Combinatorics of increasing trees:

Bijections, asymptotics and algorithms

Mehdi Naima
Under the supervision of
Olivier Bodini and Antoine Genitrini
In fulfilment of the degree Doctor of Philosophy of
Université Sorbonne Paris-Nord

Reviewer:

Vlady Ravelomanana	Université de Paris
Julien Clément	Université de Caen
Alois Panholzer	Technische Universität Wien

Jury:

Frédérique Bassino	Université Sorbonne Paris Nord
Olivier Bodini	Université Sorbonne Paris Nord
Julien Clément	Université de Caen
Antoine Genitrini	Sorbonne Université
Cécile Mailler	University of Bath
Cyril Nicaud	Université Gustave Eiffel
Alois Panholzer	Technische Universität Wien
Vlady Ravelomanana	Université de Paris

18 December 2020

Table of contents

Introduction

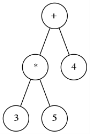
Tree structures

Tree structures are used as abstract data type to represent hierarchical relations between information that it contains. Tree structures appears extensively in computer science:

Tree structures

Tree structures are used as abstract data type to represent hierarchical relations between information that it contains. Tree structures appears extensively in computer science:

- In compilation abstract syntax trees represent the abstract syntactic structure of a source code written in a programming language.

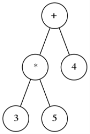


Abstract syntax tree

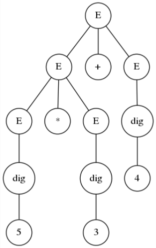
Tree structures

Tree structures are used as abstract data type to represent hierarchical relations between information that it contains. Tree structures appears extensively in computer science:

- In compilation abstract syntax trees represent the abstract syntactic structure of a source code written in a programming language.
- In computational linguistics, a parse tree represents the syntactic structure of a string according to some context-free grammar.



Abstract syntax tree

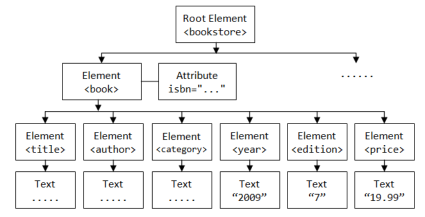


Its corresponding parse tree

Tree structures

Tree structures are used as abstract data type to represent hierarchical relations between information that it contains. Tree structures appears extensively in computer science:

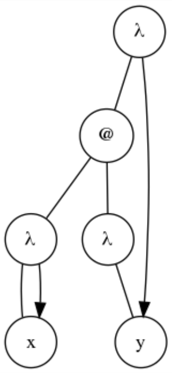
- In compilation abstract syntax trees represent the abstract syntactic structure of a source code written in a programming language.
- In computational linguistics, a parse tree represents the syntactic structure of a string according to some context-free grammar.
- Markup languages such as XML that have underlying tree structures that can be used and processed subsequently by the *Document Object Model*.



Tree structures

Tree structures are used as abstract data type to represent hierarchical relations between information that it contains. Tree structures appears extensively in computer science:

- In compilation abstract syntax trees represent the abstract syntactic structure of a source code written in a programming language.
- In computational linguistics, a parse tree represents the syntactic structure of a string according to some context-free grammar.
- Markup languages such as XML that have underlying tree structures that can be used and processed subsequently by the *Document Object Model*.
- Lambda terms in lambda calculus are enriched tree structures.

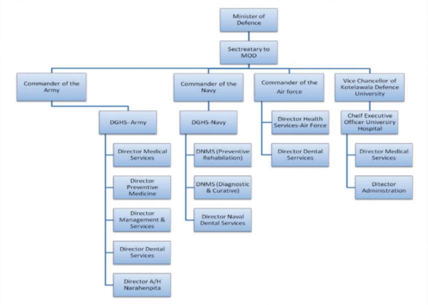


Tree structures

Tree structures also appear in a many other domains:

Tree structures

Tree structures also appear in a many other domains:



The contemporary structure of the Sri Lankan military health care services. (military-medicine)

Tree structures

Tree structures also appear in a many other domains:

- In biology and phylogenetics to represent the evolutionary relationship among species. [Fel03, Ste16]

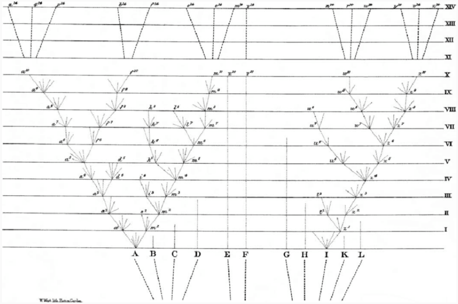


Diagram of divergence of Taxa 1871. Darwin (On the origin of species)

Tree structures

Tree structures also appear in a many other domains:

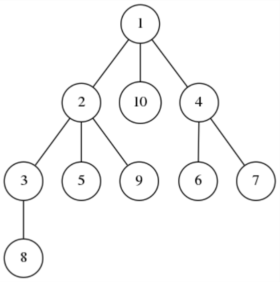
- In biology and phylogenetics to represent the evolutionary relationship among species. [Fel03, Ste16]

Eukaryotes Tree of Life 2020, showing positions of fungi and fungus-like organisms. Tricholome (wikipedia)

Tree structures

Tree structures also appear in a many other domains:

- In biology and phylogenetics to represent the evolutionary relationship among species. [Fel03, Ste16]
- Simple models for epidemics. [Moo74]

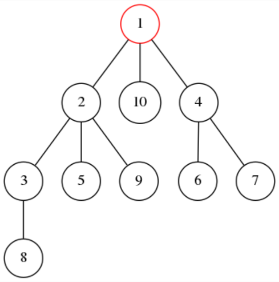


Infection spreading

Tree structures

Tree structures also appear in a many other domains:

- In biology and phylogenetics to represent the evolutionary relationship among species. [Fel03, Ste16]
- Simple models for epidemics. [Moo74]

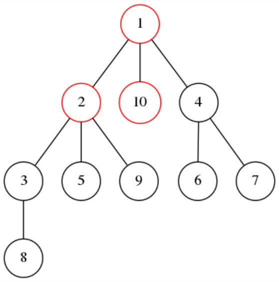


Infection spreading

Tree structures

Tree structures also appear in a many other domains:

- In biology and phylogenetics to represent the evolutionary relationship among species. [Fel03, Ste16]
- Simple models for epidemics. [Moo74]

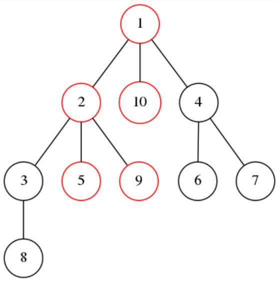


Infection spreading

Tree structures

Tree structures also appear in a many other domains:

- In biology and phylogenetics to represent the evolutionary relationship among species. [Fel03, Ste16]
- Simple models for epidemics. [Moo74]



Infection spreading

Tree structures

Tree structures also appear in a many other domains:

- In biology and phylogenetics to represent the evolutionary relationship among species. [Fel03, Ste16]
- Simple models for epidemics. [Moo74]
- In philology to construct the family tree (stemma) of preserved copies of ancient manuscript. [NH82]

TABLE 1

AW	0	0	0	4	4	4	10	20	38	52	84	124	203	205	225	482	711	726	926
MS	2	5	10	1	8	16	4	6	11	12	7	14	17	3	9	15	18	13	19

AW: accumulated weight of singular readings
MS: number of manuscript

From [NH82]

Tree structures

Tree structures also appear in a many other domains:

- In biology and phylogenetics to represent the evolutionary relationship among species. [Fel03, Ste16]
- Simple models for epidemics. [Moo74]
- In philology to construct the family tree (stemma) of preserved copies of ancient manuscript. [NH82]

TABLE 1

AW	0	0	0	4	4	4	10	20	38	52	84	124	203	205	225	482	711	726	926
MS	2	5	10	1	8	16	4	6	11	12	7	14	17	3	9	15	18	13	19

AW: accumulated weight of singular readings
MS: number of manuscript

TABLE 3

AW	0	0	0	4	4	4	10	20	38	52	84	124	203	205	225	482	711	726	926
MS	2	5	10	1	8	16	4	6	11	12	7	14	17	3	9	15	18	13	19
TN	N	N	N	N	N	N	N	N	T	TN	T	T	T	T	T	T	T	T	T

AW: accumulated weight of singular readings
MS: number of manuscripts
TN: indication whether terminal (T) or not (N)

From [NH82]

Increasing trees

- There are different varieties of trees (labelled, plane, rooted).
- Increasing trees have interesting properties.

Increasing trees

- There are different varieties of trees (labelled, plane, rooted).
- Increasing trees have interesting properties.

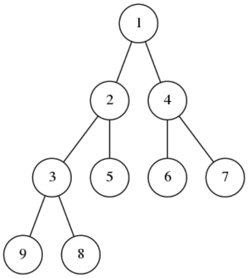
Increasing trees have labellings that are increasing along their branches. They are used in:

Increasing trees

- There are different varieties of trees (labelled, plane, rooted).
- Increasing trees have interesting properties.

Increasing trees have labellings that are increasing along their branches. They are used in:

- Analysis of permutations and data structures like binary search trees using increasing binary trees. [Drm09, Mah92, FGM+06]

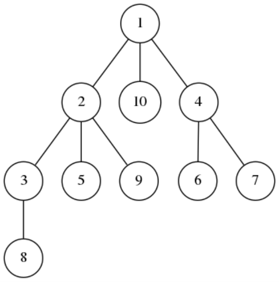


Increasing trees

- There are different varieties of trees (labelled, plane, rooted).
- Increasing trees have interesting properties.

Increasing trees have labellings that are increasing along their branches. They are used in:

- Analysis of permutations and data structures like binary search trees using increasing binary trees. [Drm09, Mah92, FGM$^+$06]
- Trees of epidemic spreading and manuscript reconstruction are also increasing trees.



Increasing trees

- There are different varieties of trees (labelled, plane, rooted).
- Increasing trees have interesting properties.

Increasing trees have labellings that are increasing along their branches. They are used in:

- Analysis of permutations and data structures like binary search trees using increasing binary trees. [Drm09, Mah92, FGM+06]
- Trees of epidemic spreading and manuscript reconstruction are also increasing trees.
- Study the number of executions of a parallel process and their synchronisations. This leads to repeated labellings. [BGP16, BGR17]

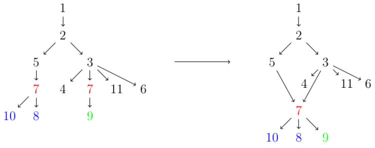


Figure 1. A synchronized tree

From [BGR17]

Different approaches

In order to study these trees several mathematical models have been introduced

Different approaches

In order to study these trees several mathematical models have been introduced

In probability theory:

Make a tree grow according to some probability distribution.

Examples

Different approaches

In order to study these trees several mathematical models have been introduced

In probability theory:

Make a tree grow according to some probability distribution.

Examples

- Galton-Watson trees to study extinction of family names.
- Yule trees to study speciation in phylogenetic trees.
- Binary search trees for the analysis of data structures.

Different approaches

In order to study these trees several mathematical models have been introduced

In probability theory:

Make a tree grow according to some probability distribution.

Examples

- Galton-Watson trees to study extinction of family names.
- Yule trees to study speciation in phylogenetic trees.
- Binary search trees for the analysis of data structures.

In combinatorics:

Describe all trees that belong to a certain class of trees and classify them according to their size. Then, count all trees of a fixed size.

Examples

Different approaches

In order to study these trees several mathematical models have been introduced

In probability theory:

Make a tree grow according to some probability distribution.

Examples

- Galton-Watson trees to study extinction of family names.
- Yule trees to study speciation in phylogenetic trees.
- Binary search trees for the analysis of data structures.

