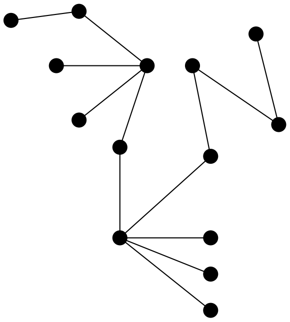
Limit distributions of tree parameters

Stephan Wagner

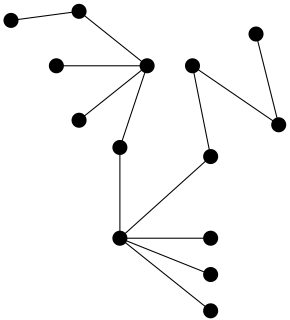
Stellenbosch University

FPSAC, 4 July 2019

Why study trees?

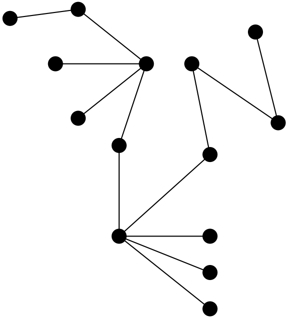


Why study trees?



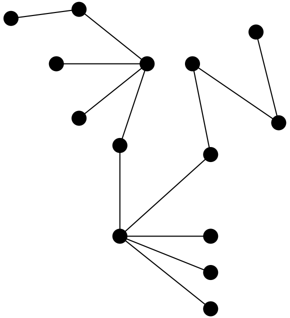
- They are simple.

Why study trees?



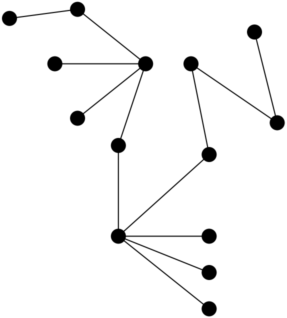
- They are simple.
- They have many nice properties.

Why study trees?



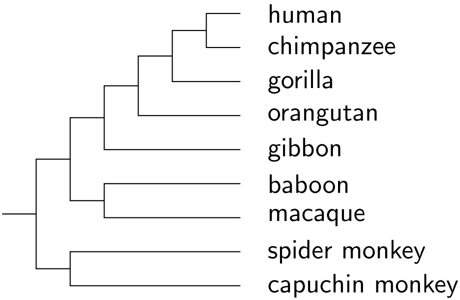
- They are simple.
- They have many nice properties.
- They are useful.

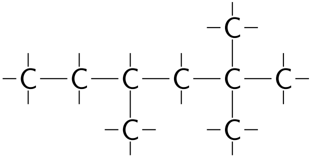
Why study trees?

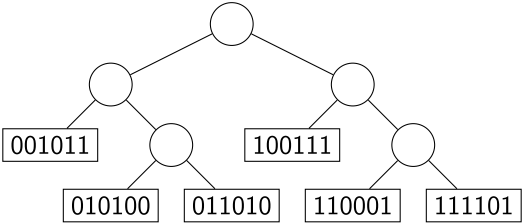


- They are simple.
- They have many nice properties.
- They are useful.

Trees are useful







Families of trees

Trees can

Families of trees

Trees can

- have labelled or unlabelled vertices,

Families of trees

Trees can

- have labelled or unlabelled vertices,
- be rooted or unrooted,

Families of trees

Trees can

- have labelled or unlabelled vertices,
- be rooted or unrooted,
- be plane or non-plane,

Families of trees

Trees can

- have labelled or unlabelled vertices,
- be rooted or unrooted,
- be plane or non-plane,
- have various restrictions (labels, vertex degrees, . . .).

Families of trees

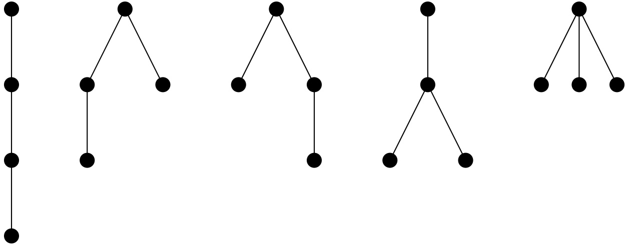
Trees can

- have labelled or unlabelled vertices,
- be rooted or unrooted,
- be plane or non-plane,
- have various restrictions (labels, vertex degrees, ...).

