat{t}_n \left( e^{\frac{i\xi}{\sqrt{\ln n}}} \right) = \frac{2}{n!} e^{-i\xi \sqrt{\ln n} + \frac{i\xi}{\sqrt{\ln n}}} \cdot \frac{\Gamma \left( n + e^{\frac{i\xi}{\sqrt{\ln n}}} \right)}{\Gamma \left( 2 + e^{\frac{i\xi}{\sqrt{\ln n}}} \right)} \\ &= \frac{2 + o(1)}{\Gamma(3 + o(1))} \frac{\left( n - 1 + e^{\frac{i\xi}{\sqrt{\ln n}}} \right)^{n + e^{\frac{i\xi}{\sqrt{\ln n}} - \frac{1}{2}}} e^n e^{-i\xi \sqrt{\ln n}}}{e^{n-1 + e^{\frac{i\xi}{\sqrt{\ln n}}} n^{n+\frac{1}{2}}}}, \end{aligned}$$

where we have used Stirling's formula. Note that

$$\lim_{n \rightarrow \infty} e^{1 - e^{\frac{i\xi}{\sqrt{\ln n}}}} = 1,$$

and  $\Gamma(3) = 2$ , which implies that

$$\begin{aligned} \mathbb{E}\left[e^{i\xi \frac{\bar{X}_n - \ln n}{\sqrt{\ln n}}}\right] &= (1 + o(1)) \frac{\left(n - 1 + e^{\frac{i\xi}{\sqrt{\ln n}}}\right)^{n + e^{\frac{i\xi}{\sqrt{\ln n}} - \frac{1}{2}}} e^{-i\xi\sqrt{\ln n}}}{n^{n+1/2}} \\ &= (1 + o(1)) \frac{n e^{\frac{i\xi}{\sqrt{\ln n}}} e^{-i\xi\sqrt{\ln n}}}{n}. \end{aligned}$$

Since

$$\begin{aligned} n e^{\frac{i\xi}{\sqrt{\ln n}}} &= \exp\left((\ln n) \left(1 + \frac{i\xi}{\sqrt{\ln n}} - \frac{\xi^2}{2\ln n} + O\left((\ln n)^{-3/2}\right)\right)\right) \\ &\underset{n \rightarrow \infty}{\sim} n e^{i\xi\sqrt{\ln n} - \xi^2/2}, \end{aligned}$$

we get

$$\mathbb{E}\left[e^{i\xi \frac{\bar{X}_n - \ln n}{\sqrt{\ln n}}}\right] = (1 + o(1)) e^{-\xi^2/2},$$

which, by Lévy's continuity theorem concludes the proof; recall that  $\bar{X}_n = n - X_n$ .  $\square$

**2.4.2. Quantitative analysis of the root node.** In this section, we show that the distribution of the degree of the root of a tree taken uniformly at random among all increasing Schröder trees of size  $n$  is in fact independent of  $n$ :

**Theorem 2.4.3.** *The degree  $A_n^T$  of the root of a tree taken uniformly at random among all increasing Schröder trees of size  $n$  satisfies, for all  $n \geq 1$ ,*

$$\mathbb{E}[A_n^T] = 2e - 3, \quad \text{and} \quad \mathbb{V}[A_n^T] = 14e - 4e^2 - 8.$$

*In fact, for all  $2 \leq k \leq n$ , we have*

$$\mathbb{P}(A_n^T = k) = \frac{2k}{(k+1)!}. \quad (9)$$

The proof of Theorem 2.4.3 relies on the following lemma.

**Lemma 2.4.4.** *If  $t_{n,k}$  is the number of increasing Schröder trees whose root has arity  $k$ , then  $t_{1,0} = 1$ , for all  $n \geq 0$ ,  $t_{n,1} = 0$  and for all  $n \geq 2$ ,  $2 \leq k \leq n - 1$ ,*

$$t_{n,k} = \frac{k n!}{(k+1)!}, \quad \text{and} \quad t_{n,n} = 1.$$

Indeed, this lemma together with the fact that  $t_n = n!/2$ , implies Equation (9). The mean and variance of this probability distribution are then straightforward to calculate, which concludes the proof of Theorem 2.4.3: it only remains to prove Lemma 2.4.4.

We refer the reader to Table 4 where the first values of  $t_{n,k}$  are listed. The sequences  $(t_n(u))_{n \geq 1}$  and  $(t_{n,k})_{2 \leq k < n}$  are related to the sequences [OEIS A094112](#) and [OEIS A092582](#), which enumerate some families of permutations (the former enumerates a family of permutations avoiding some pattern, the second permutations with initial cycle of a given size). Since the number of increasing Schröder trees of size  $n \geq 2$  is equal to  $n!/2$ , it is natural to expect some links between this family of trees and permutations: in Section 2.5 we exhibit a bijection between the two families.

1,	0						
0,	0,	1					
0,	0,	2,	1				
0,	0,	8,	3,	1			
0,	0,	40,	15,	4,	1		
0,	0,	240,	90,	24,	5,	1	
0,	0,	1680,	630,	168,	35,	6,	1

TABLE 4. Values of  $t_{n,k}$ , the number of size- $n$  increasing Schröder trees of root-arity  $k$ , and  $0 \leq k \leq n \in \{1, \dots, 7\}$ .

*Proof of Lemma 2.4.4.* In this proof, the variable  $\mathcal{U}$  marks the arity of the root (we re-use the same notation as in the previous section, but with a different meaning; this is done to avoid having too many different notations). Using the evolution process, we get that

$$\mathcal{T} = \mathcal{Z} \cup (\mathcal{U} \times \mathcal{Z} \times \text{SEQ}_{\geq 1}(\mathcal{U} \times \mathcal{Z})) \cup (\Theta_{\mathcal{Z}}(\mathcal{T} \setminus \mathcal{Z}) \times \text{SEQ}_{\geq 1} \mathcal{Z}).$$

Indeed, the root is either a leaf ( $\mathcal{Z}$ ), or it is an internal node to which is attached a sequence of at least 2 leaves ( $\mathcal{U} \times \mathcal{Z} \times \Theta_{\mathcal{Z}}(\mathcal{Z}) \times \text{SEQ}_{\geq 1}(\mathcal{U} \times \mathcal{Z})$ ), or the tree is larger, i.e. the last step in the evolution process was replacing another leaf by an internal node to which is attached a sequence of non-marked leaves ( $\Theta_{\mathcal{Z}}(\mathcal{T} \setminus \mathcal{Z}) \times \text{SEQ}_{\geq 1} \mathcal{Z}$ ). Using the symbolic method, we thus get that

$$T(z, u) = z + \frac{u^2 z^2}{1 - uz} + \frac{z^2}{1 - z} \partial_z (T(z, u) - z).$$

In the same way as before, through a direct extraction  $[z^n](1 - zu)(1 - z)T(z, u)$ , we prove that  $t_1(u) = 1$ ,  $t_2(u) = u^2$ , and for all  $n > 2$ ,

$$t_n(u) = (u - 1) u^{n-1} + n t_{n-1}(u).$$

This implies  $t_{1,0} = 1$ ,  $t_{n,n} = 1$  for all  $n \geq 1$ ,  $t_{n,k} = n t_{n-1,k}$  for all  $1 \leq k \leq n - 1$ , and  $t_{n,k} = 0$  for all  $k > n$ , which concludes the proof.  $\square$

**2.4.3. Quantitative analysis of the number of nodes of a given arity.** In this section, we prove asymptotic results for the number of nodes of a given arity in a typical increasing Schröder tree, starting with binary nodes:

**Theorem 2.4.5.** *Let  $B_n$  be the number of binary nodes (nodes of arity 2) in a tree taken uniformly at random among all increasing Schröder trees of size  $n$ . Asymptotically when  $n$  tends to infinity, we have*

$$\mathbb{E}[B_n] = n - 2 \ln n + 2\gamma - \frac{7}{3} + \mathcal{O}\left(\frac{1}{n}\right), \quad \text{and} \quad \mathbb{V}[B_n] = 4 \ln n + 4\gamma - \frac{2}{3}\pi^2 - \frac{17}{6} + \mathcal{O}\left(\frac{1}{n}\right),$$

where  $\gamma$  is the Euler-Mascheroni constant. Moreover, in distribution when  $n$  tends to infinity,

$$\frac{B_n - (n - 2 \ln n)}{2\sqrt{\ln n}} \rightarrow \mathcal{N}(0, 1).$$

In other words, on average, almost all internal nodes are binary, and only a proportion of order  $2 \ln n/n$  of internal nodes are at least ternary. In the following theorem, we show that, on average, half of all non-binary nodes are ternary.

**Theorem 2.4.6.** *Let  $C_n^{(\ell)}$  be the number of nodes of arity  $\ell \geq 3$  in a tree taken uniformly at random among all increasing Schröder trees of size  $n$ . Asymptotically when  $n$  tends to infinity, we have*

$$\mathbb{E}C_n^{(3)} = \ln n + \mathcal{O}(1), \quad \text{and } \mathbb{E}C_n^{(\ell)} \sim c_\ell,$$

for some positive constants  $(c_\ell)_{\ell \geq 4}$ ; and, for  $\ell = 4$ , we have  $c_4 = 23/90$ .

*Proof of Theorem 2.4.5.* Here the specification is easy to exhibit, and its translation via the symbolic method is direct (in this proof,  $\mathcal{U}$  marks the binary nodes):

$$\mathcal{T} = \mathcal{Z} \cup (\Theta_{\mathcal{Z}}\mathcal{T} \times (\mathcal{U} \times \mathcal{Z} \cup \text{SEQ}_{\geq 2} \mathcal{Z}));$$

$$T(z, u) = z + \left( u z^2 + \frac{z^3}{1-z} \right) \partial_z T(z, u).$$

The method we use to analyse this differential equation is similar to [CHY00]. For all  $n \geq 3$ ,

$$t_n(u) = (1 + u(n-1))t_{n-1}(u) + (1-u)(n-2)t_{n-2}(u), \quad (10)$$

with  $t_1(u) = 1$  and  $t_2(u) = u$ . Once again (see also Lemma 2.4.2) we take the normalised row-reversed generating function  $\hat{t}_n(u) = \frac{2}{n!} \sum_{k=0}^{n-1} t_{n,k} u^{n-k} = \frac{2}{n!} u^n t_n(1/u)$ . From Equation (10), we get that, for all  $n \geq 3$ ,

$$\hat{t}_n(u) = \frac{n+u-1}{n} \hat{t}_{n-1}(u) + \frac{u(u-1)(n-2)}{n(n-1)} \hat{t}_{n-2}(u), \quad (11)$$

with  $\hat{t}_1(u) = 2u$  and  $\hat{t}_2(u) = u$ .