In combinatorics:

Describe all trees that belong to a certain class of trees and classify them according to their size. Then, count all trees of a fixed size.

Examples

- Simply generated trees for arithmetic expressions.
- Classical increasing trees for the analysis of data structures.

Different approaches

In order to study these trees several mathematical models have been introduced

In probability theory:

Make a tree grow according to some probability distribution.

Examples

- Galton-Watson trees to study extinction of family names.
- Yule trees to study speciation in phylogenetic trees.
- Binary search trees for the analysis of data structures.

In combinatorics:

Describe all trees that belong to a certain class of trees and classify them according to their size. Then, count all trees of a fixed size.

Examples

- Simply generated trees for arithmetic expressions.
- Classical increasing trees for the analysis of data structures.

Both approaches are complementary. It is possible to study random trees and derive similar type of results.

Analytic combinatorics

Analytic combinatorics

Definition

A combinatorial class is a countable set of structures (with a notion of size defined on them) where the number of elements of any given size is finite.

Framework to analyse properties of large random structures belonging to specifiable combinatorial classes

Definition

A combinatorial class is a countable set of structures (with a notion of size defined on them) where the number of elements of any given size is finite.

Framework to analyse properties of large random structures belonging to specifiable combinatorial classes

- **Symbolic method** developed in [FS09] is a grammar used to define (specify) combinatorial classes:
  1. Elementary constructions are the neutral class and the atomic class.
  2. Basic operators such that the disjoint union, Cartesian product and sequences.
  3. Combining (1) and (2) $\Longrightarrow$ create complex classes.

Analytic combinatorics

Definition

A combinatorial class is a countable set of structures (with a notion of size defined on them) where the number of elements of any given size is finite.

Framework to analyse properties of large random structures belonging to specifiable combinatorial classes

- **Symbolic method** developed in [FS09] is a grammar used to define (specify) combinatorial classes:
  1. Elementary constructions are the neutral class and the atomic class.
  2. Basic operators such that the disjoint union, Cartesian product and sequences.
  3. Combining (1) and (2) $\implies$ create complex classes.

Example: Words over the alphabet $\{\bullet, \bullet, \bullet\}$ where the size is the number of letters.

$\bullet\bullet\bullet\bullet \implies$ size 4

$\bullet\bullet\bullet\bullet\bullet\bullet \implies$ size 6.

These words can be specified using atomic classes, disjoint union and sequence.

Analytic combinatorics

Definition

A combinatorial class is a countable set of structures (with a notion of size defined on them) where the number of elements of any given size is finite.

Framework to analyse properties of large random structures belonging to specifiable combinatorial classes

- **Symbolic method** developed in [FS09] is a grammar used to define (specify) combinatorial classes:
    1. Elementary constructions are the neutral class and the atomic class.
    2. Basic operators such that the disjoint union, Cartesian product and sequences.
    3. Combining (1) and (2) $\implies$ create complex classes.
- **Generating functions** are functions with a formal variable that encompass information about the number of objects of each size of the combinatorial class.

Example: Words over the alphabet $\{\bullet, \bullet, \bullet\}$ where the size is the number of letters.

$\bullet\bullet\bullet\bullet \implies$ size 4

$\bullet\bullet\bullet\bullet\bullet\bullet \implies$ size 6.

These words can be specified using atomic classes, disjoint union and sequence.

Analytic combinatorics

Definition

A combinatorial class is a countable set of structures (with a notion of size defined on them) where the number of elements of any given size is finite.

Framework to analyse properties of large random structures belonging to specifiable combinatorial classes

- **Symbolic method** developed in [FS09] is a grammar used to define (specify) combinatorial classes:
  1. Elementary constructions are the neutral class and the atomic class.
  2. Basic operators such that the disjoint union, Cartesian product and sequences.
  3. Combining (1) and (2) $\Longrightarrow$ create complex classes.
- **Generating functions** are functions with a formal variable that encompass information about the number of objects of each size of the combinatorial class.

Example: Words over the alphabet $\{\bullet, \bullet, \bullet\}$ where the size is the number of letters.

$\bullet\bullet\bullet\bullet \Longrightarrow$ size 4

$\bullet\bullet\bullet\bullet\bullet\bullet \Longrightarrow$ size 6.

These words can be specified using atomic classes, disjoint union and sequence.

Result

Operations in the symbolic method translates directly to operations on generating functions.

- Theorems for automatic asymptotic estimates.
- Theorems for the shape of large random structures.

Ordinary generating functions

For a combinatorial class $\mathcal{C}$ we define its *ordinary generating function* (OGF) to be $C(z) = \sum_{n=0}^{\infty} C_n z^n$ where C_n counts the number of objects in $\mathcal{C}$ of size n.

Ordinary generating functions

For a combinatorial class $\mathcal{C}$ we define its *ordinary generating function* (OGF) to be $C(z) = \sum_{n=0}^{\infty} C_n z^n$ where C_n counts the number of objects in $\mathcal{C}$ of size n.

Symbolic method of ordinary generating functions [FS09]

Operation	Notation	Description	OGF
Neutral class	ϵ	Class consisting of single object of size 0	1
Atomic class	$\mathcal{Z}$	Class consisting of single object of size 1	z
Disjoint Union	$\mathcal{F} + \mathcal{G}$	Disjoint of objects from $\mathcal{F}$ and $\mathcal{G}$	$F(z) + G(z)$
Cartesian product	$\mathcal{F} \times \mathcal{G}$	Ordered pairs of objects one from $\mathcal{F}$ and one from $\mathcal{G}$	$F(z) \cdot G(z)$
Sequence	$\textsc{Seq}\,\mathcal{F}$	Sequences of objects from $\mathcal{F}$	$\frac{1}{1 - F(z)}$
Substitution	$\mathcal{F} \circ \mathcal{G}$	Substitute elements of $\mathcal{G}$ for atoms of $\mathcal{F}$	$F(G(z))$
Erasing i atoms	$\mathcal{E}^i \mathcal{F}$	Erase i atoms from objects of $\mathcal{F}$	$\frac{F^{(i)}(z)}{i!}$
...			

Ordinary generating functions

For a combinatorial class $\mathcal{C}$ we define its *ordinary generating function* (OGF) to be $C(z) = \sum_{n=0}^{\infty} C_n z^n$ where C_n counts the number of objects in $\mathcal{C}$ of size n.

Symbolic method of ordinary generating functions [FS09]

Operation	Notation	Description	OGF
Neutral class	ϵ	Class consisting of single object of size 0	1
Atomic class	$\mathcal{Z}$	Class consisting of single object of size 1	z
Disjoint Union	$\mathcal{F} + \mathcal{G}$	Disjoint of objects from $\mathcal{F}$ and $\mathcal{G}$	$F(z) + G(z)$
Cartesian product	$\mathcal{F} \times \mathcal{G}$	Ordered pairs of objects one from $\mathcal{F}$ and one from $\mathcal{G}$	$F(z) \cdot G(z)$
Sequence	$\text{SEQ}\,\mathcal{F}$	Sequences of objects from $\mathcal{F}$	$\dfrac{1}{1 - F(z)}$
Substitution	$\mathcal{F} \circ \mathcal{G}$	Substitute elements of $\mathcal{G}$ for atoms of $\mathcal{F}$	$F(G(z))$
Erasing i atoms	$\mathcal{E}^i \mathcal{F}$	Erase i atoms from objects of $\mathcal{F}$	$\dfrac{F^{(i)}(z)}{i!}$
...			

Example: Words over the 3 letter alphabet $\{{\color{blue}\bullet}, {\color{green}\bullet}, {\color{red}\bullet}\}$ For example $w = {\color{red}\bullet}{\color{green}\bullet}{\color{red}\bullet}{\color{blue}\bullet}{\color{green}\bullet}{\color{blue}\bullet}{\color{blue}\bullet}{\color{green}\bullet}{\color{blue}\bullet}{\color{green}\bullet}{\color{red}\bullet}{\color{blue}\bullet} \in \mathcal{W}$

$$\mathcal{W} = \text{SEQ}\,(\mathcal{Z}_{\color{blue}\bullet} + \mathcal{Z}_{\color{green}\bullet} + \mathcal{Z}_{\color{red}\bullet}) \overset{\text{Symbolic method}}{\Longrightarrow} W(z) = \frac{1}{1 - 3z}$$

The terms are then obtained by coefficient extraction $[z^n]W(z)$
$W(z) = 1 + 3z + 9z^2 + 27z^3 + 81z^4 + 243z^5 + 729z^6 + 2187z^7 + 6561z^8 + 19683z^9 + 59049z^{10} + \ldots$

Plane simple trees varieties

Plane simple trees are rooted unlabelled trees

Plane simple trees varieties

Plane simple trees are rooted unlabelled trees

Definition (Weighted degree function)

For a class of trees with ϕ_i colours for i-ary nodes, we define its degree function to be $\phi(z) = \sum_{i\geq 0} \phi_i z^i$.

For example

$$\phi(z) = 1 + z^2 + 10z^3 + 2z^5.$$

Corresponds to a class of trees having:

- One type of leaves.
- Binary nodes of 1 colour.
- Ternary nodes of 10 colours.
- 5-ary nodes of 2 colours.

Plane simple trees varieties

Plane simple trees are rooted unlabelled trees

Definition (Weighted degree function)

For a class of trees with ϕ_i colours for i-ary nodes, we define its degree function to be $\phi(z) = \sum_{i \geq 0} \phi_i z^i$.

For example

$$\phi(z) = 1 + z^2 + 10z^3 + 2z^5.$$

Corresponds to a class of trees having:

- One type of leaves.
- Binary nodes of 1 colour.
- Ternary nodes of 10 colours.
- 5-ary nodes of 2 colours.

Given a weighted degree function $\phi(z)$ such that $\phi_0 > 0$, the variety of plane simple trees parameterised by ϕ is specified in world of OGF by,

$$\mathcal{T} = \mathcal{Z} \times (\phi \circ \mathcal{T})$$

which gives,

$$T(z) = z\,\phi(T(z))$$

Plane simple trees varieties

Plane simple trees are rooted unlabelled trees

Definition (Weighted degree function)

For a class of trees with ϕ_i colours for i-ary nodes, we define its degree function to be $\phi(z) = \sum_{i \geq 0} \phi_i z^i$.

For example

$$\phi(z) = 1 + z^2 + 10z^3 + 2z^5.$$

Corresponds to a class of trees having:

- One type of leaves.
- Binary nodes of 1 colour.
- Ternary nodes of 10 colours.
- 5-ary nodes of 2 colours.

Given a weighted degree function $\phi(z)$ such that $\phi_0 > 0$, the variety of plane simple trees parameterised by ϕ is specified in world of OGF by,

$$\mathcal{T} = \mathcal{Z} \times (\phi \circ \mathcal{T})$$

which gives,

$$T(z) = z\,\phi(T(z))$$

Example:

Binary trees are parameterised by

$\phi(z) = 1 + 2z + z^2$, then,

$B(z) = z\phi(B(z))$

Solves to,

$B(z) = -1 + \frac{(1-\sqrt{1-4\,z})}{2\,z}$

Plane simple trees varieties

Plane simple trees are rooted unlabelled trees

Definition (Weighted degree function)

For a class of trees with ϕ_i colours for i-ary nodes, we define its degree function to be $\phi(z) = \sum_{i \geq 0} \phi_i z^i$.

For example

$$\phi(z) = 1 + z^2 + 10z^3 + 2z^5.$$

Corresponds to a class of trees having:

- One type of leaves.
- Binary nodes of 1 colour.
- Ternary nodes of 10 colours.
- 5-ary nodes of 2 colours.