Depending on these, many different classes of trees have been studied in the literature.

Families of trees

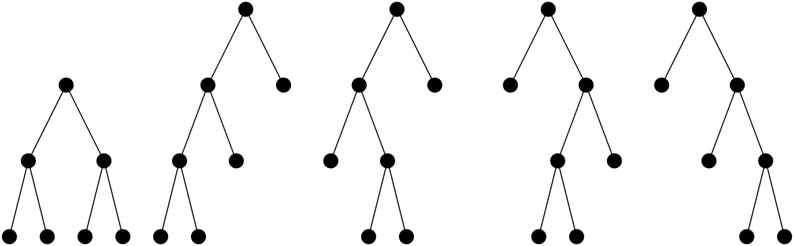
(Planted) plane trees: rooted trees embedded in the plane



The number of plane trees with n vertices is the *Catalan number* $\frac{1}{n}\binom{2n-2}{n-1}$.

Families of trees

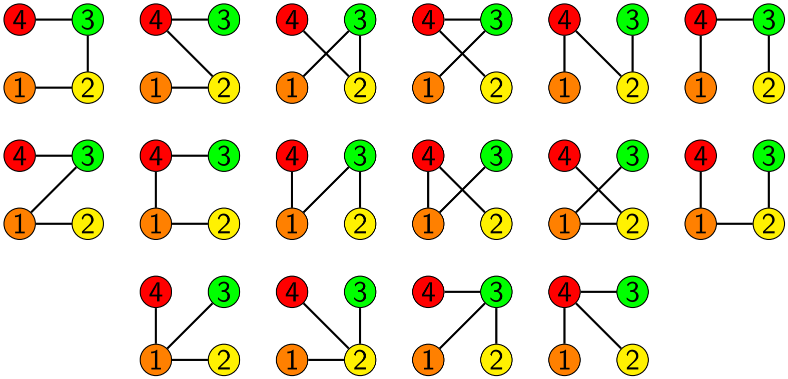
Binary trees: rooted trees where every vertex is either a leaf or has exactly two children (left and right).



The number of binary trees with n internal vertices is the *Catalan number* $\frac{1}{n+1}\binom{2n}{n}$.

Families of trees

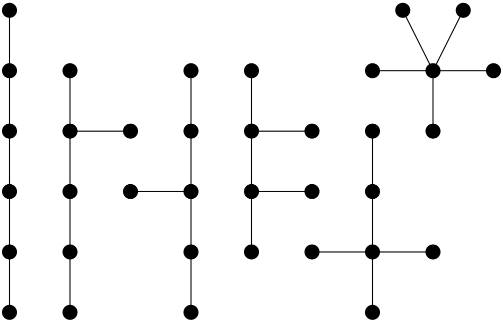
Labelled trees: each vertex has a unique label from 1 up to n (can be rooted or unrooted).



The number of labelled (unrooted) trees with n vertices is n^{n-2}.

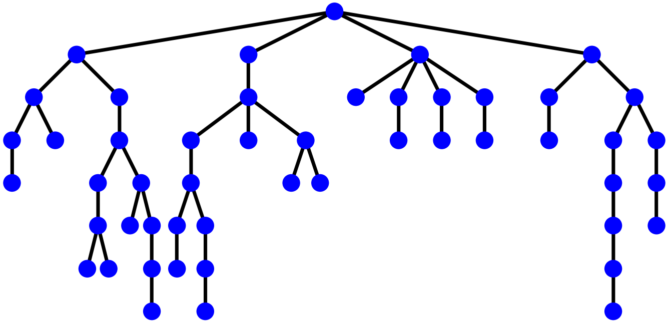
Families of trees

Unlabelled (unrooted) trees:



There is no simple formula for the number of unlabelled trees of a given size. The counting sequence starts $1, 1, 1, 2, 3, 6, 11, 23, 47, \ldots$, and there is an asymptotic formula for the number of trees with n vertices: $0.53495 \cdot n^{-5/2} \cdot 2.95577^n$.