We now define  $F(z, u) = \sum_{n \geq 1} n \hat{t}_n(u) z^n$ ; this generating function satisfies the following differential equation:

$$z(1-z) \partial_z F(z, u) = (1 + uz - u(1-u)z^2) F(z, u) + 2u^2 z^2,$$

with initial condition  $\partial_z F(z, u)|_{z=0} = 2u$ . This last equation gives

$$F(z, u) = \frac{2uz}{(1-z)^{u^2+1}} e^{u(1-u)z} \cdot \left( 1 - \int_0^z u(1-t)^{u^2} e^{-u(1-u)t} dt \right).$$

We remark that  $F(z, u)$  is singular at  $z = 1$  with a pole of order  $u^2 + 1$ , thus

$$F(z, u) \underset{z=1}{\sim} \frac{g(u)}{(1-z)^{u^2+1}},$$

with

$$g(u) = 2ue^{u(1-u)} \left( 1 - \int_0^1 u(1-t)^{u^2} e^{-u(1-u)t} dt \right).$$

Therefore, asymptotically when  $n \rightarrow +\infty$ ,

$$n \hat{t}_n(u) = \frac{g(u)}{\Gamma(u^2+1)} n^{u^2} \left( 1 + \mathcal{O}\left(\frac{1}{n}\right) \right),$$

uniformly for all  $u$  such that  $|u-1| \leq \delta$  for some  $\delta > 0$ . This thus falls into the scope of the quasi-powers framework and Theorem IX.8 [FS09, p. 645] is applicable with  $B(u) = \exp(2u)$  and  $\beta_n = \ln n$ , which concludes the proof (the mean and variance expansions can be calculated automatically using, e.g., a computer software such as Maple).  $\square$

*Proof of Theorem 2.4.6.* We first look at ternary nodes: the specification (with  $\mathcal{U}$  marking ternary nodes) is given by

$$\mathcal{T} = \mathcal{Z} \cup (\Theta_{\mathcal{Z}}\mathcal{T} \times (\mathcal{Z} \cup \mathcal{U} \times \mathcal{Z}^2 \cup \text{SEQ}_{\geq 3} \mathcal{Z}))$$

which implies

$$T(z, u) = z + \left( z^2 + u z^3 + \frac{z^4}{1-z} \right) \partial_z T(z, u),$$

and thus, for all  $n \geq 4$ :

$$t_n(u) = n t_{n-1}(u) + (u-1)(n-2)t_{n-2}(u) + (n-3)(1-u)t_{n-3}(u),$$

with  $t_0(u) = 0$ ,  $t_1(u) = 1$  and  $t_2(u) = 1$ . Differentiating this equation, we get that, for all  $n \geq 5$ ,

$$t'_n(u)|_{u=1} = n t'_{n-1}(u)|_{u=1} + \frac{(n-2)(n-2)!}{2} - \frac{(n-3)(n-3)!}{2}.$$

This thus implies that, for all  $n \geq 5$ ,

$$\begin{aligned} \mathbb{E}[C_n^{(3)}] &= \mathbb{E}[C_{n-1}^{(3)}] + \frac{(n-2)}{n(n-1)} - \frac{(n-3)}{n(n-1)(n-2)} \\ &= \frac{5}{12} + \sum_{\ell=5}^n \left( \frac{(k-2)}{k(k-1)} - \frac{(k-3)}{k(k-1)(k-2)} \right). \end{aligned}$$

since  $\mathbb{E}[C_4^{(3)}] = 5/12$ . Using the fact that  $\sum_{k=1}^n \frac{1}{k} = \ln n + \mathcal{O}(1)$  and  $\sum_{k=1}^n \frac{1}{k^2} = \mathcal{O}(1)$  when  $n$  tends to infinity, we get

$$\mathbb{E}[C_n^{(3)}] = \ln n + \mathcal{O}(1),$$

as claimed.

We reason similarly for  $\ell = 4$  ( $\mathcal{U}$  now marks nodes of arity 4):

$$\mathcal{T} = \mathcal{Z} \cup (\Theta_{\mathcal{Z}}\mathcal{T} \times (\mathcal{Z} \cup \mathcal{Z}^2 \cup \mathcal{U} \times \mathcal{Z}^3 \cup \text{SEQ}_{\geq 4} \mathcal{Z}));$$

$$T(z, u) = z + \left( z^2 + z^3 + u z^4 + \frac{z^5}{1-z} \right) \partial_z T(z, u).$$

Thus, for all  $n \geq 4$ , we have

$$t_n(u) = n t_{n-1}(u) + (u-1)(n-3)t_{n-3}(u) + (n-4)(1-u)t_{n-4}(u),$$

with  $t_0 = 0$ ,  $t_1(u) = 1$ ,  $t_2(u) = 1$  and  $t_3(u) = 3$ , which, after differentiating at  $u = 1$  and dividing by  $t_n$  gives, for  $n \geq 6$ ,

$$\mathbb{E}[C_n^{(4)}] = \mathbb{E}[C_{n-1}^{(4)}] + \frac{(n-3)}{n(n-1)(n-2)} - \frac{(n-4)}{n(n-1)(n-2)(n-3)},$$

with  $\mathbb{E}[C_5^{(4)}] = 6/60$ . The recurrence can be solved explicitly and we obtain,

$$\mathbb{E}[C_n^{(4)}] = \frac{23}{90} - \frac{13}{6n} - \frac{1}{6(n-2)} + \frac{4}{3(n-1)},$$

which proves the statement for  $\ell = 4$ .

We now treat the general  $\ell \geq 5$  case ( $\mathcal{U}$  now marks nodes of arity  $\ell$ ):

$$\mathcal{T} = \mathcal{Z} \cup (\Theta_{\mathcal{Z}}\mathcal{T} \times ((\cup_{i=1}^{\ell-2} \mathcal{Z}^i) \cup \mathcal{U} \times \mathcal{Z}^{\ell-1} \cup \text{SEQ}_{\geq \ell} \mathcal{Z}));$$

$$T(z, u) = z + \left( \left( \sum_{i=2}^{\ell-1} z^i \right) + u z^{\ell} + \frac{z^{\ell+1}}{1-z} \right) \partial_z T(z, u).$$

This implies that, for all  $n \geq \ell$ :

$$t_n(u) = nt_{n-1}(u) + (u - 1)(n - \ell + 1)t_{n-\ell+1}(u) + (n - \ell)(1 - u)t_{n-\ell}(u),$$

with  $t_n(u) = t_n$  for all  $n < \ell$ . Therefore, we get, for  $n \geq \ell + 2$ ,

$$\begin{aligned} \mathbb{E}[C_n^{(\ell)}] &= \mathbb{E}[C_{n-1}^{(\ell)}] + \frac{n - \ell + 1}{n(n - 1) \cdots (n - \ell + 2)} - \frac{n - \ell}{n(n - 1) \cdots (n - \ell + 1)} \\ &= \mathbb{E}[C_{\ell+1}^{(\ell)}] + \sum_{k=\ell+1}^n \frac{(k - \ell + 1)}{k(k - 1) \cdots (k - \ell + 2)} - \frac{(k - \ell)}{k(k - 1) \cdots (k - \ell + 1)}. \end{aligned}$$

Since

$$\frac{k - \ell + 1}{k(k - 1) \cdots (k - \ell + 2)} - \frac{k - \ell}{k(k - 1) \cdots (k - \ell + 1)} \underset{k \rightarrow \infty}{\sim} \frac{1}{k^{\ell-2}},$$

which implies that, for all  $\ell \geq 4$ ,

$$\begin{aligned} &\lim_{n \rightarrow \infty} \mathbb{E}[C_n^{(\ell)}] \\ &= \mathbb{E}[C_{\ell+1}^{(\ell)}] + \sum_{k=\ell+1}^{\infty} \frac{k - \ell + 1}{k(k - 1) \cdots (k - \ell + 2)} - \frac{k - \ell}{k(k - 1) \cdots (k - \ell + 1)} < +\infty. \end{aligned}$$

All these sequences converge to constants that get smaller and smaller when  $\ell$  increases.  $\square$

Table 3 gives a summary of the typical number of nodes for the smallest arities: the first three values appear explicitly in the latter proofs and an obvious adaptation of Theorem 2.4.6 gives the constants  $c_\ell$  for larger  $\ell$ .

	2-ary	3-ary	4-ary	5-ary	6-ary	7-ary	8-ary	9-ary	10-ary
$\mathbb{E}C_n^{(\ell)}$	$n - 2 \ln n$	$\ln n$	$\frac{23}{90}$	$\frac{1}{32}$	$\frac{107}{25200}$	$\frac{47}{86400}$	$\frac{101}{1587600}$	$\frac{229}{33868800}$	$\frac{659}{1005903360}$

FIGURE 3. The asymptotic number of  $\ell$ -ary nodes

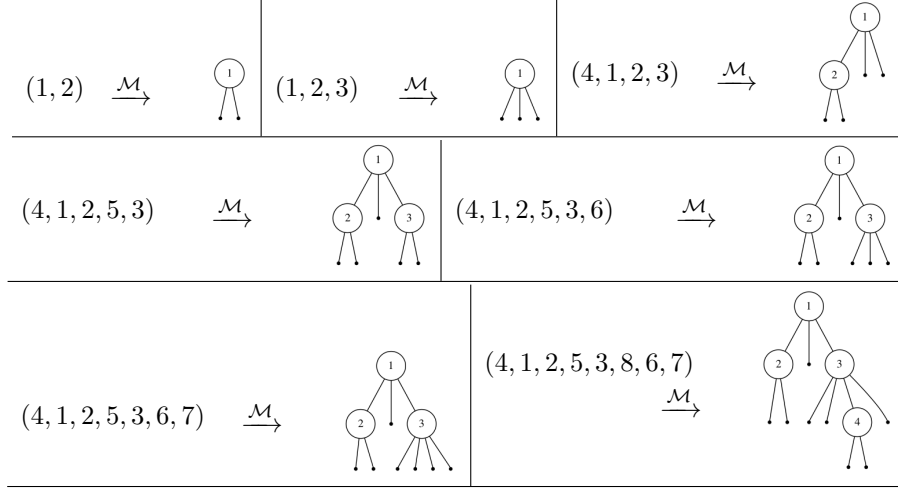
**2.5. Bijection with permutations.** Since the number of increasing Schröder trees of size  $n$  is equal to  $t_n = n!/2$ , it hints at the existence of a relationship between our model of increasing trees and a subclass of permutations. In this section, we aim at exhibiting this relationship.

We denote by  $\sigma = (\sigma_1, \dots, \sigma_n)$  the size- $n$  permutation that sends  $i$  to  $\sigma_i \in \{1, \dots, n\}$  for all  $i \in \{1, \dots, n\}$ . For all  $k \in \{1, \dots, n\}$ , we denote by  $\sigma^{-1}(k)$  the pre-image of  $k$  by  $\sigma$ , and sometimes call  $\sigma^{-1}(k)$  the “position” of  $k$  in the permutation  $\sigma$ .

We now define recursively a map  $\mathcal{M}$  between  $\mathcal{HP}$ , the class of permutations such that 1 appears before 2, and the class  $\mathcal{T}$  of increasing Schröder trees.

The only element of  $\mathcal{HP}$  of size 2 is the permutation (1, 2); we set its image to be the tree whose root is labelled by 1 and has two (unlabelled) leaf-children. Now assume that we have defined  $\mathcal{M}(\sigma)$  for all permutations  $\sigma \in \mathcal{HP}$  of size at most  $n - 1$  for some  $n \geq 2$  and let  $\sigma$  be a size- $n$  permutation in  $\mathcal{HP}$ . We distinguish two cases according to the pre-image of  $n$  by  $\sigma$ ; we denote by  $\hat{\sigma}_i = \sigma_i$  if  $\sigma_i < \sigma_n$  and  $\hat{\sigma}_i = \sigma_i - 1$  otherwise. For example, if  $\sigma = (4, 1, 5, 2, 3)$ , then  $\hat{\sigma} = (3, 1, 4, 2)$ ;  $\hat{\sigma}$  can be seen as the permutation induced by  $\sigma$  on  $\{1, \dots, n - 1\}$ .

- If  $\sigma^{-1}(n) = n$  then, we set  $\mathcal{M}(\sigma)$  to be the tree  $\mathcal{M}(\hat{\sigma})$  in which we add a new rightmost leaf to the internal node with the largest label.
- If  $\sigma^{-1}(n) = k < n$ , then, we build  $\mathcal{M}(\sigma)$  as follows: create a new binary node  $\nu$  labelled with the smallest integer that does not appear as a label in  $\mathcal{M}(\hat{\sigma})$  and attach two new leaves to this internal node. Insert this tree in  $\mathcal{M}(\hat{\sigma})$  by placing  $\nu$  in the  $k$ -th leaf (we assume, for example, that the leaves are ordered in the depth-first order) of  $\mathcal{M}(\hat{\sigma})$ .

FIGURE 4. A size-8 example of the mapping  $\mathcal{M}$ 

In Figure 4 we present the mapping on an example, where we have ordered the steps reversely to understand the process in a constructive way.