Given a weighted degree function $\phi(z)$ such that $\phi_0 > 0$, the variety of plane simple trees parameterised by ϕ is specified in world of OGF by,

$$\mathcal{T} = \mathcal{Z} \times (\phi \circ \mathcal{T})$$

which gives,

$$T(z) = z\,\phi(T(z))$$

Example:

Binary trees are parameterised by

$\phi(z) = 1 + 2z + z^2$, then,

$B(z) = z\phi(B(z))$

Solves to,

$B(z) = -1 + \frac{(1 - \sqrt{1 - 4\,z})}{2\,z}$

In plane simple trees nodes can be decorated but do not bear labels

Exponential generating functions

The *exponential generating function* (EGF) to be $C(z) = \sum_{n=0}^{\infty} C_n \frac{z^n}{n!}$ where C_n counts the number of objects in $\mathcal{C}$ of size n.

Exponential generating functions

The *exponential generating function* (EGF) to be $C(z) = \sum_{n=0}^{\infty} C_n \frac{z^n}{n!}$ where C_n counts the number of objects in $\mathcal{C}$ of size n.

Symbolic method of exponential generating functions [FS09]

Operation	Notation	Description	EGF
Neutral class	ϵ	Class consisting of single object of size 0	1
Atomic class	$\mathcal{Z}$	Class consisting of single object of size 1	z
Disjoint Union	$\mathcal{F} + \mathcal{G}$	Disjoint of objects from $\mathcal{F}$ and $\mathcal{G}$	$F(z) + G(z)$
Labelled product	$\mathcal{F} \star \mathcal{G}$	well-labelled ordered pairs of objects one from $\mathcal{F}$ and one from $\mathcal{G}$	$F(z) \cdot G(z)$
Sequence	$\textsc{Seq}\,\mathcal{F}$	Sequences of objects from $\mathcal{F}$	$\frac{1}{1 - F(z)}$
Set	$\textsc{Set}\,\mathcal{F}$	Set of objects from $\mathcal{F}$	$\exp(F(z))$
...			

Exponential generating functions

The *exponential generating function* (EGF) to be $C(z) = \sum_{n=0}^{\infty} C_n \frac{z^n}{n!}$ where C_n counts the number of objects in $\mathcal{C}$ of size n.

Symbolic method of exponential generating functions [FS09]

Operation	Notation	Description	EGF
Neutral class	ϵ	Class consisting of single object of size 0	1
Atomic class	$\mathcal{Z}$	Class consisting of single object of size 1	z
Disjoint Union	$\mathcal{F} + \mathcal{G}$	Disjoint of objects from $\mathcal{F}$ and $\mathcal{G}$	$F(z) + G(z)$
Labelled product	$\mathcal{F} \star \mathcal{G}$	well-labelled ordered pairs of objects one from $\mathcal{F}$ and one from $\mathcal{G}$	$F(z) \cdot G(z)$
Sequence	$\textsc{Seq}\,\mathcal{F}$	Sequences of objects from $\mathcal{F}$	$\frac{1}{1 - F(z)}$
Set	$\textsc{Set}\,\mathcal{F}$	Set of objects from $\mathcal{F}$	$\exp(F(z))$
...			

Example: Ordered set partitions. $[\{2, 4, 5\}, \{1, 7\}, \{3, 6\}]$ is an ordered partition of size 7.

$$\mathcal{B} = \textsc{Seq}\left(\textsc{Set}_{\geq 1}(\mathcal{Z})\right) \overset{\text{symbolic method}}{\Longrightarrow} B(z) = \frac{1}{1 - (\exp(z) - 1)} = \frac{1}{2 - \exp(z)}$$

The terms are then obtained by $n![z^n]B(z)$ and are called Ordered Bell numbers:
$B(z) = 1z + 3z^2 + 13z^3 + 75z^4 + 541z^5 + 4683z^6 + 47293z^7 + 545835z^8 + \ldots$

Greene operator and increasing trees

- Labelled structures are naturally specified with EGF since each atom bears an integer label. Then the normalisation $\frac{z^n}{n!}$ insures the generating function to be convergent and analytic methods apply.
- The term increasing trees classically refers to trees such that the labels are strictly increasing along branches and have no label repetitions.[BFS92]

Greene operator and increasing trees

- Labelled structures are naturally specified with EGF since each atom bears an integer label. Then the normalisation $\frac{z^n}{n!}$ insures the generating function to be convergent and analytic methods apply.
- The term increasing trees classically refers to trees such that the labels are strictly increasing along branches and have no label repetitions.[BFS92]

The boxed product (Greene operator) is defined in the EGF world. That is defined as the label product with the additional constraint that the smallest left has to appear on the left class $\mathcal{B}$.

$$\mathcal{A} = \mathcal{B}^{\square} \star \mathcal{C} \to A(z) = \int_0^z (\partial_t B(t)) \cdot C(t) dt$$

Greene operator and increasing trees

- Labelled structures are naturally specified with EGF since each atom bears an integer label. Then the normalisation $\frac{z^n}{n!}$ insures the generating function to be convergent and analytic methods apply.
- The term increasing trees classically refers to trees such that the labels are strictly increasing along branches and have no label repetitions.[BFS92]

The boxed product (Greene operator) is defined in the EGF world. That is defined as the label product with the additional constraint that the smallest left has to appear on the left class $\mathcal{B}$.

$$\mathcal{A} = \mathcal{B}^{\square} \star \mathcal{C} \rightarrow A(z) = \int_0^z (\partial_t B(t)) \cdot C(t) dt$$

Example: Increasing binary trees.

$$\mathcal{B} = \epsilon + \mathcal{Z}^{\square} \star (\mathcal{B} \star \mathcal{B}) \stackrel{\text{symbolic method}}{\Longrightarrow} B(z) = 1 + \int_0^z 1 \cdot B^2(t)\, dt$$

Greene operator and increasing trees

- Labelled structures are naturally specified with EGF since each atom bears an integer label. Then the normalisation $\frac{z^n}{n!}$ insures the generating function to be convergent and analytic methods apply.
- The term increasing trees classically refers to trees such that the labels are strictly increasing along branches and have no label repetitions.[BFS92]

The boxed product (Greene operator) is defined in the EGF world. That is defined as the label product with the additional constraint that the smallest left has to appear on the left class $\mathcal{B}$.

$$\mathcal{A} = \mathcal{B}^{\square} \star \mathcal{C} \rightarrow A(z) = \int_0^z (\partial_t B(t)) \cdot C(t) dt$$

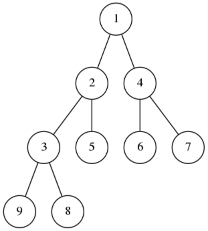
Example: Increasing binary trees.

$$\mathcal{B} = \epsilon + \mathcal{Z}^{\square} \star (\mathcal{B} \star \mathcal{B}) \overset{\text{symbolic method}}{\Longrightarrow} B(z) = 1 + \int_0^z 1 \cdot B^2(t)\, dt$$

which solves to

$$B'(z) = B^2(z), B(0) = 1 \implies B(z) = \frac{1}{1-z}$$

$$B_n = n![z^n]B(z) = n!.$$

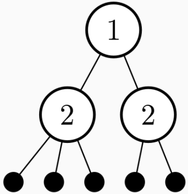


An evolution process

- Differenciations appear over periods of times and can appear simultaneously in different individuals.
- We are interested in the number of living individuals.
- Differenciations are not necessarily binary.

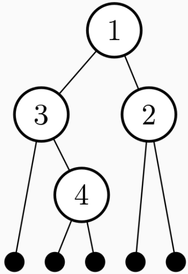
An evolution process

- Differenciations appear over periods of times and can appear simultaneously in different individuals.
- We are interested in the number of living individuals.
- Differenciations are not necessarily binary.



An evolution process

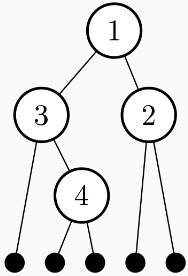
- Differenciations appear over periods of times and can appear simultaneously in different individuals.
- We are interested in the number of living individuals.
- Differenciations are not necessarily binary.



An evolution process

- Differenciations appear over periods of times and can appear simultaneously in different individuals.
- We are interested in the number of living individuals.
- Differenciations are not necessarily binary.

Can be modeled using trees such that

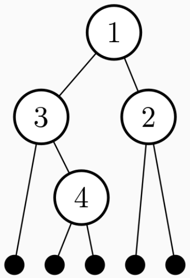


An evolution process

- Differenciations appear over periods of times and can appear simultaneously in different individuals.
- We are interested in the number of living individuals.
- Differenciations are not necessarily binary.

Can be modeled using trees such that

- Internal nodes bear integer labels corresponding to the time of differentiation (label repetitions are allowed).
- The size of the tree is its number of leaves.
- Nodes can have different arities.
- Branches are strictly increasing (label repetitions allowed).



Parametrisable evolution process for classes of strict monotonic Schröder trees

Parameters of the evolution process

Definition (Coloured degree function)

For a class of trees with ϕ_i colours of i-ary nodes, we define its coloured degree function to be $\phi(z) = \sum_{i \geq 1} \phi_i z^i$.

Definition (Set of allowed repetitions)

The set $r \subset \mathbb{N}^*$.

Parameters of the evolution process

Definition (Coloured degree function)

For a class of trees with ϕ_i colours of i-ary nodes, we define its coloured degree function to be $\phi(z) = \sum_{i \geq 1} \phi_i z^i$.

For example

$$\phi(z) = z^2 + 10z^3 + 2z^5.$$

Corresponds to a class of trees having:

- Binary nodes of 1 colour.
- Ternary nodes of 10 colours.
- 5-ary nodes of 2 colours.

A coloured degree function is a weighted degree function where $\phi_0 = 0$.

Definition (Set of allowed repetitions)

The set $r \subset \mathbb{N}^*$.

Parameters of the evolution process

Definition (Coloured degree function)

For a class of trees with ϕ_i colours of i-ary nodes, we define its coloured degree function to be $\phi(z) = \sum_{i \geq 1} \phi_i z^i$.

For example

$$\phi(z) = z^2 + 10z^3 + 2z^5.$$

Corresponds to a class of trees having:

- Binary nodes of 1 colour.
- Ternary nodes of 10 colours.
- 5-ary nodes of 2 colours.

A coloured degree function is a weighted degree function where $\phi_0 = 0$.

Definition (Set of allowed repetitions)

The set $r \subset \mathbb{N}^*$.

For example

$$r = \{2, 3, 5\}.$$

At each iteration step there are either 2, 3 or 5 repetitions of the same label (i.e the number of leaves that evolves at each step is constrained to lie in r).

Evolution process for a variety of strict monotonic trees

The evolution process has two parameters a coloured degree function and a set of allowed repetitions.

$$\phi(z) = 2z^2 + 2z^3 \quad \text{and } r = \mathbb{N}^*$$

We have two colours of binary nodes (w,g) and two colours of ternary nodes (w,g).

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

1. Choose a non-empty subset L of leaves of the so-far built tree such that $|L| \in r$.
2. For each $\ell \in L$ choose an integer $k > 1$ such that $\phi_k > 0$, and one of the $1 \leq c \leq \phi_k$ colours.
3. Replace ℓ with an internal node labelled i with the chosen colour and attach to it k new leaves.

Evolution process for a variety of strict monotonic trees

The evolution process has two parameters a coloured degree function and a set of allowed repetitions.

$$\phi(z) = 2z^2 + 2z^3 \quad \text{and } r = \mathbb{N}^*$$

We have two colours of binary nodes (w,g) and two colours of ternary nodes (w,g).