Random trees



A random tree with 50 vertices. What is the underlying model?

Random tree models

Random trees play a role in many areas, from computational biology (phylogenetic trees) to the analysis of algorithms. Depending on the specific application, various random models have been brought forward, such as:

Random tree models

Random trees play a role in many areas, from computational biology (phylogenetic trees) to the analysis of algorithms. Depending on the specific application, various random models have been brought forward, such as:

- Uniform models (e.g. uniformly random labelled or binary trees),

Random tree models

Random trees play a role in many areas, from computational biology (phylogenetic trees) to the analysis of algorithms. Depending on the specific application, various random models have been brought forward, such as:

- Uniform models (e.g. uniformly random labelled or binary trees),
- Branching processes (e.g. Galton-Watson trees),

Random tree models

Random trees play a role in many areas, from computational biology (phylogenetic trees) to the analysis of algorithms. Depending on the specific application, various random models have been brought forward, such as:

- Uniform models (e.g. uniformly random labelled or binary trees),
- Branching processes (e.g. Galton-Watson trees),
- Increasing tree models (e.g. recursive trees),

Random tree models

Random trees play a role in many areas, from computational biology (phylogenetic trees) to the analysis of algorithms. Depending on the specific application, various random models have been brought forward, such as:

- Uniform models (e.g. uniformly random labelled or binary trees),
- Branching processes (e.g. Galton-Watson trees),
- Increasing tree models (e.g. recursive trees),
- Models based on random strings or permutations (e.g. tries, binary search trees).

Uniform models

The simplest type of model uses the uniform distribution on the set of trees of a given order within a specified family (e.g. the family of all labelled trees, all unlabelled trees or all binary trees).

Uniform models

The simplest type of model uses the uniform distribution on the set of trees of a given order within a specified family (e.g. the family of all labelled trees, all unlabelled trees or all binary trees).

The analysis of such models often involves exact counting and generating functions.

Uniform models

The simplest type of model uses the uniform distribution on the set of trees of a given order within a specified family (e.g. the family of all labelled trees, all unlabelled trees or all binary trees).

The analysis of such models often involves exact counting and generating functions.

In particular, this is the case for *simply generated families of trees*.

Simply generated families

On the set of all rooted ordered (plane) trees, we impose a weight function by first specifying a sequence $1 = w_0, w_1, w_2, \ldots$ and then setting

$$w(T) = \prod_{i \geq 0} w_i^{N_i(T)},$$

where $N_i(T)$ is the number of vertices of outdegree i in T. Then we pick a tree of given order n at random, with probabilities proportional to the weights. For instance,

Simply generated families

On the set of all rooted ordered (plane) trees, we impose a weight function by first specifying a sequence $1 = w_0, w_1, w_2, \dots$ and then setting

$$w(T) = \prod_{i \geq 0} w_i^{N_i(T)},$$

where $N_i(T)$ is the number of vertices of outdegree i in T. Then we pick a tree of given order n at random, with probabilities proportional to the weights. For instance,

- $w_0 = w_1 = w_2 = \cdots = 1$ generates random plane trees,

Simply generated families

On the set of all rooted ordered (plane) trees, we impose a weight function by first specifying a sequence $1 = w_0, w_1, w_2, \ldots$ and then setting

$$w(T) = \prod_{i \geq 0} w_i^{N_i(T)},$$

where $N_i(T)$ is the number of vertices of outdegree i in T. Then we pick a tree of given order n at random, with probabilities proportional to the weights. For instance,

- $w_0 = w_1 = w_2 = \cdots = 1$ generates random plane trees,
- $w_0 = w_2 = 1$ (and $w_i = 0$ otherwise) generates random binary trees,

Simply generated families

On the set of all rooted ordered (plane) trees, we impose a weight function by first specifying a sequence $1 = w_0, w_1, w_2, \ldots$ and then setting

$$w(T) = \prod_{i \geq 0} w_i^{N_i(T)},$$

where $N_i(T)$ is the number of vertices of outdegree i in T. Then we pick a tree of given order n at random, with probabilities proportional to the weights. For instance,

- $w_0 = w_1 = w_2 = \cdots = 1$ generates random plane trees,
- $w_0 = w_2 = 1$ (and $w_i = 0$ otherwise) generates random binary trees,
- $w_i = \frac{1}{i!}$ generates random rooted labelled trees.