**Theorem 2.5.1.** *The map  $\mathcal{M}$  is a one-to-one map between  $\mathcal{HP}$  and  $\mathcal{T}$ .*

*Proof.* First note that the image by  $\mathcal{M}$  of a permutation of size  $n$  is a Schröder tree of size  $n$ : indeed, at each iteration we remove exactly one element from the permutation and add exactly one leaf to the tree by either adding a leaf to the node with largest label or by removing one leaf and adding two new ones. Since the number of permutations of size  $n$  in  $\mathcal{HP}$  is equal to the number of Schröder trees of size  $n$ , it is enough to prove that  $\mathcal{M}$  is injective to conclude the proof. The mapping is injective since by induction at each iteration we remove the greatest element of the permutation and the following actions are performed on the resulting tree in a non-ambiguous manner. This concludes the proof.  $\square$

**2.6. Uniform random sampling.** In this section, we present an algorithm that samples a Schröder tree uniformly at random among all Schröder trees of a given size. Our aim is to use this algorithm to generate trees of large size (typically several thousands of leaves): we thus provide a detailed analysis of the complexity of our sampler.

Note that the uniform sampling of structures with increasing labelling constraints is not so classical in the context of analytic combinatorics. Martínez and Molinero [MM03, Mol05] focus on the recursive method: using and generalising recursive and unranking generation methods, they give a method that, given a combinatorial specification, automatically outputs a uniform generation algorithm and its complexity analysis. Using a different approach based on Boltzmann generation, Bodini, Roussel and Soria [BRS12] give an algorithmic framework to develop Boltzmann samplers in the context of specifications that lead to differential equations of the first order. The paper [BDF<sup>+</sup>16] shows that this framework can be extended to the context of differential equations of higher order; in particular, they apply this method to the generation of diamonds satisfying differential equations of order 2.

The bijection presented in Section 2.5 immediately gives an algorithm that samples a tree uniformly among all Schröder trees of size  $n$ : first sample a permutation uniformly at random among all permutations of size  $n$  in  $\mathcal{HP}$ , and then build its image by  $\mathcal{M}$ . While there exists fast algorithms to sample permutations (see for example [BBHT17]), it is not clear how to make the application of  $\mathcal{M}$  efficient.

Instead, we use the bijection  $\mathcal{M}$  as a basis for a direct *probabilistic construction*. Indeed, one can sample a permutation uniformly at random in  $\mathcal{HP}$  by doing the following recursive procedure: if  $n = 2$ , then return  $\sigma^{(2)} = (1, 2)$ . If  $n \geq 3$ , assume we have sampled  $\sigma^{(n-1)}$  uniformly among all permutations of size  $n - 1$ . Draw an integer  $k_n$  uniformly at random in  $\{1, \dots, n\}$ , and set  $\sigma_n^{(n)} = k_n$ , and

$$\sigma_i^{(n)} = \begin{cases} \sigma_i^{(n-1)} & \text{if } \sigma_i^{(n-1)} < k_n \\ \sigma_i^{(n-1)} + 1 & \text{otherwise.} \end{cases}$$

One can indeed check that  $\sigma^{(n)}$  is uniformly distributed among all permutations of size  $n$  in  $\mathcal{HP}$ . Executing this random sampling of  $\sigma^{(n)}$  simultaneously with  $\mathcal{M}$  (note that, for all  $n \geq 3$ ,  $\sigma^{(n-1)} = \hat{\sigma}^{(n)}$ , where the notation  $\hat{\sigma}$  is defined in the definition of  $\mathcal{M}$ ) is the idea of our sampler:

---

**Algorithm 1** Increasing Schröder Tree Builder

---

```

1: function TREEBUILDER( $n$ )
2:   if  $n = 1$  then
3:     return the single leaf
4:    $T =$  the root labelled by 1 and attached to two leaves
5:    $\ell = 2$ 
6:   for  $i$  from 3 to  $n$  do
7:      $k = \text{rand\_int}(1, i)$ 
8:     if  $k = i$  then
9:       Add a new leaf to the last added internal node in  $T$ 
10:    else
11:      Create a new binary node at position  $k$  in  $T$ 
12:        with label  $\ell$  and attached to two leaves
13:       $\ell = \ell + 1$ 
14:   return  $T$ 

```

The function  $\text{rand\_int}(a, b)$  returns uniformly at random an integer in  $\{a, a + 1, \dots, b\}$ .

---

Using adequate data structures, as for example by keeping an array of pointers to all leaves and another one to the last inserted internal node, each insertion in the tree under construction is done in constant time. We thus get

**Theorem 2.6.1.** *The function  $\text{TREEBUILDER}(n)$  in Algorithm 1 is a uniform sampling algorithm for size- $n$  trees. Asymptotically, it operates in  $\mathcal{O}(n)$  operations on trees and necessitates  $\mathcal{O}(n \ln n)$  random bits.*

**2.7. Analysis of the height of a typical increasing Schröder tree.** The probabilistic construction used in our uniform sampler allows us to prove the following result.

**Theorem 2.7.1.** *For all  $n \geq 2$ , let  $H_n$  be the height of a tree taken uniformly at random among all increasing Schröder trees of size  $n$ . Asymptotically when  $n$  tends to infinity,*

$$\mathbb{P}\left(\frac{H_n}{\ln n} \in [1 - \varepsilon, \gamma + \varepsilon]\right) \rightarrow 1,$$

where  $\gamma = \inf\{c \geq 2: c - 1 + c \ln(2/c) < 0\} \approx 4.311$ . This implies in particular that  $\mathbb{E}[H_n] = \Theta(\ln n)$  when  $n$  tends to infinity.

Note that, equivalently,  $\gamma$  can be defined as the smallest solution of  $c - 1 + c \ln(2/c) = 0$  on  $[2, \infty)$ .

We remark all other results in this section on increasing Schröder trees are more precise than a  $\Theta$  order. However, since the height involves a maximum (maximum height of a leaf in the tree), it is not a surprise that the height is more intricate to study than other parameters. In the literature, precise results on the height are known for two classes of random trees: the trees for which a scaling limit result can be proved (e.g. Catalan trees' scaling limit is Aldous' continuous random tree), or the split trees of Devroye (see, e.g., [Dev90]). Increasing Schröder trees fall in neither of these two classes; however, we are able to adapt Devroye's techniques to get the upper bound in Theorem 2.7.1.

We split the proof of Theorem 2.7.1 in two parts: we prove the upper bound in Section 2.7.1, and the lower bound in Section 2.7.2

**2.7.1. Upper bound in Theorem 2.7.1.**

**Definition 2.7.2.** *Given a sequence of integers  $\mathbf{d} = (d_i)_{i \geq 1}$ , we define the random  $\mathbf{d}$ -ary tree  $(\tau_n^{(\mathbf{d})})_{n \geq 0}$  recursively as follows:  $\tau_0^{(\mathbf{d})}$  is reduced to its root, given  $\tau_{\ell-1}^{(\mathbf{d})}$ , we build  $\tau_\ell^{(\mathbf{d})}$  as the tree obtained by picking a leaf uniformly at random in  $\tau_{\ell-1}^{(\mathbf{d})}$  and replacing it by a node to which  $d_\ell$  leaves are attached.*

**Lemma 2.7.3.** *Let  $\mathbf{D} = (D_\ell)_{\ell \geq 1}$  be the sequence of integer-valued random variables defined for all  $k \geq 2$  by:*

- (i)  $\mathbb{P}(D_1 = k) = 2k/(k+1)!$ , and
- (ii) if, for all  $\ell \geq 1$ , we denote by  $\bar{D}_\ell = \sum_{i=1}^{\ell} D_i$ , then,

$$\mathbb{P}(D_{\ell+1} = k | D_1, \dots, D_\ell) = \frac{(\bar{D}_\ell + 1)!(k - 1 + \bar{D}_\ell)}{(k + \bar{D}_\ell)!}.$$

Then, for all  $\ell \geq 1$ , the tree  $\tau_\ell^{(\mathbf{D})}$  is equal in distribution to an increasing Schröder tree taken uniformly at random among all trees of its size.

*Proof.* To prove this claim, it is enough to show that if, for all  $i \geq 1$ ,  $D_i$  is the degree of node  $i$  in an increasing Schröder tree taken uniformly at random among all trees of its size, then the distribution of  $(D_i)_{i \geq 1}$  satisfies (i) and (ii). We show how this follows from Theorem 2.6.1: Indeed, note that the degree of the last inserted internal node increases as long as the random integer  $k = k_i$  (see line 7

of Algorithm 1) drawn in the  $i$ -th loop is equal to  $i$ . Note that this happens with probability  $1/i$ . For example, the degree of the root starts at 2, we draw the first integer  $k_3 \in \{1, 2, 3\}$  and if  $k_3 \neq 3$ , then we can conclude that  $D_1 = 2$ , otherwise, we know that  $D_1 \geq 3$  and we need to look at  $k_4$ . Therefore,  $\mathbb{P}(D_1 = 2) = 2/3$ , as claimed, and  $\mathbb{P}(D_1 \geq 3) = 1/3$ . Iterating this argument, we get that

$$\mathbb{P}(D_1 \geq k) = \prod_{i=3}^k \mathbb{P}(k_i = i) = \prod_{i=3}^k \frac{1}{i} = \frac{2}{k!},$$

and thus

$$\mathbb{P}(D_1 = k) = \mathbb{P}(D_1 \geq k) - \mathbb{P}(D_1 \geq k+1) = \frac{2}{k!} - \frac{2}{(k+1)!} = \frac{2k}{(k+1)!},$$

as claimed.

By definition of our sampling algorithm, we know that the  $(\ell+1)$ -st internal node is inserted into the tree during the loop number  $i = D_1 + \dots + D_\ell + 1 = \bar{D}_\ell + 1$ . Therefore, we get

$$\mathbb{P}(D_{\ell+1} = 2 | D_1, \dots, D_\ell) = \mathbb{P}(k_{i+1} \neq i+1) = 1 - \frac{1}{\bar{D}_\ell + 2},$$

as claimed, and

$$\mathbb{P}(D_{\ell+1} \geq 3 | D_1, \dots, D_\ell) = \frac{1}{\bar{D}_\ell + 2}.$$

Iterating this argument, we get that, for all  $k \geq 3$ ,

$$\mathbb{P}(D_{\ell+1} \geq k | D_1, \dots, D_\ell) = \prod_{j=\bar{D}_\ell+2}^{\bar{D}_\ell+k-1} \mathbb{P}(k_j = j) = \prod_{j=\bar{D}_\ell+2}^{\bar{D}_\ell+k-1} \frac{1}{j} = \frac{(\bar{D}_\ell + 1)!}{(\bar{D}_\ell + k - 1)!}.$$

This concludes the proof because

$$\begin{aligned} & \mathbb{P}(D_{\ell+1} = k | D_1, \dots, D_\ell) \\ &= \mathbb{P}(D_{\ell+1} \geq k | D_1, \dots, D_\ell) - \mathbb{P}(D_{\ell+1} \geq k+1 | D_1, \dots, D_\ell) \\ &= \frac{(\bar{D}_\ell + 1)!}{(\bar{D}_\ell + k - 1)!} - \frac{(\bar{D}_\ell + 1)!}{(\bar{D}_\ell + k)!} = \frac{(\bar{D}_\ell + 1)! (\bar{D}_\ell + k - 1)}{(k + \bar{D}_\ell)!}. \end{aligned}$$