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

1. Choose a non-empty subset L of leaves of the so-far built tree such that $|L| \in r$.
2. For each $\ell \in L$ choose an integer $k > 1$ such that $\phi_k > 0$, and one of the $1 \leq c \leq \phi_k$ colours.
3. Replace ℓ with an internal node labelled i with the chosen colour and attach to it k new leaves.

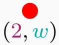
Evolution process for a variety of strict monotonic trees

The evolution process has two parameters a coloured degree function and a set of allowed repetitions.

$$\phi(z) = 2z^2 + 2z^3 \quad \text{and } r = \mathbb{N}^*$$

We have two colours of binary nodes (w,g) and two colours of ternary nodes (w,g).

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



1. Choose a non-empty subset L of leaves of the so-far built tree such that $|L| \in r$.
2. For each $\ell \in L$ choose an integer $k > 1$ such that $\phi_k > 0$, and one of the $1 \leq c \leq \phi_k$ colours.
3. Replace ℓ with an internal node labelled i with the chosen colour and attach to it k new leaves.

Evolution process for a variety of strict monotonic trees

The evolution process has two parameters a coloured degree function and a set of allowed repetitions.

$$\phi(z) = 2z^2 + 2z^3 \quad \text{and } r = \mathbb{N}^*$$

We have two colours of binary nodes (w,g) and two colours of ternary nodes (w,g).

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

1. Choose a non-empty subset L of leaves of the so-far built tree such that $|L| \in r$.
2. For each $\ell \in L$ choose an integer $k > 1$ such that $\phi_k > 0$, and one of the $1 \leq c \leq \phi_k$ colours.
3. Replace ℓ with an internal node labelled i with the chosen colour and attach to it k new leaves.

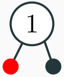
Evolution process for a variety of strict monotonic trees

The evolution process has two parameters a coloured degree function and a set of allowed repetitions.

$$\phi(z) = 2z^2 + 2z^3 \quad \text{and } r = \mathbb{N}^*$$

We have two colours of binary nodes (w,g) and two colours of ternary nodes (w,g).

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



1. Choose a non-empty subset L of leaves of the so-far built tree such that $|L| \in r$.
2. For each $\ell \in L$ choose an integer $k > 1$ such that $\phi_k > 0$, and one of the $1 \leq c \leq \phi_k$ colours.
3. Replace ℓ with an internal node labelled i with the chosen colour and attach to it k new leaves.

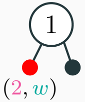
Evolution process for a variety of strict monotonic trees

The evolution process has two parameters a coloured degree function and a set of allowed repetitions.

$$\phi(z) = 2z^2 + 2z^3 \quad \text{and } r = \mathbb{N}^*$$

We have two colours of binary nodes (w,g) and two colours of ternary nodes (w,g).

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



1. Choose a non-empty subset L of leaves of the so-far built tree such that $|L| \in r$.
2. For each $\ell \in L$ choose an integer $k > 1$ such that $\phi_k > 0$, and one of the $1 \leq c \leq \phi_k$ colours.
3. Replace ℓ with an internal node labelled i with the chosen colour and attach to it k new leaves.

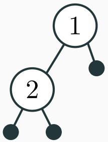
Evolution process for a variety of strict monotonic trees

The evolution process has two parameters a coloured degree function and a set of allowed repetitions.

$$\phi(z) = 2z^2 + 2z^3 \quad \text{and } r = \mathbb{N}^*$$

We have two colours of binary nodes (w,g) and two colours of ternary nodes (w,g).

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



1. Choose a non-empty subset L of leaves of the so-far built tree such that $|L| \in r$.
2. For each $\ell \in L$ choose an integer $k > 1$ such that $\phi_k > 0$, and one of the $1 \leq c \leq \phi_k$ colours.
3. Replace ℓ with an internal node labelled i with the chosen colour and attach to it k new leaves.

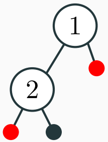
Evolution process for a variety of strict monotonic trees

The evolution process has two parameters a coloured degree function and a set of allowed repetitions.

$$\phi(z) = 2z^2 + 2z^3 \quad \text{and} \quad r = \mathbb{N}^*$$

We have two colours of binary nodes (w,g) and two colours of ternary nodes (w,g).

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



1. Choose a non-empty subset L of leaves of the so-far built tree such that $|L| \in r$.
2. For each $\ell \in L$ choose an integer $k > 1$ such that $\phi_k > 0$, and one of the $1 \leq c \leq \phi_k$ colours.
3. Replace ℓ with an internal node labelled i with the chosen colour and attach to it k new leaves.

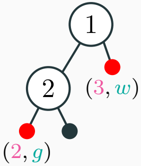
Evolution process for a variety of strict monotonic trees

The evolution process has two parameters a coloured degree function and a set of allowed repetitions.

$$\phi(z) = 2z^2 + 2z^3 \quad \text{and } r = \mathbb{N}^*$$

We have two colours of binary nodes (w,g) and two colours of ternary nodes (w,g).

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



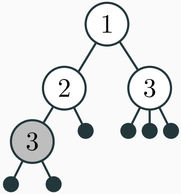
1. Choose a non-empty subset L of leaves of the so-far built tree such that $|L| \in r$.
2. For each $\ell \in L$ choose an integer $k > 1$ such that $\phi_k > 0$, and one of the $1 \leq c \leq \phi_k$ colours.
3. Replace ℓ with an internal node labelled i with the chosen colour and attach to it k new leaves.

The evolution process has two parameters a coloured degree function and a set of allowed repetitions.

$$\phi(z) = 2z^2 + 2z^3 \quad \text{and } r = \mathbb{N}^*$$

We have two colours of binary nodes (w,g) and two colours of ternary nodes (w,g).

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



1. Choose a non-empty subset L of leaves of the so-far built tree such that $|L| \in r$.
2. For each $\ell \in L$ choose an integer $k > 1$ such that $\phi_k > 0$, and one of the $1 \leq c \leq \phi_k$ colours.
3. Replace ℓ with an internal node labelled i with the chosen colour and attach to it k new leaves.

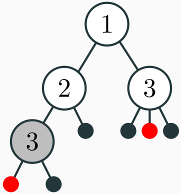
Evolution process for a variety of strict monotonic trees

The evolution process has two parameters a coloured degree function and a set of allowed repetitions.

$$\phi(z) = 2z^2 + 2z^3 \quad \text{and } r = \mathbb{N}^*$$

We have two colours of binary nodes (w,g) and two colours of ternary nodes (w,g).

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



1. Choose a non-empty subset L of leaves of the so-far built tree such that $|L| \in r$.
2. For each $\ell \in L$ choose an integer $k > 1$ such that $\phi_k > 0$, and one of the $1 \leq c \leq \phi_k$ colours.
3. Replace ℓ with an internal node labelled i with the chosen colour and attach to it k new leaves.

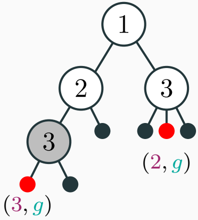
Evolution process for a variety of strict monotonic trees

The evolution process has two parameters a coloured degree function and a set of allowed repetitions.

$$\phi(z) = 2z^2 + 2z^3 \quad \text{and } r = \mathbb{N}^*$$

We have two colours of binary nodes (w,g) and two colours of ternary nodes (w,g).

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



1. Choose a non-empty subset L of leaves of the so-far built tree such that $|L| \in r$.
2. For each $\ell \in L$ choose an integer $k > 1$ such that $\phi_k > 0$, and one of the $1 \leq c \leq \phi_k$ colours.
3. Replace ℓ with an internal node labelled i with the chosen colour and attach to it k new leaves.

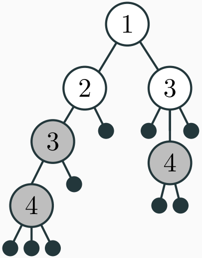
Evolution process for a variety of strict monotonic trees

The evolution process has two parameters a coloured degree function and a set of allowed repetitions.

$$\phi(z) = 2z^2 + 2z^3 \quad \text{and } r = \mathbb{N}^*$$

We have two colours of binary nodes (w,g) and two colours of ternary nodes (w,g).

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



1. Choose a non-empty subset L of leaves of the so-far built tree such that $|L| \in r$.
2. For each $\ell \in L$ choose an integer $k > 1$ such that $\phi_k > 0$, and one of the $1 \leq c \leq \phi_k$ colours.
3. Replace ℓ with an internal node labelled i with the chosen colour and attach to it k new leaves.

Specification for the evolution process

- Increasing trees are usually specified in the framework of EGF with the use of Greene operator.

Specification for the evolution process

- Increasing trees are usually specified in the framework of EGF with the use of Greene operator.
- But it is not well suited for increasing trees with labels repetitions.

Specification for the evolution process

- Increasing trees are usually specified in the framework of EGF with the use of Greene operator.
- But it is not well suited for increasing trees with labels repetitions.
- A specification with OGF is more natural.
- Problem : the specification is then only formal and classical analytic methods do not apply.

Specification for the evolution process

- Increasing trees are usually specified in the framework of EGF with the use of Greene operator.
- But it is not well suited for increasing trees with labels repetitions.
- A specification with OGF is more natural.
- Problem : the specification is then only formal and classical analytic methods do not apply.

Let $\phi(z)$ a coloured degree function, $r \subset \mathbb{N}^*$ a set of allowed repetitions and $m = min(r)$, then

$$\mathcal{B} = \mathcal{Z}^m + \sum_{i \in r} \left(\mathcal{E}^i\, \mathcal{B}\right) \times \left(\phi^i \backslash (\phi_1 \mathcal{Z})^i\right).$$

Specification for the evolution process

- Increasing trees are usually specified in the framework of EGF with the use of Greene operator.
- But it is not well suited for increasing trees with labels repetitions.
- A specification with OGF is more natural.
- Problem : the specification is then only formal and classical analytic methods do not apply.

Let $\phi(z)$ a coloured degree function, $r \subset \mathbb{N}^*$ a set of allowed repetitions and $m = min(r)$, then

$$\mathcal{B} = \mathcal{Z}^m + \sum_{i \in r} \left(\mathcal{E}^i\, \mathcal{B}\right) \times \left(\phi^i \backslash (\phi_1 \mathcal{Z})^i\right).$$

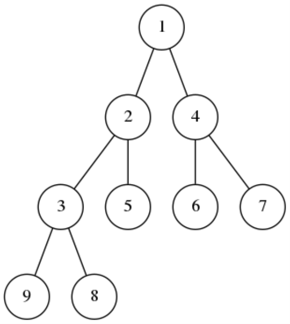
Which translates to,

$$B(z) = z^m + \sum_{i \in r} \frac{1}{i!} B^{(i)}(z) \left(\phi(z)^i - (\phi_1\, z)^i\right).$$

$\frac{B^{(i)}(z)}{i!}$ corresponds to erasing i leaves.

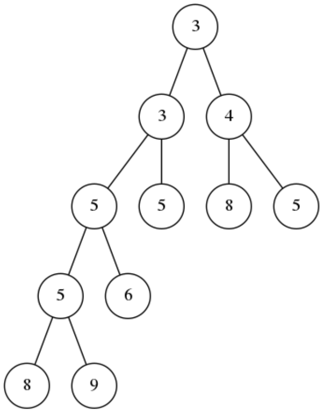
Related works on increasing trees

- Increasing trees [BFS92].
  No label repetitions and labellings along branches are strictly increasing.



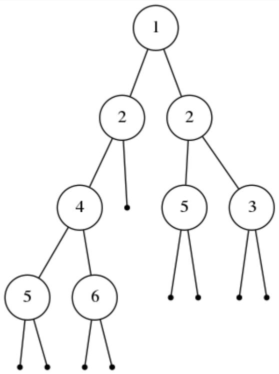
Related works on increasing trees

- Increasing trees [BFS92].
  No label repetitions and labellings along branches are strictly increasing.
- Monotone functions on trees [PU83].
  The maximum label is fixed and does not depend on the size of the tree. Labellings along branches are weakly increasing and some labels may be skipped.