Branching processes

A classical branching model to generate random trees is the *Galton-Watson tree model*: fix a probability distribution on the set $\{0, 1, 2, \ldots\}$.

Branching processes

A classical branching model to generate random trees is the *Galton-Watson tree model*: fix a probability distribution on the set $\{0, 1, 2, \ldots\}$.

- Start with a single vertex, the root.

Branching processes

A classical branching model to generate random trees is the *Galton-Watson tree model*: fix a probability distribution on the set $\{0, 1, 2, \ldots\}$.

- Start with a single vertex, the root.
- At time t, all vertices at level t (i.e., distance t from the root) produce a number of children, independently at random according to the fixed distribution (some of the vertices might therefore not have children at all).

Branching processes

A classical branching model to generate random trees is the *Galton-Watson tree model*: fix a probability distribution on the set $\{0, 1, 2, \ldots\}$.

- Start with a single vertex, the root.
- At time t, all vertices at level t (i.e., distance t from the root) produce a number of children, independently at random according to the fixed distribution (some of the vertices might therefore not have children at all).
- A random Galton-Watson tree of order n is obtained by conditioning the process.

Branching processes

A classical branching model to generate random trees is the *Galton-Watson tree model*: fix a probability distribution on the set $\{0, 1, 2, \ldots\}$.

- Start with a single vertex, the root.
- At time t, all vertices at level t (i.e., distance t from the root) produce a number of children, independently at random according to the fixed distribution (some of the vertices might therefore not have children at all).
- A random Galton-Watson tree of order n is obtained by conditioning the process.

Simply generated trees and Galton-Watson trees are essentially equivalent. For example, a geometric distribution for branching will result in a random plane tree, a Poisson distribution in a random rooted labelled tree.

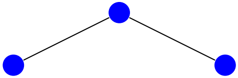
Branching processes

Construction of a random binary tree according to the Galton-Watson model: each vertex has either no children or precisely two.

$t = 0$

Branching processes

Construction of a random binary tree according to the Galton-Watson model: each vertex has either no children or precisely two.

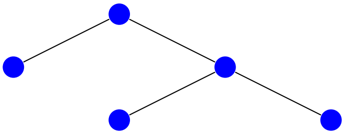


$t = 0$

$t = 1$

Branching processes

Construction of a random binary tree according to the Galton-Watson model: each vertex has either no children or precisely two.



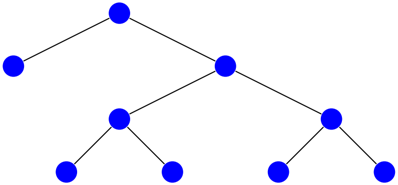
$t = 0$

$t = 1$

$t = 2$

Branching processes

Construction of a random binary tree according to the Galton-Watson model: each vertex has either no children or precisely two.



$t = 0$

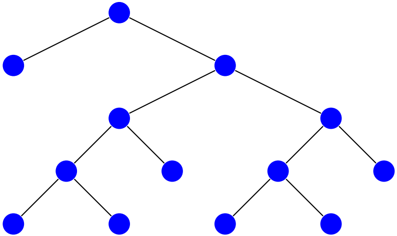
$t = 1$

$t = 2$

$t = 3$

Branching processes

Construction of a random binary tree according to the Galton-Watson model: each vertex has either no children or precisely two.



$t = 0$

$t = 1$

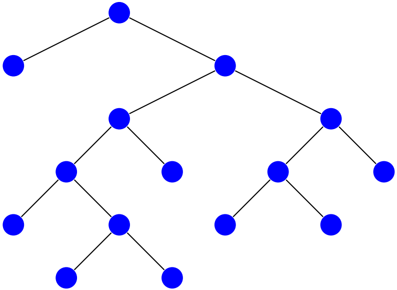
$t = 2$

$t = 3$

$t = 4$

Branching processes

Construction of a random binary tree according to the Galton-Watson model: each vertex has either no children or precisely two.