□

This lemma allows us to prove the upper-bound in Theorem 2.7.1: This proof is an adaptation of Devroye [Dev90] in which the case of regular trees is treated (in regular trees, nodes have all the same degree; they are also known as random  $k$ -ary trees). We denote by  $N_1(n), \dots, N_{D_1}(n)$  the sizes of the subtrees of the root of  $\tau_n^{(D)}$ ; a straightforward adaptation of [Dev90, Lemma 2] gives that, conditionally on  $D_1$ ,

$$\mathbb{P}((n - m + 2)S_1 \geq x) \leq \mathbb{P}(N_1(n) \geq x) \leq \mathbb{P}(nS_1 \geq x), \quad (12)$$

where  $S_1$  is the minimum of  $D_1 - 1$  i.i.d. random variables uniform on  $[0, 1]$ . We reason conditionally on the sequence  $\mathbf{D}$  of random degrees, and denote by  $\mathbb{P}_{\mathbf{D}}$  the law under this conditioning. We denote by  $S_1, \dots, S_{D_1}$  the spacings induced on  $[0, 1]$  by a sample of  $D_1 - 1$  i.i.d. random variables uniform on  $[0, 1]$ . Using the

fact that the sizes of the subtrees of the root,  $N_1(n), \dots, N_{D_1}(n)$  all have the same distribution, we get

$$\begin{aligned} \mathbb{P}_{\mathcal{D}}(H_n \geq k) &\leq \sum_{i=1}^{D_1} \mathbb{P}_{\mathcal{D}}(H_{N_i(n)} \geq k-1) = D_1 \mathbb{P}_{\mathcal{D}}(H_{N_1(n)} \geq k-1) \\ &\leq D_1 \mathbb{P}_{\mathcal{D}}(H_{nS_1} \geq k-1), \end{aligned}$$

where we have used Equation (12) in the last inequality. We now iterate this identity: we denote by  $I(n) = n \prod_{i=1}^k S(D_i)$ , where, for all  $d \geq 2$ ,  $S(d)$  is the minimum of  $d-1$  i.i.d. random variables uniform on  $[0, 1]$ . We get

$$\mathbb{P}_{\mathcal{D}}(H_n \geq k) \leq \left( \prod_{i=1}^k D_i \right) \mathbb{P}_{\mathcal{D}}(H_{I(n)} \geq 0) = \left( \prod_{i=1}^k D_i \right) \mathbb{P}_{\mathcal{D}} \left( n \prod_{i=1}^k S(D_i) \geq 1 \right),$$

because a tree has height at least 1 as soon as it has at least one internal node. We now use Chebychev's inequality, which implies that, for all  $\alpha \geq 1$ ,

$$\mathbb{P}_{\mathcal{D}}(H_n \geq k) \leq \left( \prod_{i=1}^k D_i \right) n^\alpha \mathbb{E}_{\mathcal{D}} \left[ \prod_{i=1}^k S(D_i)^\alpha \right] = n^\alpha \prod_{i=1}^k \left( \frac{\Gamma(D_i + 1)}{\prod_{i=1}^{D_i-1} (\alpha + i)} \right).$$

See [Dev90, Equation (1)] for the last equality. For all  $\alpha \geq 1$ , and for all  $d \geq 2$ , we have

$$\begin{aligned} \ln \Gamma(d+2) - \sum_{i=1}^d \ln(\alpha + i) &= \ln \Gamma(d+1) - \sum_{i=1}^{d-1} \ln(\alpha + i) + \ln(d+1) - \ln(\alpha + d) \\ &\leq \ln \Gamma(d+1) - \sum_{i=1}^{d-1} \ln(\alpha + i). \end{aligned}$$

Therefore, since  $D_i \geq 2$  almost surely for all  $i \geq 1$ , we get

$$\mathbb{P}_{\mathcal{D}}(H_n \geq k) \leq n^\alpha \prod_{i=1}^k \left( \frac{\Gamma(3)}{\alpha + 1} \right) = n^\alpha \left( \frac{2}{\alpha + 1} \right)^k.$$

This expression is minimised for  $\alpha = k/\ln n - 1$ ; taking  $k = c \ln n$  and  $\alpha = c - 1$ , we get that, for all  $c \geq 2$  (because  $\alpha \geq 1$ ),

$$\mathbb{P}_{\mathcal{D}}(H_n \geq c \ln n) \leq n^{c-1+c \ln(2/c)}.$$

If we take  $c > \gamma$  where  $\gamma = \inf\{c \geq 2: c-1+c \ln(2/c) < 0\}$ , then

$$\mathbb{P}_{\mathcal{D}}(H_n \geq c \ln n) \xrightarrow[n \rightarrow \infty]{} 0,$$

which concludes the proof for the upper bound in Theorem 2.7.1.

**2.7.2. Lower bound in Theorem 2.7.1.** For this proof, we use the fact that the height of a tree is larger or equal to the depth of the leftmost leaf; the following lemma gives information on the behaviour of this quantity:

**Lemma 2.7.4.** *Let  $Y_n$  be the depth of the leftmost leaf in a tree taken uniformly at random among all increasing Schröder trees of size  $n$ . For  $n \geq 1$ ,  $Y_n \stackrel{d}{=} n - X_n$ , where  $X_n$  is the number of internal nodes in a typical increasing Schröder tree of*

size  $n$  (see Theorem 2.4.1), and thus, we have convergence in distribution when  $n$  tends to infinity:

$$\frac{Y_n - \ln n}{\sqrt{\ln n}} \xrightarrow{d} \mathcal{N}(0, 1).$$

Note that the choice of the leftmost leaf is arbitrary, although it has the advantage that the specification is straightforward.

*Proof.* We directly look at the differential equation satisfied by  $T(z, u)$ , where  $u$  marks the internal nodes that belong to the leftmost path (between the root and the leftmost leaf).

$$T(z, u) = z + \partial_z \left( \frac{T(z, u)}{z} \right) \frac{z^3}{1-z} + T(z, u) \frac{uz}{1-z}.$$

Indeed, the tree is either a unique leaf (which is thus also the leftmost leaf) at height zero ( $z$ ), or at the last step of the evolution process, we have selected a leaf that is not the leftmost one and replaced it by a sequence of at least two leaves ( $\partial_z(T(z, u)/z) \frac{z^3}{1-z}$ ), or we have replaced the leftmost leaf by an internal node and a sequence of at least two leaves ( $T(z, u) \frac{uz}{1-z}$ ). We rewrite this equation as

$$(1 - uz)T(z, u) = z(1 - z) + z^2 \partial_z T(z, u),$$

and thus, identifying the coefficient of  $z^n$  on both sides gives that

$$t_n(u) = (u + n - 1)t_{n-1}(u),$$

$t_1(u) = 1$ , and  $t_2(u) = u$ . This implies that, for all  $n \geq 3$ ,

$$t_n(u) = u \prod_{i=2}^{n-1} (u + i) = \frac{1}{1+u} SC_n(u),$$

where  $SC_n(u)$ , defined in Lemma 2.4.2, is the generating function of all size  $n$  permutations with  $k$  cycles. Therefore, using Lemma 2.4.2, we get that  $Y_n \stackrel{d}{=} n - X_n$  in distribution, where  $X_n$  is the number of internal nodes in a typical increasing Schröder tree, which concludes the proof, by Theorem 2.4.1.  $\square$

The lower bound of Theorem 2.7.1 follows from Lemma 2.7.4 since, almost surely for all  $\ell \geq 1$ ,  $H_\ell \geq Y_{\bar{D}_\ell+1}$ , where we recall that  $Y_n$  is the depth of the leftmost leaf in an  $n$ -leaf uniform increasing Schröder tree and  $\bar{D}_\ell = \sum_{i=1}^\ell D_i$ . Indeed, by Lemma 2.7.4, we have that, for all  $\varepsilon > 0$ ,

$$\mathbb{P}(H_n \leq (1 - \varepsilon) \ln n) \leq \mathbb{P}(Y_n \leq (1 - \varepsilon) \ln n) \leq \mathbb{P}\left(\frac{Y_n - \ln n}{\sqrt{\ln n}} \leq -\varepsilon \sqrt{\ln n}\right) \rightarrow 0,$$

when  $n$  tends to infinity, which concludes the proof of the lower bound of Theorem 2.7.1.

### 3. STRICT MONOTONIC SCHRÖDER TREES

**3.1. The model and its context.** In this section we introduce and study a generalisation of the increasing Schröder trees, which we call *strict monotonic Schröder trees*. The main difference between the two models is that in strict monotonic Schröder trees, several internal nodes can be labelled by the same integer as long as they are not on the same ancestral line:

**Definition 3.1.1.** A strict monotonic Schröder tree is a classical Schröder tree structure whose internal nodes are labelled by the integers between 1 and  $\ell$  (for some  $\ell \geq 1$ ), in such a way that each integer in  $\{1, \dots, \ell\}$  appears at least once in the tree and the sequence of labels in the path from the root to any leaf is (strictly) increasing.

We remark that the trees are qualified by “strict” in the sense that the sequence of labels along the paths from the root to any leaf is strictly increasing.

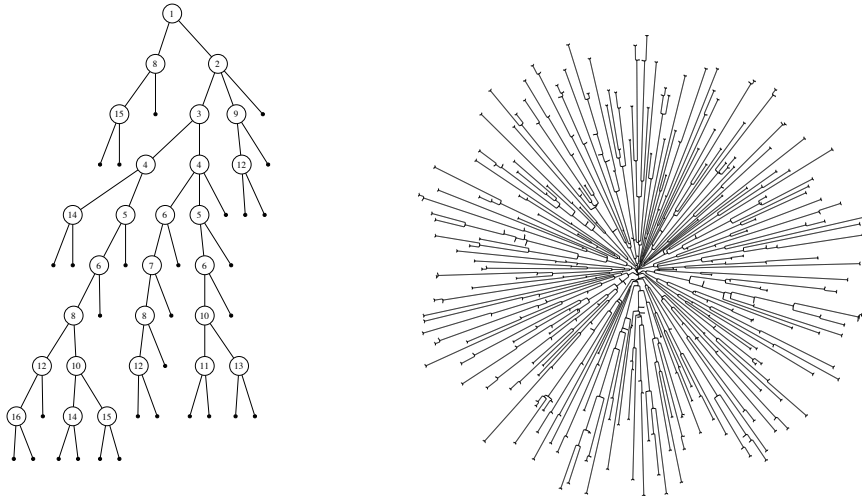


FIGURE 5. Two strict monotonic Schröder trees

In Figure 5 we show two strict monotonic trees: the left-hand-side one is of size 30 with 16 distinct labels, the right-hand-side one is of size 500 (sampled uniformly at random among all trees of size 500), with 495 internal nodes labelled with 372 distinct labels.

Because of the possible repetition of labels, this class of labelled trees cannot be directly specified using the classical analytic combinatorics operators for labelled structures. However, the following recursive construction allows us to specify the class of strict monotonic Schröder trees using operators for unlabelled structures. Every strict monotonic Schröder tree can be built as follows:

- Start with a single (unlabelled) leaf.
- At step  $\ell$  (for  $\ell \geq 1$ ), select a non-empty subset of leaves and replace each of them by an internal node with label  $\ell$  attached to a sequence of at least two leaves.

**3.2. Overview of the main results.** As in Section 2, we first solve the counting problem and its asymptotic equivalent for this family of trees (see Section 3.3). We then notice that the enumeration problem corresponds to the one of *ordered set partitions* which are counted by *ordered Bell numbers*, and, in Section 3.4, we exhibit a bijection between weakly increasing Schröder trees and ordered set partitions. Solving the counting problem and exhibiting this bijection allows us to

study some typical parameters of these trees (see Section 3.5; the results of this section are summarised in Table 5 below). Finally, in Section 3.6, we conclude this part on weakly increasing Schröder trees by designing a uniform random sampler, which also gives an unranking algorithm.

	Mean	Var.	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	$2 \ln 2 + 1$	$2 \ln 2 (1 - \ln 2)$	(shifted) zero-truncated Poisson

TABLE 5. Summary of the main results on parameters of *Strict monotonic Schröder trees*

**3.3. Enumeration and relationship with ordered Bell numbers.** Using the iterative construction described above, we deduce the following specification for the class  $\mathcal{G}$  of all strict monotonic Schröder trees:

$$\mathcal{G} = \mathcal{Z} \cup (\mathcal{G}[\mathcal{Z} \rightarrow (\mathcal{Z} \cup \text{SEQ}_{\geq 2} \mathcal{Z})]) \setminus \mathcal{G}.$$

Note that again the labelling does not appear directly in the specification. The combinatorial meaning of this specification is the following: A tree of  $\mathcal{G}$  is either a single leaf, or it is obtained by taking an already constructed tree in  $\mathcal{G}$ , and replace each leaf by either a leaf (i.e. no change) or an internal node attached to a sequence of at least two leaves. Furthermore we omit the case where no leaf is changing (this is why we subtract the set  $\mathcal{G}$ ). Note that subtracting  $\mathcal{G}$  is important, otherwise some integer values could be absent in the final tree. For example, if there is no change at step 2 but then the evolution continues, then 2 would not appear in the final tree but larger integers would appear as labels.