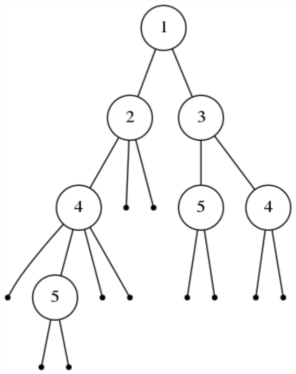
Related works on increasing trees

- Increasing trees [BFS92].
  No label repetitions and labellings along branches are strictly increasing.
- Monotone functions on trees [PU83].
  The maximum label is fixed and does not depend on the size of the tree. Labellings along branches are weakly increasing and some labels may be skipped.
- Strictly monotonic binary [BGGW20].
  Specific case of strictly monotonic trees with arity 2.



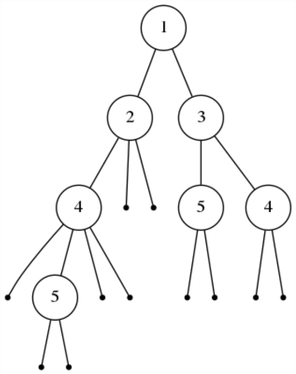
Related works on increasing trees

- Increasing trees [BFS92].
  No label repetitions and labellings along branches are strictly increasing.
- Monotone functions on trees [PU83].
  The maximum label is fixed and does not depend on the size of the tree. Labellings along branches are weakly increasing and some labels may be skipped.
- Strictly monotonic binary [BGGW20].
  Specific case of strictly monotonic trees with arity 2.
- Ranked Schröder trees [BGN19].
  Two increasing labellings where all arities are allowed.



Related works on increasing trees

- Increasing trees [BFS92].
  No label repetitions and labellings along branches are strictly increasing.
- Monotone functions on trees [PU83].
  The maximum label is fixed and does not depend on the size of the tree. Labellings along branches are weakly increasing and some labels may be skipped.
- Strictly monotonic binary [BGGW20].
  Specific case of strictly monotonic trees with arity 2.
- Ranked Schröder trees [BGN19].
  Two increasing labellings where all arities are allowed.
- Families of monotonic trees [BGNS20].
  A general asymptotic for cases where the number of repetitions allowed is not bounded.



Tree classes built with this evolution process

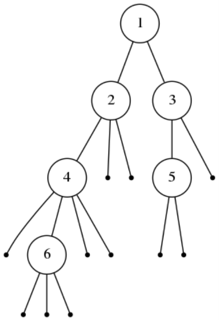
r	$\phi(z)$	Name	References
$\{1\}$	z^d	Plane d-ary increasing	[BFS92]
$\{1\}$	$\frac{z^2}{1-z}$	Increasing Schröder	[BGN19]
$\mathbb{N}^*$	z^2	Strict monotonic binary	[BGGW20]
$\mathbb{N}^*$	$\frac{z^2}{1-z}$	Strict monotonic Schröder	[BGN19]
$\mathbb{N}^*$	$\frac{z}{1-z}$	Strict monotonic general Schröder	[BGMN20]
$\mathbb{N}^*$	plane d-ary	Monotonic d-ary trees	[BGNS20]
$\{1, 2\}$	z^2	Supertrees	[SDH$^+$04]
$\{d\}$	z^2	Increasing binary with d label repetitions	

Table 1: Some of examples of tree classes covered by the evolution process

Three particular cases of the evolution process

Increasing Schröder trees

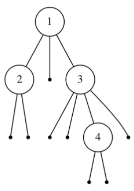
Corresponds to the parameters $\phi(z) = \frac{z^2}{1-z}$ and $r = \{1\}$. Therefore, a single leaf can evolve at each iteration step.



Increasing Schröder trees

Corresponds to the parameters $\phi(z) = \frac{z^2}{1-z}$ and $r = \{1\}$. Therefore, a single leaf can evolve at each iteration step.

- Number of trees is $\frac{n!}{2}$.
- Bijections with half permutations that preserves several parameters. Number of internal nodes and the depth of the leftmost leaf are related to the the number of cycles in a permutation.



This tree with 8 leaves Corresponds to the permutation $(1, 4, 5)(2)(3)(6, 8)(7)$ it has $8 + 1 - 5 = 4$ internal nodes.

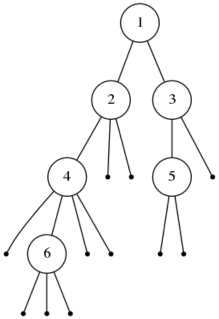
Increasing Schröder trees

Corresponds to the parameters $\phi(z) = \frac{z^2}{1-z}$ and $r = \{1\}$. Therefore, a single leaf can evolve at each iteration step.

- Number of trees is $\frac{n!}{2}$.
- Bijections with half permutations that preserves several parameters. Number of internal nodes and the depth of the leftmost leaf are related to the the number of cycles in a permutation.
- Typical parameters are:

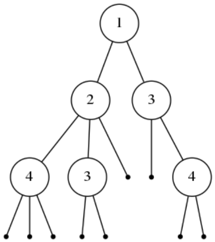
	Mean	Variance	Limit law
Internal nodes	$n - \ln n$	$\ln n$	Normal
Depth of the leftmost leaf	$\ln n$	$\ln n$	Normal
Height of the tree	$\Theta(\ln n)$		
Degree of the root	$2\,\mathrm{e} - 3$	$14\,\mathrm{e} - 4\,\mathrm{e}^2 - 8$	Modified Poisson

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



Strict monotonic Schröder trees

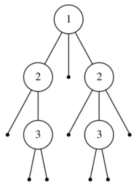
Corresponds to the parameters $\phi(z) = \frac{z^2}{1-z}$ and $r = \mathbb{N}^*$. Any number of repetitions is allowed and the node degrees can be anything ≥ 2.



Strict monotonic Schröder trees

Corresponds to the parameters $\phi(z) = \frac{z^2}{1-z}$ and $r = \mathbb{N}^*$. Any number of repetitions is allowed and the node degrees can be anything ≥ 2.

- Bijection with ordered set partitions (Ordered Bell numbers) with the number of iteration steps corresponding to the number of subsets in the partitions.
- $g_n = B_{n-1} \underset{n\to\infty}{=} \frac{(n-1)!}{2(\ln 2)^n}$, ($B_n$ is the n-th ordered Bell number).



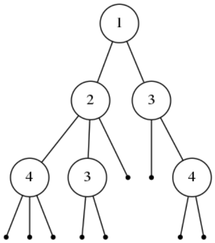
This tree with 8 leaves Corresponds to the ordered set partition $(\{3,4\},\{1,5,7\},\{2,6\})$. The tree has 3 distinct labels and the partition 3 subsets.

Strict monotonic Schröder trees

Corresponds to the parameters $\phi(z) = \frac{z^2}{1-z}$ and $r = \mathbb{N}^*$. Any number of repetitions is allowed and the node degrees can be anything ≥ 2.

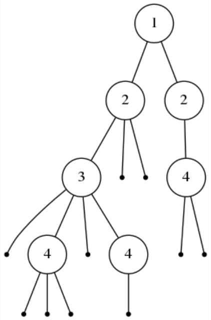
- Bijection with ordered set partitions (Ordered Bell numbers) with the number of iteration steps corresponding to the number of subsets in the partitions.
- $g_n = B_{n-1} \underset{n\to\infty}{=} \frac{(n-1)!}{2(\ln 2)^n}$, ($B_n$ is the n-th ordered Bell number).
- Typical parameters are:

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



Strict monotonic general Schröder trees

Corresponds to the parameters $\phi(z) = \frac{z}{1-z}$ and $r = \mathbb{N}^*$. Any number of repetitions is allowed and the node degrees can be anything ≥ 1 **including unary nodes**.

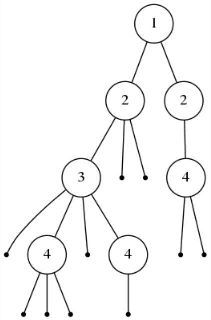


Strict monotonic general Schröder trees

Corresponds to the parameters $\phi(z) = \frac{z}{1-z}$ and $r = \mathbb{N}^*$. Any number of repetitions is allowed and the node degrees can be anything ≥ 1 **including unary nodes**.

- $f_n \underset{n\to\infty}{=} c\, 2^{\frac{(n-1)(n-2)}{2}}\, (n-1)!$.
- Typical parameters are:

	Mean
Internal nodes	$\Theta(n^2)$
Distinct labels	$\Theta(n)$
Unary nodes	$\Theta(n^2)$
Depth of the leftmost leaf	$\Theta(n)$
Height	$\Theta(n)$

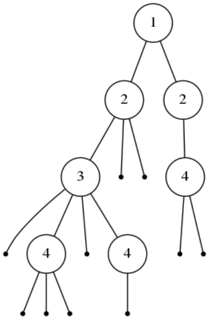


Strict monotonic general Schröder trees

Corresponds to the parameters $\phi(z) = \frac{z}{1-z}$ and $r = \mathbb{N}^*$. Any number of repetitions is allowed and the node degrees can be anything ≥ 1 **including unary nodes**.

- $f_n \underset{n\to\infty}{=} c\, 2^{\frac{(n-1)(n-2)}{2}} (n-1)!$.
- Typical parameters are:

	Mean
Internal nodes	$\Theta(n^2)$
Distinct labels	$\Theta(n)$
Unary nodes	$\Theta(n^2)$
Depth of the leftmost leaf	$\Theta(n)$
Height	$\Theta(n)$



Remark

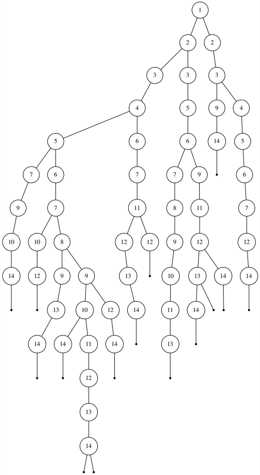
Unary nodes change significantly the number of trees and its typical shape.

Summary on the three models

	Number of trees	Number of internal nodes
Increasing Schröder trees	$n!/2$	$n - \ln n$
Strict monotonic Schröder trees	$(n-1)!/(2\,(\ln 2)^n)$	$n - \ln 2\ \ln n$
Strict monotonic general trees	$c\ 2^{(n-1)(n-2)/2}\ (n-1)!$	$\Theta(n^2)$

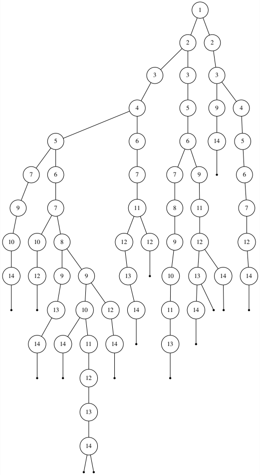
Remark

For all three models, it seems that typical large trees have nodes that are mostly from the first allowed arity, and little from the second allowed arity while other arities appear very rarely.



Summary on the three models

	Number of trees	Number of internal nodes
Increasing Schröder trees	$n!/2$	$n - \ln n$
Strict monotonic Schröder trees	$(n-1)!/(2\,(\ln 2)^n)$	$n - \ln 2\ \ln n$
Strict monotonic general trees	$c\ 2^{(n-1)(n-2)/2}\ (n-1)!$	$\Theta(n^2)$



Remark

For all three models, it seems that typical large trees have nodes that are mostly from the first allowed arity, and little from the second allowed arity while other arities appear very rarely.

This idea is developed in the next theorems in the form of asymptotic enumeration.

Main theorem I

Condition

Let $r \subset \mathbb{N}^*$ and $m = min(r)$. Let $\phi(z)$ be a *coloured degree function* and such that $\phi_1 = 0$, $\phi_2 \geq 1$ and $\phi_n \underset{n \to \infty}{=} O\left(\frac{n!}{m!^{n/m}\ n^{m+4}}\right)$.

Let $B_n^{\phi,r}$ be the number of trees of size n built via the evolution process

Main theorem I

Condition

Let $r \subset \mathbb{N}^*$ and $m = min(r)$. Let $\phi(z)$ be a *coloured degree function* and such that $\phi_1 = 0$, $\phi_2 \geq 1$ and $\phi_n \underset{n\to\infty}{=} O\left(\frac{n!}{m!^{n/m}\ n^{m+4}}\right)$.

Let $B_n^{\phi,r}$ be the number of trees of size n built via the evolution process

Theorem

Let $\phi(z)$ be a coloured degree function as in the condition, and $r \subset \mathbb{N}^*$, with $r \neq \emptyset$. Let $m = min(r)$, then as n tends to infinity and is of the form $n \equiv 0 \mod m$,

$$B_n^{r,\phi} \underset{n\to\infty}{\sim} \kappa\ n!\ \left(\frac{\phi_2}{\rho}\right)^n\ n^{-1+\frac{\rho\,\phi_3}{\phi_2^2}-\frac{\rho\, f''(\rho)}{f'(\rho)}},$$

where κ is a constant that depends on $\phi(z)$ and r. Let $f(z) = \sum_{i\in r} \frac{z^i}{i!}$, then ρ is the smallest positive real of the equation $f(z) - 1 = 0$.

Main theorem II

Condition

Let $\phi(z)$ be a *coloured degree function* and such that $\phi_1 \geq 1$, $\phi_2 \geq 1$ and $\phi_n = O\left(\frac{n!}{n^5}\right)$.

Let $B_n^{\phi,r}$ be the number of trees of size n built via the evolution process

Main theorem II

Condition

Let $\phi(z)$ be a *coloured degree function* and such that $\phi_1 \geq 1$, $\phi_2 \geq 1$ and $\phi_n = O\left(\frac{n!}{n^5}\right)$.

Let $B_n^{\phi,r}$ be the number of trees of size n built via the evolution process

Theorem

Let $\phi(z)$ be a coloured degree function as in the condition, let $r \subset \mathbb{N}^*$, $r \neq \emptyset$, and $r \neq \{1\}$, then as n tends to infinity,

$$B_n^{r,\phi} \underset{n\to\infty}{\sim} \kappa\,(n-1)!\,\phi_2^{n-1} \prod_{k=1}^{n-1} \left(\sum_{i=1, i\in r}^{n-k} \phi_1^{i-1} \binom{n-k-1}{i-1} \right),$$

where κ is a constant that depends on $\phi(z)$ and r.

The condition on ϕ_2 can be relaxed and a similar result holds.

Applications

Increasing binary trees with d repetitions

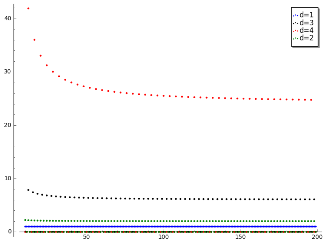
- $\mathcal{B}^d$ be the class of increasing binary trees with d label repetitions at each iteration step.
- At each iteration step exactly d leaves are chosen to expand (we start with a single root that has d leaves).

Increasing binary trees with d repetitions

- $\mathcal{B}^d$ be the class of increasing binary trees with d label repetitions at each iteration step.
- At each iteration step exactly d leaves are chosen to expand (we start with a single root that has d leaves).
- Specification with $\phi(z) = z^2$ and $r = \{d\}$.

d	Asymptotics	References
1	$(n-1)!$	EIS A000142
2	$c_2\, n!\, \left(2^{1/2}\right)^{-n} n^{-2}$	EIS A000680
3	$c_3\, n!\, \left(3!^{1/3}\right)^{-n} n^{-3}$	EIS A014606
4	$c_4\, n!\, \left(4!^{1/4}\right)^{-n} n^{-4}$	EIS A014608

Table 2: Asymptotic behaviour for B_n^d for $d \in \{1,2,3,4\}$ when $n \equiv 0 \mod d$. The sequences in OEIS appear shifted (without periodicities).



Simulation for $n \in \{10, 200\}$ of B_n^d divided by their expected asymptotic behaviour with $d \in \{1,2,3,4\}$.

Evolution process for monotonic trees

Example: Monotonic binary trees

Take $\phi(z)$ to be unlabelled binary trees (or Catalan trees)

$$\phi(z) = cat(z) - z = z^2 + 2z^3 + 5z^4 + \dots \quad \text{and } r = \mathbb{N}^*$$

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

1. Choose a non-empty subset L of leaves of the so-far built tree such that $|L| \in r$.
2. For each $\ell \in L$ choose an integer $k > 1$ such that $\phi_k > 0$, and one of the $1 \leq c \leq \phi_k$ possible unlabelled trees with k leaves.
3. Replace ℓ with the chosen unlabelled tree, labelling all its internal nodes i.

Example: **Monotonic binary trees**

Take $\phi(z)$ to be unlabelled binary trees (or Catalan trees)

$$\phi(z) = cat(z) - z = z^2 + 2z^3 + 5z^4 + \dots \quad \text{and } r = \mathbb{N}^*$$

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

1. Choose a non-empty subset L of leaves of the so-far built tree such that $|L| \in r$.
2. For each $\ell \in L$ choose an integer $k > 1$ such that $\phi_k > 0$, and one of the $1 \leq c \leq \phi_k$ possible unlabelled trees with k leaves.
3. Replace ℓ with the chosen unlabelled tree, labelling all its internal nodes i.

Evolution process for monotonic trees

Example: Monotonic binary trees

Take $\phi(z)$ to be unlabelled binary trees (or Catalan trees)

$$\phi(z) = cat(z) - z = z^2 + 2z^3 + 5z^4 + \dots \quad \text{and } r = \mathbb{N}^*$$

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

(4, )

1. Choose a non-empty subset L of leaves of the so-far built tree such that $|L| \in r$.
2. For each $\ell \in L$ choose an integer $k > 1$ such that $\phi_k > 0$, and one of the $1 \leq c \leq \phi_k$ possible unlabelled trees with k leaves.
3. Replace ℓ with the chosen unlabelled tree, labelling all its internal nodes i.

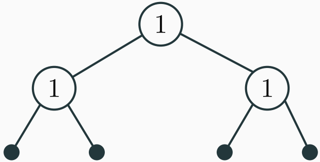
Evolution process for monotonic trees

Example: Monotonic binary trees

Take $\phi(z)$ to be unlabelled binary trees (or Catalan trees)

$$\phi(z) = cat(z) - z = z^2 + 2z^3 + 5z^4 + \dots \quad \text{and } r = \mathbb{N}^*$$

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



1. Choose a non-empty subset L of leaves of the so-far built tree such that $|L| \in r$.
2. For each $\ell \in L$ choose an integer $k > 1$ such that $\phi_k > 0$, and one of the $1 \leq c \leq \phi_k$ possible unlabelled trees with k leaves.
3. Replace ℓ with the chosen unlabelled tree, labelling all its internal nodes i.

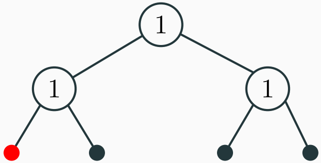
Evolution process for monotonic trees

Example: **Monotonic binary trees**

Take $\phi(z)$ to be unlabelled binary trees (or Catalan trees)

$$\phi(z) = cat(z) - z = z^2 + 2z^3 + 5z^4 + \dots \quad \text{and } r = \mathbb{N}^*$$

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



1. Choose a non-empty subset L of leaves of the so-far built tree such that $|L| \in r$.
2. For each $\ell \in L$ choose an integer $k > 1$ such that $\phi_k > 0$, and one of the $1 \leq c \leq \phi_k$ possible unlabelled trees with k leaves.
3. Replace ℓ with the chosen unlabelled tree, labelling all its internal nodes i.

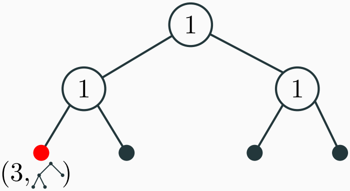
Evolution process for monotonic trees

Example: Monotonic binary trees

Take $\phi(z)$ to be unlabelled binary trees (or Catalan trees)

$$\phi(z) = cat(z) - z = z^2 + 2z^3 + 5z^4 + \ldots \quad \text{and } r = \mathbb{N}^*$$

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



1. Choose a non-empty subset L of leaves of the so-far built tree such that $|L| \in r$.
2. For each $\ell \in L$ choose an integer $k > 1$ such that $\phi_k > 0$, and one of the $1 \leq c \leq \phi_k$ possible unlabelled trees with k leaves.
3. Replace ℓ with the chosen unlabelled tree, labelling all its internal nodes i.

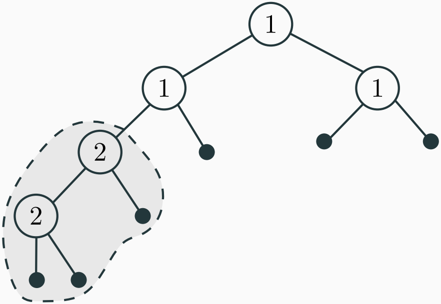
Evolution process for monotonic trees

Example: **Monotonic binary trees**

Take $\phi(z)$ to be unlabelled binary trees (or Catalan trees)

$$\phi(z) = cat(z) - z = z^2 + 2z^3 + 5z^4 + \dots \quad \text{and } r = \mathbb{N}^*$$

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



1. Choose a non-empty subset L of leaves of the so-far built tree such that $|L| \in r$.
2. For each $\ell \in L$ choose an integer $k > 1$ such that $\phi_k > 0$, and one of the $1 \leq c \leq \phi_k$ possible unlabelled trees with k leaves.
3. Replace ℓ with the chosen unlabelled tree, labelling all its internal nodes i.

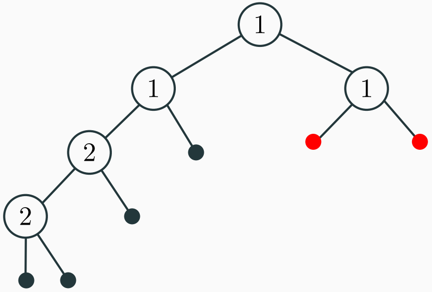
Evolution process for monotonic trees

Example: Monotonic binary trees

Take $\phi(z)$ to be unlabelled binary trees (or Catalan trees)

$$\phi(z) = cat(z) - z = z^2 + 2z^3 + 5z^4 + \dots \quad \text{and } r = \mathbb{N}^*$$

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



1. Choose a non-empty subset L of leaves of the so-far built tree such that $|L| \in r$.
2. For each $\ell \in L$ choose an integer $k > 1$ such that $\phi_k > 0$, and one of the $1 \leq c \leq \phi_k$ possible unlabelled trees with k leaves.
3. Replace ℓ with the chosen unlabelled tree, labelling all its internal nodes i.