Using the symbolic method, we can translate this specification into a functional equation (with substitution) for the ordinary generating series:

$$G(z) = z + G\left(z + \frac{z^2}{1 - z}\right) - G(z) = z + G\left(\frac{z}{1 - z}\right) - G(z). \quad (13)$$

From this equation we extract the recurrence for the number  $g_n$  of strict monotonic Schröder trees with  $n$  leaves: we get

$$g_n = [z^n] \left( z + G\left(z + \frac{z^2}{1 - z}\right) - G(z) \right) = \delta_{n,1} - g_n + \sum_{\ell \geq 1} g_\ell [z^{n-\ell}] \left( \frac{1}{1 - z} \right)^\ell.$$

We use Kronecker's notation:  $\delta_{n,1} = 1$  if  $n = 1$  and 0 otherwise. The last coefficient follows from the binomial theorem. This implies

$$g_n = \begin{cases} 1 & \text{if } n = 1, \\ \sum_{\ell=1}^{n-1} \binom{n-1}{\ell-1} g_\ell & \text{otherwise.} \end{cases} \quad (14)$$

The first terms are equal to a shift of the sequence of ordered Bell numbers (also called Fubini numbers or surjection numbers) referenced as [OEIS A000670](#):

$$(g_n)_{n \in \mathbb{N}} = (0, 1, 1, 3, 13, 75, 541, 4683, 47293, 545835, 7087261, 102247563, \dots).$$

We recall that the  $n$ -th ordered Bell number counts the number of ordered partitions of a set of size  $n$ , where an ordered partition of a set  $\mathcal{S}$  is an ordered sequence of

disjoint subsets of  $\mathcal{S}$  whose union is equal to  $\mathcal{S}$ . Ordered Bell numbers are specified by

$$B = \text{SEQ}(\text{SET}_{\geq 1} \mathcal{Z}). \quad (15)$$

Motivated by this remark, we define in Section 3.4 a bijection between the set of strict monotonic Schröder trees and the set of ordered partitions.

Following the approach developed by Pippenger in [Pip10] for ordered Bell numbers, we compute the exponential generating function of  $\mathcal{G}$ , i.e. we apply the Borel transform on  $G(z)$ . But first we recall some basic properties of the latter transform. The *Borel transform*, denoted by  $\mathcal{B}$ , takes as an argument an ordinary generating function and gives as its image the corresponding exponential generating series. More precisely, for all real-valued sequence  $(a_n)_{n \geq 0}$ , we set

$$\mathcal{B} \left[ \sum_{n \geq 0} a_n z^n \right] = \sum_{n \geq 0} a_n \frac{z^n}{n!}.$$

Note that if  $t_n \leq \rho^n n!$  for  $n$  sufficiently large then  $\mathcal{B}[T(z)]$  is analytic around 0. It is easy to check that:

**Fact 3.3.1.** *For all ordinary generating functions  $f = f(z)$ , we have*

$$(i) \mathcal{B}[zf(z)] = \int_0^z \mathcal{B}[f(t)] dt \quad \text{and} \quad (ii) \mathcal{B}[f'(z)] = (\mathcal{B}[f(z)])' + z(\mathcal{B}[f(z)])''.$$

**Proposition 3.3.2.** *The exponential generating function enumerating strict monotonic Schröder trees is*

$$\mathcal{B}[G(z)] = \frac{1}{2} (z - \ln(2 - e^z)).$$

*Proof.* From Equation (14) we obtain

$$2g_n = \delta_{n,1} + \sum_{\ell=1}^n \binom{n-1}{\ell-1} g_\ell.$$

This recurrence can be directly used to derive an equation for the exponential generating function of  $\mathcal{G}$ :

$$2\mathcal{B}[G(z)] = z + \sum_{n \geq 1} \sum_{\ell=1}^n \binom{n-1}{\ell-1} g_\ell \frac{z^n}{n!},$$

which is the classical equation satisfied by the ordered Bell numbers. Following the approach of [Pip10], we differentiate the equation with respect to  $z$  and get

$$2(\mathcal{B}[G(z)])' = 1 + \sum_{n \geq 1} \sum_{\ell=1}^n \binom{n-1}{\ell-1} g_\ell \frac{z^{n-1}}{(n-1)!}.$$

Since the sum is the convolution of  $(\mathcal{B}[G(z)])'$  with  $\exp(z)$ , we get

$$(\mathcal{B}[G(z)])' = \frac{1}{2 - e^z},$$

which implies  $\mathcal{B}[G(z)] = (z - \ln(2 - e^z))/2$  as claimed.  $\square$

Recall that ordered Bell numbers are specified by  $\mathcal{B} = \text{SEQ}(\text{SET}_{\geq 1} \mathcal{Z})$  and thus have exponential generating function  $B(z) = 1/(2 - e^z)$ . This directly implies that our sequence  $(g_n)_{n \geq 0}$  is equal to the sequence of ordered Bell numbers shifted by one, since  $B(z)$  is the derivative of  $\mathcal{B}[G(z)]$ . This link between strict monotonic trees and ordered Bell numbers has the interesting following consequence: we have shown that the (shifted) ordinary generating function of the ordered Bell numbers satisfies Equation (13). As far as we can tell, this was not known before.

The asymptotic behaviour of ordered Bell numbers is known (see, e.g., [FS09, p. 109]): if we denote by  $b_n$  the  $n$ -th ordered Bell number, then

$$b_n = \sum_{\ell=0}^n \ell! \left\{ \begin{matrix} n \\ \ell \end{matrix} \right\} \underset{n \rightarrow \infty}{\sim} \frac{n!}{2(\ln 2)^{n+1}},$$

where the  $\left\{ \begin{matrix} n \\ \ell \end{matrix} \right\}$ 's are the Stirling partition numbers (also called Stirling numbers of the second kind, see [FS09, Appendix A.8]). They count the number of ways to partition a set of  $n$  objects into  $k$  non-empty subsets.

The number  $b_n$  is equal to the number  $g_{n+1}$  of strict monotonic Schröder trees of size  $n + 1$ , which implies that, for all  $n \geq 1$ ,

$$g_n = \sum_{\ell=0}^{n-1} \ell! \left\{ \begin{matrix} n-1 \\ \ell \end{matrix} \right\} \underset{n \rightarrow \infty}{\sim} \frac{(n-1)!}{2(\ln 2)^n}.$$

**3.4. Bijection with ordered Bell numbers.** Since the number of strict monotonic Schröder trees of size  $n + 1$  is equal to the number of ordered partitions of a set of size  $n$ , it is natural to try to find an explicit bijection between the two classes. In this section, we exhibit such a bijection.

To describe precisely the bijection we need the following definitions and notations. Recall that the subsets of an ordered partition are ordered but the elements inside each subset are not. In the following, we denote by  $p = (p_1, p_2, \dots, p_\ell)$  the ordered partition of subsets  $p_1, \dots, p_\ell$ ; for example, we have  $(\{3, 4\}, \{1, 5, 7\}, \{2, 6\}) \neq (\{2, 6\}, \{3, 4\}, \{1, 5, 7\})$ . We denote by  $|p_i|$  the size of the  $i$ -th subset of  $p$ , and by  $|p| = \sum_{i=1}^{\ell} p_i$  its total size (i.e. the number of elements of  $\cup_{i=1}^{\ell} p_i$ ). Let  $a = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$  (with  $r \geq 1$ ) be a subset of  $\mathbb{N}$ ; without loss of generality, we can assume that  $\alpha_1 < \alpha_2 < \dots < \alpha_r$ . A **run** of  $a$  is a maximal sequence  $(\alpha_i, \alpha_{i+1}, \dots, \alpha_j)$  ( $1 \leq i \leq j \leq r$ ) of consecutive integers, i.e.  $(\alpha_i, \alpha_{i+1}, \dots, \alpha_j) = (\alpha_i, \alpha_i + 1, \dots, \alpha_i + j - i)$ ,  $\alpha_{i-1} < \alpha_i - 1$  and  $\alpha_{j+1} > \alpha_j + 1$ , unless  $\alpha_i = \alpha_1$  or  $\alpha_j = \alpha_r$ . We define the function **runs** as the function that lists all the runs of a subset: for example,  $\text{runs}(\{3, 4\}) = (\{3, 4\})$  and  $\text{runs}(\{1, 3, 6, 7\}) = (\{1\}, \{3\}, \{6, 7\})$ .

An ordered partition  $p = (p_1, \dots, p_\ell)$  is called *incomplete* if and only if  $\cup_{i=1}^{\ell} p_i \neq \{1, 2, \dots, |p|\}$ : e.g. the partition  $(\{3, 4\}, \{1, 5, 7\})$  is incomplete due to the fact that  $\cup_{i=1}^{\ell} p_i = \{1, 3, 4, 5, 7\} \neq \{1, 2, 3, 4, 5\}$ . We define the normalisation of a partition  $p$  (either incomplete or not), denoted by  $\text{norm}(p)$ , as the ordered partition of  $\{1, \dots, |p|\}$  that keeps the relative order between the elements. For example, if  $p = (\{3, 4\}, \{1, 5, 7\})$ , then  $\text{norm}(p) = [\{2, 3\}, \{1, 4, 5\}]$ .

We are now ready to describe our bijection: we first define the mapping  $M'$ , which associates a strict monotonic Schröder tree to each (possibly incomplete) ordered partition  $p = (p_1, \dots, p_\ell)$ . Before starting we fix an arbitrary order for the leaves in the tree once and for all (for example, the one given by the postorder traversal of the tree). Then tree  $M'(p)$  is the result of the following recursive procedure:

- At time 1, consider a tree with one internal node labelled by 1 to which are attached  $|p_1| + 1$  leaves.
- At each time  $2 \leq i \leq \ell$ , we denote by  $p'_1, \dots, p'_i$  the ordered subsets of the renormalization of  $(p_1, \dots, p_i)$ , i.e.  $\mathbf{norm}((p_1, \dots, p_i)) = (p'_1, \dots, p'_i)$ . We denote by  $r_1, \dots, r_j$  the runs of  $p'_i$ , i.e.  $\mathbf{runs}(p'_i) = (r_1, \dots, r_j)$ ; recall that each of  $r_1, \dots, r_j$  is a set of successive integers, possibly reduced to a singleton. Now, we iterate the following process: 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  $k$  attached to  $|r_k| + 1$  leaves.

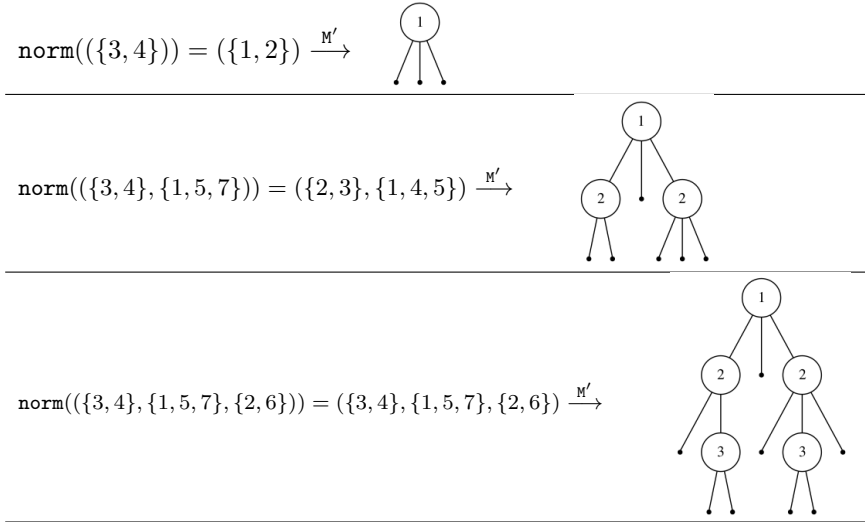


FIGURE 6. The constructive bijection between an ordered partition and a strict monotonic Schröder tree

In Figure 6 we show how to construct  $M'(p)$  when  $p = (\{3, 4\}, \{1, 5, 7\}, \{2, 6\})$ . The resulting strict monotonic Schröder tree is of size 8. It is straightforward to check that  $M'$  is indeed a bijection.