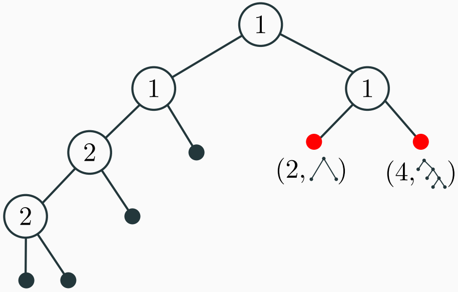
Evolution process for monotonic trees

Example: Monotonic binary trees

Take $\phi(z)$ to be unlabelled binary trees (or Catalan trees)

$$\phi(z) = cat(z) - z = z^2 + 2z^3 + 5z^4 + \ldots \quad \text{and } r = \mathbb{N}^*$$

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



1. Choose a non-empty subset L of leaves of the so-far built tree such that $|L| \in r$.
2. For each $\ell \in L$ choose an integer $k > 1$ such that $\phi_k > 0$, and one of the $1 \leq c \leq \phi_k$ possible unlabelled trees with k leaves.
3. Replace ℓ with the chosen unlabelled tree, labelling all its internal nodes i.

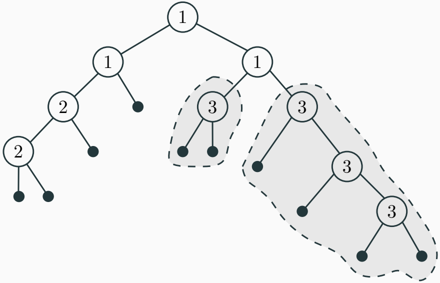
Evolution process for monotonic trees

Example: Monotonic binary trees

Take $\phi(z)$ to be unlabelled binary trees (or Catalan trees)

$$\phi(z) = cat(z) - z = z^2 + 2z^3 + 5z^4 + \dots \quad \text{and } r = \mathbb{N}^*$$

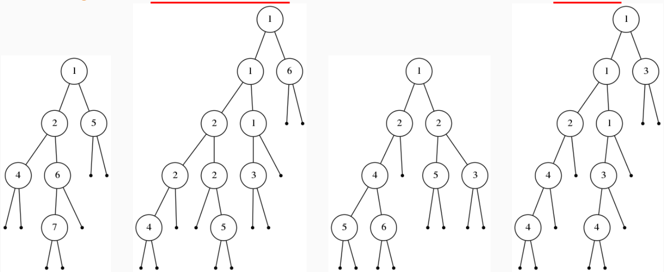
Start at step 0 with a leaf; at each step $i \geq 1$ do:



1. Choose a non-empty subset L of leaves of the so-far built tree such that $|L| \in r$.
2. For each $\ell \in L$ choose an integer $k > 1$ such that $\phi_k > 0$, and one of the $1 \leq c \leq \phi_k$ possible unlabelled trees with k leaves.
3. Replace ℓ with the chosen unlabelled tree, labelling all its internal nodes i.

Increasing labellings of binary trees

As a result we propose new **increasing labellings models** on trees which allow label repetitions:



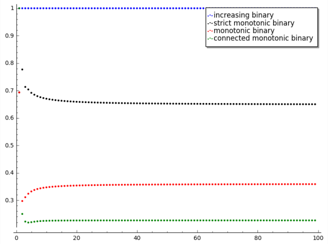
	Increasing	Connected monotonic	Strict monotonic	Monotonic
Label repetitions	no	yes	yes	yes
Branches	strictly increasing	weakly increasing	strictly increasing	weakly increasing
Repetitions		in the same subtree	anywhere	anywhere

- Using different values of ϕ and r we can specify and enumerate trees (counted by their leaves) with the different increasing labellings.
- Theorem I applies to give the asymptotic behaviours.

Comparison of binary trees increasing labellings

	r	$\phi(z)$	Asymptotics	References
Increasing	$\{1\}$	z^2	$(n-1)!$	[FS09], Theorem I
Connected monotonic	$\{1\}$	$(cat(z)-z)$	$c_3\, n!\, n$	Theorem I
Strict monotonic	$\mathbb{N}^*$	z^2	$c_4\,(n-1)!\,(\frac{1}{\ln 2})^n\, n^{-\ln 2}$	[BGGW20], Theorem I
Monotonic (weakly increasing)	$\mathbb{N}^*$	$(cat(z)-z)$	$c_5\,(n-1)!\,(\frac{1}{\ln 2})^n\, n^{\ln 2}$	Theorem I

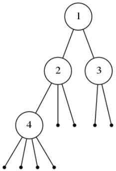
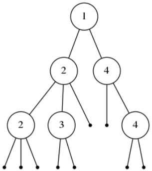
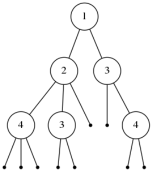
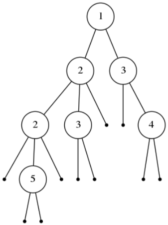
Table 3: Comparison of the asymptotic behaviour of labelled binary trees under different labelling models.



Simulation for $n \in \{1, 100\}$ of binary trees with different increasing labellings divided by their expected asymptotic first order.

Increasing labellings of Schröder trees

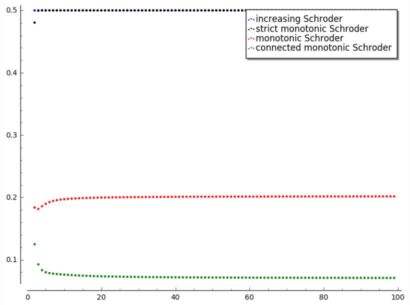
On **Schröder trees** the different increasing labellings give:

	Increasing	Connected monotonic	Strict monotonic	Monotonic
				
Label repetitions	no	yes	yes	yes
Branches	strictly increasing	weakly increasing	strictly increasing	weakly increasing
Repetitions		in the same subtree	anywhere	anywhere

Comparison of Schröder trees under different increasing labellings

	r	$\phi(z)$	Asymptotics	References
Increasing Schröder trees	$\{1\}$	$\frac{z^2}{1-z}$	$\frac{1}{2}\, n!$	[BGN19], Theorem I
C. M. Schröder trees	$\{1\}$	$(S(z)-z)$	$\alpha\ n!\ n^2$	Theorem I
Strictly monotonic Schröder	$\mathbb{N}^*$	$\frac{z^2}{1-z}$	$\frac{1}{2}(n-1)!\ (\frac{1}{\ln 2})^n$	[BGN19], Theorem I
Monotonic Schröder	$\mathbb{N}^*$	$(S(z)-z)$	$\kappa(n-1)!\ (\frac{1}{\ln 2})^n\ n^{2\ln 2}$	Theorem I

Table 4: Comparison of the asymptotic behaviour of families of labelled Schröder trees. $S(z)$ is the GF of Schröder trees.



Simulation for $n \in \{1, 100\}$ of Schröder trees with different increasing labellings divided by their expected asymptotic first order.

Conclusion and future works

Conclusion

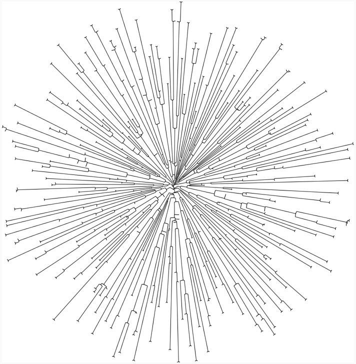
- We studied a general parametrisable evolution process and showed general asymptotic enumeration formulas.
- We also studied three cases separately with their typical parameters $\overset{\text{Need}}{\Longrightarrow}$ Sample uniformly and efficiently large trees.
- It raised questions about the average compaction rate of increasing trees.

Conclusion

- We studied a general parametrisable evolution process and showed general asymptotic enumeration formulas.
- We also studied three cases separately with their typical parameters $\overset{\text{Need}}{\Longrightarrow}$ Sample uniformly and efficiently large trees.
- It raised questions about the average compaction rate of increasing trees.

Random generation: [BGN19, BGMN20]

- Uniform random generation for any ϕ and r is hard in the general case (it relies on the generation of integer partitions).

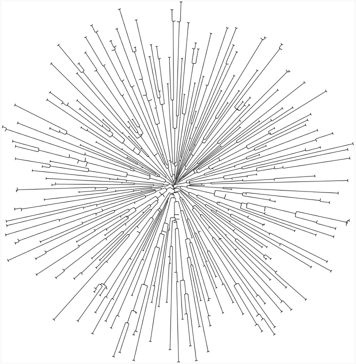


Conclusion

- We studied a general parametrisable evolution process and showed general asymptotic enumeration formulas.
- We also studied three cases separately with their typical parameters $\overset{\text{Need}}{\Longrightarrow}$ Sample uniformly and efficiently large trees.
- It raised questions about the average compaction rate of increasing trees.

Random generation: [BGN19, BGMN20]

- Uniform random generation for any ϕ and r is hard in the general case (it relies on the generation of integer partitions).
- In the three cases presented simplifications occur and trees of large sizes (up to 1000 leaves) can be uniformly sampled.



Conclusion

- We studied a general parametrisable evolution process and showed general asymptotic enumeration formulas.
- We also studied three cases separately with their typical parameters $\overset{\text{Need}}{\Longrightarrow}$ Sample uniformly and efficiently large trees.
- It raised questions about the average compaction rate of increasing trees.

Random generation: [BGN19, BGMN20]

- Uniform random generation for any ϕ and r is hard in the general case (it relies on the generation of integer partitions).
- In the three cases presented simplifications occur and trees of large sizes (up to 1000 leaves) can be uniformly sampled.

Tree compaction: [BGGLN20]

- In the case of simple trees a uniform tree with n nodes has an average compaction rate of $\Theta\left(\frac{n}{\sqrt{\ln n}}\right)$. [FSS90, BMLMN15]

Conclusion

- We studied a general parametrisable evolution process and showed general asymptotic enumeration formulas.
- We also studied three cases separately with their typical parameters $\overset{\text{Need}}{\Longrightarrow}$ Sample uniformly and efficiently large trees.
- It raised questions about the average compaction rate of increasing trees.

Random generation: [BGN19, BGMN20]

- Uniform random generation for any ϕ and r is hard in the general case (it relies on the generation of integer partitions).
- In the three cases presented simplifications occur and trees of large sizes (up to 1000 leaves) can be uniformly sampled.

Tree compaction: [BGGLN20]

- In the case of simple trees a uniform tree with n nodes has an average compaction rate of $\Theta\left(\frac{n}{\sqrt{\ln n}}\right)$. [FSS90, BMLMN15]
- We developed a generic method to compute the compaction rate of increasing trees.

Conclusion

- We studied a general parametrisable evolution process and showed general asymptotic enumeration formulas.
- We also studied three cases separately with their typical parameters $\xrightarrow{\text{Need}}$ Sample uniformly and efficiently large trees.
- It raised questions about the average compaction rate of increasing trees.

Random generation: [BGN19, BGMN20]

- Uniform random generation for any ϕ and r is hard in the general case (it relies on the generation of integer partitions).
- In the three cases presented simplifications occur and trees of large sizes (up to 1000 leaves) can be uniformly sampled.

Tree compaction: [BGGLN20]

- In the case of simple trees a uniform tree with n nodes has an average compaction rate of $\Theta\left(\frac{n}{\sqrt{\ln n}}\right)$. [FSS90, BMLMN15]
- We developed a generic method to compute the compaction rate of increasing trees.
- We applied it on increasing binary trees and get the average compaction rate is $\Theta\left(\frac{n}{\ln n}\right)$ and on recursive tree to get $O\left(\frac{n}{\ln n}\right)$.