**3.5. Analysis of typical parameters.** In this section, we give information about the shape of a *typical* strict monotonic Schröder tree: more precisely, we prove limit theorems for the number of distinct labels and the arity of the root as well as the average number of internal nodes in a tree picked uniformly at random among all strict monotonic Schröder trees of size  $n$  (i.e. with  $n$  leaves).

**3.5.1. Quantitative analysis of the number of iteration steps.** The main novelty of strict monotonic Schröder trees compared to increasing Schröder trees is that repetitions of labels are allowed: it is thus natural to ask how many repetitions there are in a *typical* strict monotonic Schröder tree. To answer this question, one can mark iterations by adding a new variable  $u$  in Equation (13):

$$G(z, u) = z + u G\left(\frac{z}{1-z}, u\right) - u G(z, u),$$

which implies

$$g_{n,k} = \begin{cases} 1 & \text{if } n = 1 \text{ and } k = 0, \\ \sum_{\ell=1}^{n-1} \binom{n-1}{\ell-1} g_{\ell,k-1} & \text{otherwise,} \end{cases} \quad (16)$$

with  $n$  being the size and  $k$  the number of iteration steps (i.e. the number of distinct labels). In Figure 7, we show the first values of  $(g_{n,k})$  that are stored in [OEIS A019538](#).

1,
0, 1,
0, 1, 2,
0, 1, 6, 6,
0, 1, 14, 36, 24,
0, 1, 30, 150, 240, 120,
0, 1, 62, 540, 1560, 1800, 720

FIGURE 7. Distribution of  $(g_{n,k})_k$  for  $n \in \{1, \dots, 7\}$

This recurrence is analogous to the one relating ordered Bell numbers and Stirling partition numbers (see Equation (14)).

**Theorem 3.5.1.** *The number of strict monotonic Schröder trees of size  $n$  with exactly  $k$  distinct labels is given by*

$$g_{n,k} = k! \begin{Bmatrix} n-1 \\ k \end{Bmatrix}.$$

We denote by  $X_n^{\mathcal{G}}$  the number of distinct labels in a tree picked uniformly at random among all strict monotonic Schröder trees of size  $n+1$ : for all  $n \geq 1$ ,  $X_n^{\mathcal{G}}$  is a random variable such that  $\mathbb{P}(X_n^{\mathcal{G}} = k) = g_{n,k} / \sum_{k=1}^n g_{n,k}$ . Then, asymptotically when  $n$  tends to infinity,

$$\frac{X_n^{\mathcal{G}} - \frac{n}{2 \ln 2}}{\sqrt{\frac{(1 - \ln 2)n}{(2 \ln 2)^2}}} \xrightarrow{d} \mathcal{N}(0, 1).$$

The analysis of the limiting distribution is classical in the quasi-powers framework established by Hwang [Hwa98]; see [FS09, p. 645, 653] for details and applications.

*Proof.* Recall that  $g_{n,k} = k! \begin{Bmatrix} n-1 \\ k \end{Bmatrix}$  is the number of ordered partitions of a set of size  $n$  having  $k$  non-empty parts. It is known (see, e.g. [Ben73, Example 3.4]) that, if  $K_n$  is the number of parts in an ordered set partition of size  $n$ , then

$$\frac{K_n - \frac{n}{2 \ln 2}}{\sqrt{\frac{(1 - \ln 2)n}{(2 \ln 2)^2}}} \xrightarrow{d} \mathcal{N}(0, 1),$$

in distribution. This concludes the proof since  $K_n$  has the same distribution as  $X_n^{\mathcal{G}}$  for all  $n \geq 1$ .  $\square$

3.5.2. *Quantitative analysis of the number of internal nodes.* In this model the number of internal nodes is different from the number of distinct labels that appear in the tree: this is because one integer can label several internal nodes. It is thus natural to ask how many internal nodes a typical strict monotonic Schröder trees of size  $n$  (i.e. with  $n$  leaves) has. The specification marking both leaves (with variable  $z$ ) and internal nodes (with variable  $u$ ) is

$$G(z, u) = z + G\left(z + \frac{uz^2}{1-z}, u\right) - G(z, u). \quad (17)$$

We recall that the substitution  $z \rightarrow z + \frac{uz^2}{1-z}$  means that at each iteration each leaf can be left as it is ( $z \rightarrow z$ ) or expanded into an internal node attached to an arbitrary number of leaves ( $z \rightarrow \frac{z^2}{1-z}$ ). A new internal, marked with the variable  $u$ , is created only in the second case.

1,
0, 1,
0, 1, 2,
0, 1, 5, 7,
0, 1, 9, 31, 34,
0, 1, 14, 86, 226, 214
0, 1, 20, 190, 874, 1946, 1652

FIGURE 8. Distribution of  $(g_{n,k})_k$  for  $n \in \{1, \dots, 7\}$

For all  $1 \leq n$  and  $1 \leq k \leq n-1$ , we denote by  $g_{n,k}$  the number strict monotonic Schröder trees with  $n$  leaves and  $k$  internal nodes: Figure 8 shows the values of  $(g_{n,k})_{1 \leq k \leq n-1}$  for  $n \in \{1, 2, \dots, 7\}$ . This triangle of integers is not yet stored in OEIS. However, its diagonal is equal to [OEIS A171792](#). In fact in the diagonal the numbers corresponds to the number of strict monotonic trees with  $n$  leaves and  $n-1$  internal nodes, i.e. binary strict monotonic trees: this class of trees is studied in [BGGW20].

**Theorem 3.5.2.** *If we denote by  $I_n^{\mathcal{G}}$  the (random) number of internal nodes in a tree picked uniformly at random among all strict monotonic Schröder trees of size  $n$ , then, asymptotically when  $n$  tends to infinity,*

$$\mathbb{E}[I_n^{\mathcal{G}}] \underset{n \rightarrow \infty}{=} n - (\ln 2)(\ln n) + \frac{\pi^2}{12} - 1 + (\ln 2) \left( -\gamma + \frac{\ln 2}{2} + \ln \ln 2 \right) + o(1),$$

where  $\gamma$  is the Euler-Mascheroni constant.

*Proof.* For all  $n \geq 1$ , we denote by  $h_n = \sum_{k=1}^{n-1} k g_{n,k}$ , and let  $H$  be the ordinary generating function of  $(h_n)_{n \geq 1}$ ; we have

$$H(z) = \left( \frac{\partial G(z, u)}{\partial u} \right)_{|u=1}.$$

The ratio  $h_n/g_n$  is equal to the expected number of internal nodes in a tree taken uniformly at random among all strict monotonic Schröder trees of size  $n$ ; we are

thus interested in the asymptotic behaviour of this ratio. Differentiating according to  $u$  and then substituting  $u$  by 1 in Equation (17) gives

$$H(z) = \frac{z^2}{1-z} G' \left( \frac{z}{1-z} \right) + H \left( \frac{z}{1-z} \right) - H(z), \quad (18)$$

because

$$\left( \frac{\partial G(z, u)}{\partial z} \right)_{|u=1} = G'(z).$$

Since Equation (18) is similar to Equation (13), we apply the same method as in the proof of Proposition 3.3.2. We first derive

$$(\mathcal{B}[H(z)])' = \frac{1}{2 - e^z} \left( \mathcal{B} \left[ \frac{z^2}{1-z} G' \left( \frac{z}{1-z} \right) \right] \right)'$$

Then using Equation (13) we deduce

$$\left( \mathcal{B} \left[ \frac{z^2}{1-z} G' \left( \frac{z}{1-z} \right) \right] \right)' = -z + \frac{z^2}{2} + 2 (\mathcal{B} [z^2(1-z)G'(z)])'$$

Furthermore since for any function  $F$  we have  $\mathcal{B}[zF(z)] = \int_0^z \mathcal{B}[F(t)]dt$  see Fact 3.3.1, we can simplify the equation into

$$(\mathcal{B}[H(z)])' = \frac{1}{2 - e^z} \left( -z + \frac{z^2}{2} + 2 \int_0^z \mathcal{B}[G'(t)] dt - 2 \int_0^z \int_0^t \mathcal{B}[G'(u)] du dt \right).$$

Then, since  $\int_0^z \mathcal{B}[G'(t)]dt = z(\mathcal{B}[G(z)])'$ , we obtain

$$\begin{aligned} (\mathcal{B}[H(z)])' &= \frac{1}{2 - e^z} \left( -z + \frac{z^2}{2} + 2z(\mathcal{B}[G(z)])' - 2 \int_0^z t(\mathcal{B}G(t))' dt \right) \\ &= \frac{1/2}{1 - e^z/2} \left( -\frac{\pi^2}{12} + \frac{(\ln 2)^2}{2} - z \left( 1 - \ln(1 - e^z/2) - \frac{1}{1 - e^z/2} \right) + \text{Li}_2(e^z/2) \right), \end{aligned}$$

where  $\text{Li}_2$  is the dilogarithm function, defined in [FS09, section VI.8.]. Using its asymptotic development at 1, we get

$$\begin{aligned} (\mathcal{B}[H(z)])' &\underset{z \rightarrow \ln 2}{\sim} \frac{1}{2 \ln 2} \frac{1}{(1 - z/\ln 2)^2} \\ &\quad - \left( \frac{1}{2 \ln 2} - \frac{\pi^2}{24 \ln 2} + \frac{\ln 2}{4} - \frac{\ln 2 + \ln \ln 2 + \ln(1 - z/\ln 2)}{2} \right) \frac{1}{1 - z/\ln 2} \\ &\quad - \frac{1}{2} - \frac{7 \ln 2}{24} + \frac{\pi^2}{48} + \frac{(\ln 2)^2}{8} + \frac{\ln 2 \ln \ln 2}{4} + O \left( \ln \left( \frac{1}{1 - z/\ln 2} \right) \right). \end{aligned}$$

By using classical transfer theorems we obtain the result by extracting the  $(n-1)$ -st coefficient of  $(\mathcal{B}[H(z)])'$  and dividing it by the  $n$ -th coefficient of  $\mathcal{B}[G(z)]$ .  $\square$

**3.5.3. Quantitative characteristics of the root node.** In this section, we look at the arity of the root in a typical strict monotonic Schröder tree. We denote by  $A_n^G$  the arity of the root in a tree picked uniformly at random among all strict monotonic Schröder trees of size  $n$ , and by  $p_n$  its probability generating function:

$$p_n(u) = \sum_{k \geq 0} \mathbb{P}(A_n^G = k) u^k.$$

**Theorem 3.5.3.** *Asymptotically when  $n$  tends to infinity,  $A_n^{\mathcal{G}}$  converges in distribution to a (shifted) zero-truncated Poisson law with parameter  $\ln 2$ , i.e. for all  $u \geq 0$ ,*

$$p_n(u) \xrightarrow[n \rightarrow \infty]{} ue^{u \ln 2} - u.$$

*This implies that  $\mathbb{E}[A_n^{\mathcal{G}}] \rightarrow 2 \ln 2 + 1$  when  $n$  tends to infinity.*

*Proof.* Thanks to the bijection of Section 3.4, we know that  $A_n^{\mathcal{G}}$  is equal to the size of the first subset  $+1$  in an ordered partition picked uniformly at random among all ordered partitions of  $\{1, \dots, n-1\}$ . We denote by  $\mathcal{P}$  the class of ordered partitions,  $1$  is the empty partition,  $\mathcal{Z}$  is a singleton, and  $\mathcal{U}$  marks the elements in the first subset. Here the specification is defined in the context of labelled object, thus the associated generating functions are exponential (see [FS09] for notation details):

$$\mathcal{P} = 1 + \text{SET}_{\geq 1}(\mathcal{U}\mathcal{Z}) \star \text{SEQ}(\text{SET}_{\geq 1} \mathcal{Z}).$$

Using the symbolic method for exponential generating function, we get

$$P(z, u) = 1 + \frac{e^{uz} - 1}{2 - e^z}.$$

Thus, if we set

$$\tilde{p}_n(u) = \frac{[z^n]P(z, u)}{[z^n]P(z, 1)},$$

for all  $n \geq 0$ , then

$$[z^n]P(z, u) \xrightarrow[n \rightarrow \infty]{} \frac{1}{2} (2^u - 1) (\ln 2)^{-n-1}.$$

This implies that, for all  $u \geq 0$ ,

$$\tilde{p}_n(u) \xrightarrow[n \rightarrow \infty]{} 2^u - 1.$$

Note that, by definition,  $\tilde{p}_n(u)$  is the probability generating function of the size  $S_n$  of the first subset in an ordered partition picked uniformly at random among all ordered partitions of  $\{1, \dots, n-1\}$ . Because of the bijection of Section 3.4, we know that  $A_n^{\mathcal{G}}$  and  $S_{n-1} + 1$  have the same distribution, implying that  $p_n(u) = u\tilde{p}_n(u)$ . This concludes the proof.  $\square$

**3.6. Uniform random sampling.** To sample uniformly at random a strict monotonic Schröder tree of size  $n$ , we could choose a two-step algorithm. First we sample uniformly an ordered partition of the set  $\{1, \dots, n-1\}$  and then with the use of the bijection of Section 3.4 we transform it into a strict monotonic Schröder tree. But here, in this section, we prefer to present a direct algorithm that generates uniformly a strict monotonic Schröder tree, i.e. without the intermediate step of generating another combinatorial object like an ordered partition.

The global approach for our algorithmic framework deals with the *recursive generation method* adapted to the analytic combinatorics point of view in [FZVC94]. But in our context we note that we can obtain for free (from a complexity view) an *unranking algorithm*. This fact is sufficiently rare to mention it: usually unranking algorithm are less efficient than recursive generation ones. Unranking algorithm has been developed in the 70's by Nijenhuis and Wilf [NW75] and then has been introduced to the context of analytic combinatorics by Martínez and Molinero [MM03]. Here the idea is not to draw uniformly an object, but first to define a total order over the objects under consideration (in our context, strict

monotonic Schröder trees) and then an integer (named the rank) is chosen to build deterministically the associated object. Obviously if the rank is uniformly chosen among all possible ranks, then the unranking algorithm is nothing else than a uniformly random sampler. But the unranking approach gives also a way for obtaining an exhaustive sampler, just by iterating the sampling over all possible ranks (the reader can refer to the paper [BDGV18] for an example of both methods: recursive generation and unranking).

For both types of algorithms (unranking or recursive generation) some pre-computations are done (only once before the sampling of many objects). We compute (and store) the numbers of trees of sizes from 1 to  $n$ . This calculation is done with a quadratic complexity (in the number of arithmetic operations) using the recursive formula for  $(g_n)_{n \geq 1}$  (see Equation (14)). This complexity is only achieved if we first compute and memorise all values of  $(i!)_{1 \leq i \leq n}$ .

Then it only remains to build the tree of rank  $r$  recursively. If  $r$  is sampled uniformly at random in  $\{0, 1, \dots, g_n - 1\}$  the algorithm is a uniform sampler and if  $r$  is deterministically chosen, then the algorithm is a classical unranking algorithm. To do this, we recall that (see Equation (14)), for all  $n \geq 1$ ,

$$g_n = \binom{n-1}{n-2} g_{n-1} + \binom{n-1}{n-3} g_{n-2} + \dots + \binom{n-1}{0} g_1, \quad (19)$$

and interpret this equation combinatorially: to build a tree of size  $n$ , we take a size  $\ell \in \{1, \dots, n-1\}$  tree  $T_\ell$  constructed with exactly one less iteration. To grow it into a size- $n$  tree, we interpret the binomial coefficient  $\binom{n-1}{\ell-1}$  as the number of composition of  $n$  in  $\ell$  parts: some of the  $\ell$  leaves of  $T_\ell$  are replaced by internal nodes to which leaves are attached, some leaves remain leaves. To do that we traverse the tree  $T_\ell$  and each time we see a leaf, we do the following action: if the next part (in the composition) is of value 1, we keep the leaf unchanged otherwise for a value  $s > 1$ , we replace the leaf by an internal node (labelled with the currently step number) and attached  $s$  leaves to it. We then take the next part of the composition into consideration and continue the tree traversal.

Focusing on Equation (19) and the equation above we see that a function allowing the unranking of compositions is necessary. Recall that the composition of the integer  $n$  into  $\ell$  parts is in bijection with the number of sets of combinations of  $(\ell-1)$  elements chosen in  $(n-1)$  ones. A way to prove this consists in laying  $(\ell-1)$  barriers in the sequence of  $n$  bullets in order to define  $\ell$  parts. There are classical algorithms to unrank combinations in the lexicographical order. An algorithm has been described by Buckles and Lybanon [BL77]. Furthermore a survey about such an approach has recently been published [GP21]. For these algorithms we can easily prove that their average complexity (when  $\ell$  ranges over all possibilities) is  $\Theta(n)$  in the number of arithmetic operations by having first memoized all factorial values of the numbers from 0 to  $n$ . In the following we develop a simpler approach based on the classical recursive generation without any lexicographic constraint. The algorithm is an unranking method for the composition of integers. It is based on the reverse lexicographic order (cf e.g. [Rus03]) so that we get an easier implementation<sup>2</sup>. For simplification, we suppose having memoized all values of  $\binom{r}{s}$  for  $r \in \{1, n\}$  and  $s \in \{1, r\}$ . Using the classical Pascal's rule for binomial coefficients,

<sup>2</sup>For the composition unranking, note that it would suffice to look for the rank  $\binom{n}{\ell} - 1 - r$  (instead of  $r$ ) in order to get the lexicographic order.

we obtain the following recurrence for the number of composition of  $n$  into  $\ell$  parts:

$$C_{n,\ell} = \binom{n-1}{\ell-1} = C_{n-1,\ell} + C_{n-1,\ell-1}. \quad (20)$$

We thus deduce Algorithm 2 for the unranking method where the union symbol ' $\cup$ ' used in the algorithm represents list concatenation.

---

**Algorithm 2** Reverse Lexicographic Composition Unranking

---

```

1: function UNRANKCOMPOSITION( $n, \ell, r$ )
2:   if  $n = \ell$  and  $r = 0$  then
3:     return  $(1, 1, \dots, 1)$ 
4:   if  $r < \binom{n-2}{\ell-1}$  then
5:      $C := \text{UNRANKCOMPOSITION}(n-1, \ell, r)$ 
6:      $C[0] := C[0] + 1$ 
7:     return  $C$ 
8:   else
9:      $s := r - \binom{n-2}{\ell-1}$ 
10:     $C := (1) \cup \text{UNRANKCOMPOSITION}(n-1, \ell-1, s)$ 
11:    return  $C$ 

```

---

**Theorem 3.6.1.** *The function UNRANKCOMPOSITION is an unranking algorithm (based on the reverse lexicographic order) and calling it with the parameters  $\ell \leq n$  and a uniformly-sampled integer  $r$  in  $\{0, \dots, \binom{n-1}{\ell-1} - 1\}$ , gives as output a uniform composition of  $n$  into  $\ell$  parts.*

*Using the memorisation of binomial coefficients, the algorithm needs at most  $(\ell - 1)$  arithmetic operations on big integers.*

The function UNRANKCOMPOSITION manipulates numbers upper bounded by  $\binom{n-1}{\ell-1} \leq 2^n$ . Thus the memory size of the big integers, that are used during the calculations, is  $O(n)$  bits.

*Proof.* We prove that the algorithm is correct by induction on  $n$ . The result is true when  $n = \ell$  and  $r = 0$  since the algorithm returns  $(1)$ . Fix an integer  $n$  and assume that the algorithm is correct for all  $\ell \leq n - 1$ , and that the total order over compositions is the reverse lexicographic one (see, e.g., [Rus03] for the definition of the reverse lexicographic order). Let  $\ell$  be an integer between 0 and  $n$ , and  $r$  be an integer chosen uniformly at random in  $\{0, \dots, C_{n,\ell} - 1\}$ . Equation (20) implies that a composition of  $n$  in  $\ell$  parts is either a composition of  $(n - 1)$  in  $\ell$  parts whose first part has been increased by one, or it is a composition of  $(n - 1)$  in  $(\ell - 1)$  parts, and a new part equal to 1 is added at the beginning of the composition. In both cases, the first elements are all greater than the second elements according to the lexicographic order. The recurrence hypothesis ends the proof since the rank value  $r$  (or  $s$  in the second case) is adapted to each of the latter cases.

The number of arithmetic operations is immediate when all binomial coefficients are first memorised.  $\square$

In Equation (19) the first term is much bigger than the second one, which is much bigger than the third one and so on. This approach, focusing first on the dominant terms is an adaptation to the idea underlying the *Boustrophedonic order* presented in [FZVC94]. It allows to improve essentially the average complexity of the random sampling algorithm. In our case of strict monotonic Schröder trees since they do

not follow a standard specification (cf. [FZVC94] for details), the complexity gain is even better. The loop starting in line 6 aims at determining the interesting term

---

**Algorithm 3** Strict monotonic Schröder Tree Unranking
 

---

```

1: function UNRANKTREE( $n, s$ )
2:   if  $n = 1$  then
3:     return the tree reduced to a single leaf
4:    $\ell := 1$ 
5:    $r := s$ 
6:   while  $r >= 0$  do
7:      $r := r - \binom{n-1}{\ell} \cdot g_{n-\ell}$ 
8:      $\ell := \ell + 1$ 
9:    $\ell := \ell - 1$ 
10:   $r := r + \binom{n-1}{\ell} \cdot g_{n-\ell}$ 
11:   $T := \text{UNRANKTREE}(n - \ell, r \bmod g_{n-\ell})$ 
12:  if  $n - \ell \leq n/2$  then
13:     $C := \text{UNRANKCOMPOSITION}(n, n - \ell, r // g_{n-\ell})$ 
14:  else
15:     $C := \text{UNRANKCOMPOSITION}(n, \ell + 1, ((\binom{n-1}{\ell} - 1 - r) // g_{n-\ell})$ 
16:     $C := \text{COMPLEMENTCOMPOSITION}(C)$ 
17:  Substitute in  $T$ , using traversal  $\mathcal{T}$ , some leaves according to  $C$ 
18:  return the tree  $T$ 
    
```

The sequences  $(g_\ell)_{\ell \leq n}$  and  $(\ell!)_{\ell \in \{1, \dots, n\}}$  have been pre-computed and stored.

Line 13: The operation  $//$  is the Euclidean division.

Recall a composition of  $(n, \ell)$  can be seen as a choice of  $\ell - 1$  barriers inside a sequence of  $n$  bullets. The function  $\text{COMPLEMENTCOMPOSITION}(n, \ell, r)$  returns the complement of the barriers chosen in the  $r$ -th composition of  $(n, \ell)$ .

---

in the sum (19), thus the size of the tree in the evolution process letting to build the tree of rank  $s$  and size  $n$ .

The traversal  $\mathcal{T}$  used to substitute some leaves in line 13 determines partly the total order over the strict monotonic trees. Let  $\alpha$  be an strict monotonic tree, and  $\mathcal{T}$  a given traversal of all trees. We remark that there is a single evolution process building  $\alpha$  (the construction is unambiguous). If  $\alpha$  is built at the step  $\ell$ , then we denote by  $\tilde{\alpha}$  the single tree (built with  $\ell - 1$  steps) and  $\underline{\alpha}$  the single composition such that at step  $\ell$  replacing the leaves from  $\tilde{\alpha}$  according to the composition  $\underline{\alpha}$ , using the traversal  $\mathcal{T}$ , we obtain  $\alpha$ .