Publications and preprints

O. Bodini, A. Genitrini, M. Naima	"Ranked Schröder trees". In Proceedings of the Sixteenth Workshop on Analytic Algorithmics and Combinatorics (ANALCO), 2019.
O. Bodini, A. Genitrini, M. Naima, A. Singh	"Families of Monotonic Trees: Combinatorial Enumeration and Asymptotics". In Proceedings of the 15th International Computer Science Symposium in Russia (CSR), 2020.
O. Bodini, A. Genitrini, C. Mailler, M. Naima	"Strict monotonic trees arising from evolutionary processes: combinatorial and probabilistic study". Submitted to a journal. Available on https://hal.sorbonne-universite.fr/hal-02865198
O. Bodini, A. Genitrini, B. Gittenberger, I. Larcher, M. Naima	"Compaction for two models of logarithmic-depth trees: Analysis and Experiments". Submitted to a journal. Available on https://arxiv.org/abs/2005.12997

Perspectives and future works

Non-plane models
We studied plane trees but it would be interesting to make the same studies on non-plane models.

Perspectives and future works

Non-plane models

We studied plane trees but it would be interesting to make the same studies on non-plane models.

Asymptotic enumeration

- Give the asymptotic behaviour of plane weakly increasing d-ary trees for $d \geq 3$. The case $d = 2$, the problem falls under Theorem I.
- More generally, find universal asymptotic theorems when there are no binary nodes.

Perspectives and future works

Non-plane models

We studied plane trees but it would be interesting to make the same studies on non-plane models.

Asymptotic enumeration

- Give the asymptotic behaviour of plane weakly increasing d-ary trees for $d \geq 3$. The case $d = 2$, the problem falls under Theorem I.
- More generally, find universal asymptotic theorems when there are no binary nodes.

Tree compaction

Is to possible to show that trees belonging to classical increasing have an average compaction size of $\Theta(n/\ln n)$?

Perspectives and future works

Non-plane models
We studied plane trees but it would be interesting to make the same studies on non-plane models.

Asymptotic enumeration

- Give the asymptotic behaviour of plane weakly increasing d-ary trees for $d \geq 3$. The case $d = 2$, the problem falls under Theorem I.
- More generally, find universal asymptotic theorems when there are no binary nodes.

Tree compaction
Is to possible to show that trees belonging to classical increasing have an average compaction size of $\Theta(n/\ln n)$?

Increasing labellings
We obtained results on different increasing labellings of tree structures counted by their number of leaves. What about trees in which we count all nodes as it is mostly the case?

Thank you

References

[BFS92] François Bergeron, Philippe Flajolet, and Bruno Salvy. Varieties of increasing trees. In *CAAP*, pages 24–48, 1992.

[BGGW20] Olivier Bodini, Antoine Genitrini, Bernhard Gittenberger, and Stephan Wagner. On the number of increasing trees with label repetitions. *Discrete Mathematics*, 343(8):111722, 2020.

[BGMN20] Olivier Bodini, Antoine Genitrini, Cécile Mailler, and Mehdi Naima. Strict monotonic trees arising from evolutionary processes: combinatorial and probabilistic study. 2020.

[BGN19] Olivier Bodini, Antoine Genitrini, and Mehdi Naima. Ranked Schröder Trees. In *2019 Proceedings of the Sixteenth Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*, pages 13–26. SIAM, 2019.

[BGNS20] Olivier Bodini, Antoine Genitrini, Mehdi Naima, and Alexandros Singh. Families of Monotonic Trees: Combinatorial Enumeration and Asymptotics. In *15th International Computer Science Symposium in Russia (CSR)*, pages 155–168, 2020.

[BGP16] O. Bodini, A. Genitrini, and F. Peschanski. A quantitative study of pure parallel processes. *Electronic Journal of Combinatorics*, 23(1):1–25, 2016.

[BGR17] Olivier Bodini, Antoine Genitrini, and Nicolas Rolin. Extended boxed product and application to synchronized trees. *Electronic Notes in Discrete Mathematics*, 59:189–202, 2017.

[BMLMN15] Mireille Bousquet-Mélou, Markus Lohrey, Sebastian Maneth, and Eric Noeth. XML Compression via Directed Acyclic Graphs. *Theory of Computing Systems*, 57(4):1322–1371, 2015.

[Drm09] Michael Drmota. *Random trees: An interplay between combinatorics and probability*. Springer Science & Business Media, 2009.

[Fel03] Joseph Felsenstein. *Inferring phylogenies*. Sinauer Associates, 2003.

[FGM+06] Philippe Flajolet, Xavier Gourdon, Conrado Martinez, Philippe Flajolet, Xavier Gourdon, Conrado Martinez, Random Binary, and Search Trees. Patterns in Random Binary Search Trees To cite this version : HAL Id : inria-00073700 apport de recherche. 2006.

[FS09] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009.

[FSS90] Philippe Flajolet, Paolo Sipala, and Jean Marc Steyaert. Analytic variations on the common subexpression problem. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 443 LNCS, pages 220–234. Springer, New York, 1990.

[Mah92] Hosam M Mahmoud. *Evolution of random search trees*. Wiley New York, 1992.

[Moo74] John W Moon. *The distance between nodes in recursive trees*, pages 125–132. London Mathematical Society Lecture Note Series. Cambridge University Press, 1974.

[NH82] Dietmar Najock and C C Heyde. On the number of terminal vertices in certain random trees with an application to stemma construction in philology. *Journal of Applied Probability*, pages 675–680, 1982.

[PU83] Helmut Prodinger and Friedrich J Urbanek. On monotone functions of tree structures. *Discrete Applied Mathematics*, 5(2):223–239, 1983.

[SDH+04] Charles Semple, Philip Daniel, Wim Hordijk, Roderic D.M. Page, and Mike Steel. Supertree algorithms for ancestral divergence dates and nested taxa. *Bioinformatics*, 2004.

[Ste16] Mike Steel. *Phylogeny Discrete and Random Processes in Evolution*, volume 89 of *CBMS-NSF regional conference series in applied mathematics*. SIAM, 2016.

Compaction of trees

- Another interesting equation related to trees is their optimal representation memory.

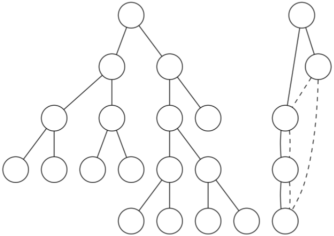


Figure 1: (left) A binary tree of size 17. (right) Its compacted version that has size 5.

Compaction of trees

- Another interesting equation related to trees is their optimal representation memory.
- The idea is that in a single tree, some subtrees can be isomorphic and therefore when compressing the tree we can keep only one occurrence of a repeated subtree and put pointers to it.

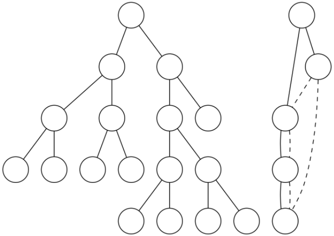


Figure 1: (left) A binary tree of size 17. (right) Its compacted version that has size 5.

Compaction of trees

- The average compaction ration of a simple tree of size n is $\Theta(n/\sqrt{\ln n})$. [FSS90]

Compaction of trees

- The average compaction ration of a simple tree of size n is $\Theta(n/\sqrt{\ln n})$. [FSS90]
- The average compaction of binary increasing trees (equivalently binary search trees) is known to be $\Theta(n/\ln n)$. [FGM+06]

Compaction of trees

- The average compaction ration of a simple tree of size n is $\Theta(n/\sqrt{\ln n})$. [FSS90]
- The average compaction of binary increasing trees (equivalently binary search trees) is known to be $\Theta(n/\ln n)$. [FGM+06]
- We found a generic method to obtain the average compaction rate of classes of increasing trees based on a perturbed generating function.

Compaction of trees

- The average compaction ration of a simple tree of size n is $\Theta(n/\sqrt{\ln n})$. [FSS90]
- The average compaction of binary increasing trees (equivalently binary search trees) is known to be $\Theta(n/\ln n)$. [FGM+06]
- We found a generic method to obtain the average compaction rate of classes of increasing trees based on a perturbed generating function.
- We applied our approach to rederive known results on binary increasing trees and to obtain an upper bound on recursive trees $O(n/\ln n)$. We have reasons to believe that this bound is already sharp.

Random generation

- We are interested in the uniform random generation of trees generated by the parameterised evolution process (the parameters are $\phi(z)$ and r).

Random generation

- We are interested in the uniform random generation of trees generated by the parameterised evolution process (the parameters are $\phi(z)$ and r).
- For the general case, the speciation is not convergent (Boltzmann sampling can not be applied).

Random generation

- We are interested in the uniform random generation of trees generated by the parameterised evolution process (the parameters are $\phi(z)$ and r).
- For the general case, the speciation is not convergent (Boltzmann sampling can not be applied).
- But we have a general recurrence that relies on integer partitions (Recursive generation and unranking can be applied).

Random generation

- We are interested in the uniform random generation of trees generated by the parameterised evolution process (the parameters are $\phi(z)$ and r).
- For the general case, the speciation is not convergent (Boltzmann sampling can not be applied).
- But we have a general recurrence that relies on integer partitions (Recursive generation and unranking can be applied).
- But the complexity of the recursive generation corresponds to the one of generating integer partitions.

Improved algorithms for the three particular cases

- For some values of the parameters $\phi(z)$ and r, simplifications occur. This is the case for the three models that we have presented.

- For some values of the parameters $\phi(z)$ and r, simplifications occur. This is the case for the three models that we have presented.

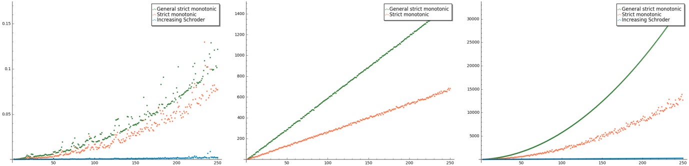


Figure 2: Complexity of uniform samplers for the the three models of increasing Schröder trees. (left) Time complexity of the sampling in milliseconds. (middle) Arithmetic operations on big numbers complexity. (right) Arithmetic operations complexity.

- For some values of the parameters $\phi(z)$ and r, simplifications occur. This is the case for the three models that we have presented.

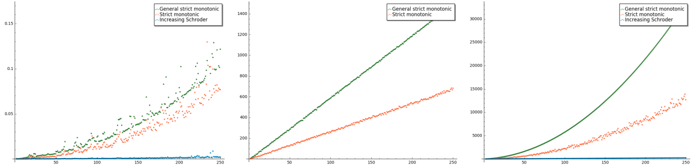


Figure 2: Complexity of uniform samplers for the the three models of increasing Schröder trees. (left) Time complexity of the sampling in milliseconds. (middle) Arithmetic operations on big numbers complexity. (right) Arithmetic operations complexity.

- For the model of increasing Schröder trees ($\phi(z) = z/(1-z)$ and $r = \{1\}$) we have an incremental process (no recursive generation or operations on big numbers).

Simulation of some parameters

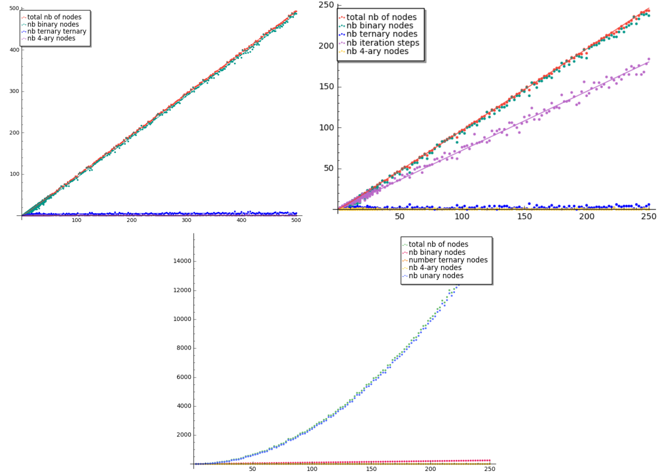


Figure 3: The number of d-ary nodes and the number of iteration steps. (up left) Increasing Schröder trees. (up right) Strict monotonic Schröder trees. (down) general monotonic Schröder trees.