Here we remark that the whole tree  $\alpha$  is strongly dependent on the traversal of the leaves of  $\tilde{\alpha}$  (while some leaves are substituted by an internal nodes attached to new leaves according to  $\underline{\alpha}$ ). We define now how to compare strict monotonic trees (we use the analogous notations also for the latter for all trees).

**Definition 3.6.2.** *Let  $\alpha$  and  $\beta$  be two trees. We define  $\alpha < \beta$  if*

- *the size of  $\alpha$  is smaller than the one of  $\beta$ , or*
- *if both sizes are equal to  $n$  and if the size of  $\tilde{\alpha}$  is strictly greater than the one of  $\tilde{\beta}$  or if both sizes of  $\tilde{\alpha}$  and  $\tilde{\beta}$  are equal and the composition  $\underline{\alpha}$  is smaller than  $\underline{\beta}$ , using the reverse lexicographic order over compositions.*

**Proposition 3.6.3.** *The order defined over strict monotonic trees is a total order.*

The result is immediate since all possible cases according to the trees  $\alpha$  and  $\beta$  for comparing them are explored.

**Theorem 3.6.4.** *The function UNRANKTREE is an unranking algorithm and calling it with the parameters  $n$  and a uniformly-sampled integer  $s$  in  $\{0, \dots, g_n - 1\}$  gives as output a uniform strict monotonic Schröder tree of size  $n$ .*

The correctness of the algorithm follows directly from the total order over the trees and Equation (19).

**Theorem 3.6.5.** *Once the pre-computations have been done, the function UNRANKTREE needs  $O(n)$  arithmetic operations on big integers to construct a tree of size  $n$ .*

Here the algorithm works with number of order  $g_n$ , thus the big integers that are taken into account are encoded with  $O(n \log n)$  bits.

*Proof.* We assume that all binomial coefficients  $(C_{n,\ell})_{0 \leq \ell \leq n}$  have been memorised and prove that, with this information stored, the complexity in terms of arithmetic operations (on big integers) is of order  $O(n)$ . Note that if we only memorise the factorial numbers  $(i!)_{0 \leq i \leq n-1}$  the complexity is at most three times the complexity obtained when memorising the binomial coefficient and thus still of order  $O(n)$ .

For all  $n \geq 1$ , we denote by  $a_n$  the number of arithmetic operations that come from the loop in line 7 and the calls in lines 11 and 12, when building all trees of size  $n$  (i.e. we sum the number needed for each  $r \in \{0, \dots, C_{n,\ell} - 1\}$ ). The exact value of arithmetic operations is  $a_n + O(n g_n)$ , because at each recursive call there is at most a constant number of operations that are not counted in  $a_n$ . We first analyse  $a_n$ : we have

$$a_n = \sum_{\ell=1}^{n-1} \binom{n-1}{\ell} ((\min(\ell, n-1-\ell) - 1 + 2\ell) g_{n-\ell} + a_{n-\ell}).$$

In fact, for the terms with index  $\ell$ , we are interested in the trees  $\alpha$  of size  $n$  such that their corresponding tree  $\tilde{\alpha}$  is of size  $n - \ell$ . Thus such trees  $\alpha$  are counted by  $\binom{n-1}{\ell} g_{n-\ell}$ . And for each of them the factor  $\min(\ell, n-1-\ell) - 1$  is the number of operations needed for the unranking of the composition (we use the symmetry in the binomial coefficients), the factor  $2\ell$  is the number of multiplication and subtractions in the loop in line 7. Furthermore we have  $a_1 = 0$ . By taking an upper bound for the min function, we get that if  $\bar{a}_1 = 0$ , and, for all  $n \geq 2$ ,

$$\bar{a}_n = \sum_{\ell=1}^{n-1} \binom{n-1}{\ell} (3\ell g_{n-\ell} + \bar{a}_{n-\ell}),$$

then  $a_n \leq \bar{a}_n$  for all  $n \geq 1$ . Using similar calculations as in the proof of Proposition 3.3.2, we obtain an equation satisfied by the Borel transform of the series associated to  $(\bar{a}_n)$ :

$$2(\mathcal{B}[\bar{A}(z)])' = e^z (\mathcal{B}[\bar{A}(z)])' + 3ze^z (\mathcal{B}[G(z)])'.$$

We thus deduce  $\bar{a}_n \sim 3ng_n$ , which concludes the proof.  $\square$

We now give some final remarks about this algorithm: In order to obtain a better time complexity for the implementation, we must use an array of pointers to the leaves of the tree under construction so that the tree traversal is efficient. At each step  $\ell$ , a leaf stored in the array is replaced by  $n - \ell + 1$  leaves that must be stored in the array. An efficient way consists in reusing the cell from the replaced leaf, and to append all other leaves at the end on the array. Thus, the most efficient traversal

$\mathcal{T}$  of the leaves consists to the left right traversal of the array. But obviously this is not really a natural traversal for the tree. Nevertheless, in practice we use this efficient traversal  $\mathcal{T}$ .

**Acknowledgement.** The authors Hsien-Kuei Hwang for his help with the proof of Theorem 2.4.5 and for telling us about the links of this proof with [CHY00]. The authors thank the anonymous referees for their comments and suggested improvements. All these remarks have increased the quality of the paper.

## REFERENCES

- [Ald96] D. Aldous. Probability distributions on cladograms. In D. Aldous and R. Pemantle, editors, *Random Discrete Structures*, pages 1–18. Springer New York, 1996.
- [BBHT17] A. Bacher, O. Bodini, H.-K. Hwang, and T.-H. Tsai. Generating random permutations by coin tossing: Classical algorithms, new analysis, and modern implementation. *ACM Trans. Algorithms*, 13(2):24:1–24:43, 2017.
- [BDF<sup>+</sup>16] O. Bodini, M. Dien, X. Fontaine, A. Genitrini, and H.-K. Hwang. Increasing Diamonds. In *LATIN 2016: Theoretical Informatics - 12th Latin American Symposium*, pages 207–219, 2016.
- [BDGV18] O. Bodini, M. Dien, A. Genitrini, and A. Viola. Beyond series-parallel concurrent systems: the case of arch processes. In *AofA '18*, pages 14:1–14:14, 2018.
- [Ben73] E. A. Bender. Central and local limit theorems applied to asymptotic enumeration. *Journal of Combinatorial Theory, Series A*, 15(1):91–111, 1973.
- [BFS92] F. Bergeron, P. Flajolet, and B. Salvy. Varieties of increasing trees. In *CAAP*, pages 24–48, 1992.
- [BGW20] O. Bodini, A. Genitrini, B. Gittenberger, and S. Wagner. On the number of increasing trees with label repetitions. *Discrete Mathematics*, 343(8):111722, 2020.
- [BGN19] O. Bodini, A. Genitrini, and M. Naima. Ranked schröder trees. In *Sixteenth Workshop on Analytic Algorithmics and Combinatorics, ANALCO 2019*, pages 13–26, 2019.
- [BGNS20] O. Bodini, A. Genitrini, M. Naima, and A. Singh. Families of Monotonic Trees: Combinatorial Enumeration and Asymptotics. In *15th International Computer Science Symposium in Russia (CSR)*, page To appear, 2020.
- [BL77] B. P. Buckles and M. Lybanon. Algorithm 515: Generation of a vector from the lexicographical index [g6]. *ACM Trans. Math. Softw.*, 3(2):180–182, June 1977.
- [BRS12] O. Bodini, O. Roussel, and M. Soria. Boltzmann samplers for first-order differential specifications. *Discrete Applied Mathematics*, 160(18):2563–2572, 2012.
- [CHY00] H.-H. Chern, H.-K. Hwang, and Y.-N. Yeh. Distribution of the number of consecutive records. *Random Structures & Algorithms*, 17(3-4):169–196, 2000.
- [Dev90] L. Devroye. On the height of random m-ary search trees. *Random Struct. Algorithms*, 1(2):191–204, 1990.
- [Drm09] M. Drmota. *Random trees*. Springer, Vienna-New York, 2009.
- [FS09] P. Flajolet and R. Sedgewick. *Analytic combinatorics*. Cambridge University Press, Cambridge, 2009.
- [FZVC94] P. Flajolet, P. Zimmermann, and B. Van Cutsem. A calculus for the random generation of labelled combinatorial structures. *Theoretical Computer Science*, 132(1-2):1–35, 1994.
- [GGKW20] A. Genitrini, B. Gittenberger, M. Kauers, and M. Wallner. Asymptotic enumeration of compacted binary trees of bounded right height. *Journal of Combinatorial Theory, Series A*, 172:105177, 2020.
- [GP21] A. Genitrini and M. Pépin. Lexicographic Unranking of Combinations Revisited. *Algorithms*, 14(3), 2021.
- [Hwa98] H.-K. Hwang. On Convergence Rates in the Central Limit Theorems for Combinatorial Structures. *Eur. J. Comb.*, 19(3):329–343, 1998.
- [MM03] C. Martínez and X. Molinero. Generic algorithms for the generation of combinatorial objects. In *28th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 572–581. Springer Berlin Heidelberg, 2003.

- [Mol05] X. Molinero. *Ordered Generation of Classes of Combinatorial Structures*. Phd thesis, Universitat Politècnica de Catalunya, 2005.
- [Moo74] J. W. Moon. *The distance between nodes in recursive trees*, pages 125–132. London Mathematical Society Lecture Note Series. Cambridge University Press, 1974.
- [NW75] A. Nijenhuis and H. S. Wilf. *Combinatorial algorithms*. Computer science and applied mathematics. Academic Press, New York, NY, 1975.
- [Pip10] N. Pippenger. The hypercube of resistors, asymptotic expansions, and preferential arrangements. *Mathematics Magazine*, 83(5):331–346, 2010.
- [Pit94] B. Pittel. Note on the heights of random recursive trees and random  $m$ -ary search trees. *Random Structures & Algorithms*, 5(2):337–347, 1994.
- [Rus03] Frank Ruskey. Combinatorial generation. *Preliminary working draft*. University of Victoria, Victoria, BC, Canada, 11:20, 2003.
- [Sch70] E. Schröder. Vier Combinatorische Probleme. *Z. Math. Phys.*, 15:361–376, 1870.
- [SDH<sup>+</sup>04] C. Semple, P. Daniel, W. Hordijk, R.D.M. Page, and M. Steel. Supertree algorithms for ancestral divergence dates and nested taxa. *Bioinformatics*, 20(15):2355–2360, 2004.
- [SM01] M. Steel and A. McKenzie. Properties of phylogenetic trees generated by yule-type specification models. *Mathematical Biosciences*, 170(1):91–112, 2001.
- [Yul25] G. U. Yule. A mathematical theory of evolution, based on the conclusions of Dr. J. C. Willis, F.R.S. *Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of a Biological Character*, 213:21–87, 1925.

OLIVIER BODINI AND MEHDI NAIMA. UNIVERSITÉ SORBONNE PARIS NORD, LABORATOIRE D'INFORMATIQUE DE PARIS NORD, CNRS, UMR 7030, F-93430, VILLETANEUSE, FRANCE.

*Email address:* {Olivier.Bodini, Mehdi.Naima}@lipn.univ-paris13.fr

ANTOINE GENITRINI. SORBONNE UNIVERSITÉ, CNRS, LABORATOIRE D'INFORMATIQUE DE PARIS 6 -LIP6- UMR 7606, F-75005 PARIS, FRANCE.

*Email address:* Antoine.Genitrini@lip6.fr

CÉCILE MAILLER. UNIVERSITY OF BATH, DEPARTMENT OF MATHEMATICAL SCIENCES, CLAVERTON DOWN, BA2 7AY BATH, UK.

*Email address:* C.Mailler@bath.ac.uk