$t = 0$

$t = 1$

$t = 2$

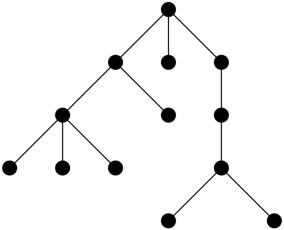
$t = 3$

$t = 4$

$t = 5$

Simply generated and Galton-Watson trees

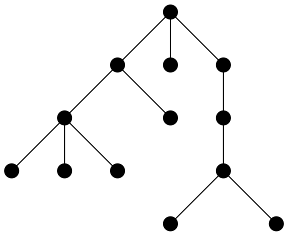
An example:



Consider the Galton-Watson process based on a geometric distribution with $P(X = k) = pq^k$ (where $p = 1 - q$).

Simply generated and Galton-Watson trees

An example:



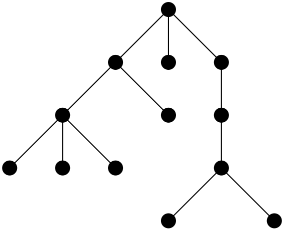
Consider the Galton-Watson process based on a geometric distribution with $P(X = k) = pq^k$ (where $p = 1 - q$).

The tree above has probability

$$p^7(pq)^2(pq^2)^2(pq^3)^2 = p^{13}q^{12},$$

Simply generated and Galton-Watson trees

An example:



Consider the Galton-Watson process based on a geometric distribution with $P(X = k) = pq^k$ (where $p = 1 - q$).

The tree above has probability

$$p^7(pq)^2(pq^2)^2(pq^3)^2 = p^{13}q^{12},$$

as does *every* tree with 13 vertices.

Random increasing trees

Another random model that produces very different shapes uses the following simple process, which generates *random recursive trees*:

Random increasing trees

Another random model that produces very different shapes uses the following simple process, which generates *random recursive trees*:

- Start with the root, which is labelled 1.

Random increasing trees

Another random model that produces very different shapes uses the following simple process, which generates *random recursive trees*:

- Start with the root, which is labelled 1.
- The n-th vertex is attached to one of the previous vertices, uniformly at random.

Random increasing trees

Another random model that produces very different shapes uses the following simple process, which generates *random recursive trees*:

- Start with the root, which is labelled 1.
- The n-th vertex is attached to one of the previous vertices, uniformly at random.

In this way, the labels along any path that starts at the root are increasing. Clearly, there are $(n-1)!$ possible recursive trees of order n, and there are indeed interesting connections to permutations.

Random increasing trees

Another random model that produces very different shapes uses the following simple process, which generates *random recursive trees*:

- Start with the root, which is labelled 1.
- The n-th vertex is attached to one of the previous vertices, uniformly at random.

In this way, the labels along any path that starts at the root are increasing. Clearly, there are $(n-1)!$ possible recursive trees of order n, and there are indeed interesting connections to permutations.

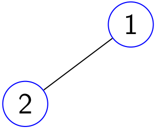
The model can be modified by not choosing a parent uniformly at random, but depending on the current outdegrees (to generate, for example, binary increasing trees).

Random increasing trees

Construction of a recursive tree with 10 vertices:

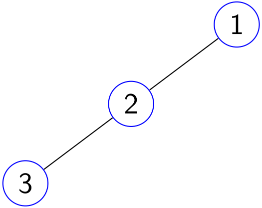
Random increasing trees

Construction of a recursive tree with 10 vertices:



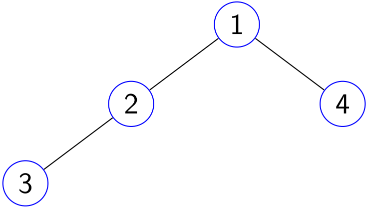
Random increasing trees

Construction of a recursive tree with 10 vertices:



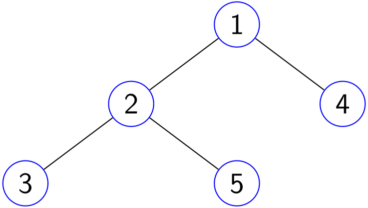
Random increasing trees

Construction of a recursive tree with 10 vertices:



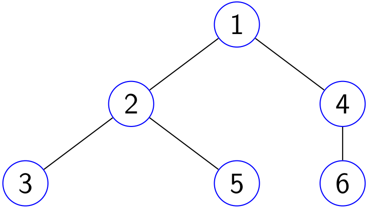
Random increasing trees

Construction of a recursive tree with 10 vertices:



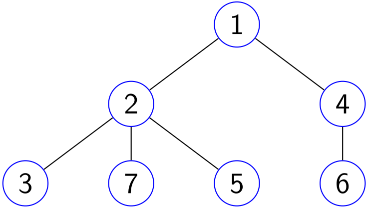
Random increasing trees

Construction of a recursive tree with 10 vertices:



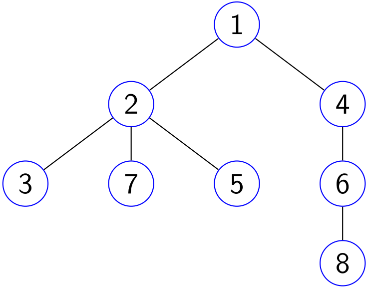
Random increasing trees

Construction of a recursive tree with 10 vertices:



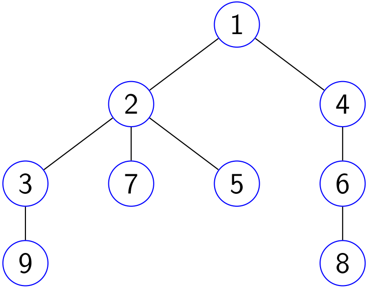
Random increasing trees

Construction of a recursive tree with 10 vertices:



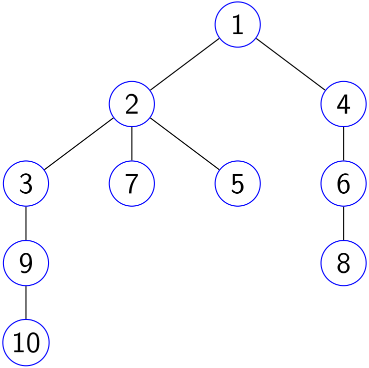
Random increasing trees

Construction of a recursive tree with 10 vertices:



Random increasing trees

Construction of a recursive tree with 10 vertices:



Processes based on random strings

In computer science, tries (short for *retrieval trees*) are a popular data structure for storing strings over a finite alphabet. A random binary trie is obtained as follows:

Processes based on random strings

In computer science, tries (short for *retrieval trees*) are a popular data structure for storing strings over a finite alphabet. A random binary trie is obtained as follows:

- Create n random binary strings of sufficient length, so that they are all distinct (for all practical purposes, one can assume that their length is infinite).

Processes based on random strings

In computer science, tries (short for *retrieval trees*) are a popular data structure for storing strings over a finite alphabet. A random binary trie is obtained as follows:

- Create n random binary strings of sufficient length, so that they are all distinct (for all practical purposes, one can assume that their length is infinite).
- All strings whose first bit is 0 are stored in the left subtree, the others in the right subtree.

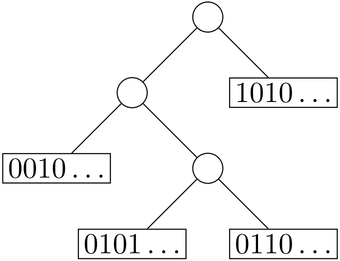
Processes based on random strings

In computer science, tries (short for *retrieval trees*) are a popular data structure for storing strings over a finite alphabet. A random binary trie is obtained as follows:

- Create n random binary strings of sufficient length, so that they are all distinct (for all practical purposes, one can assume that their length is infinite).
- All strings whose first bit is 0 are stored in the left subtree, the others in the right subtree.
- This procedure is repeated recursively.

Processes based on random strings

An example of a trie:



Tree parameters

Many different parameters of trees have been studied in the literature, such as

Tree parameters

Many different parameters of trees have been studied in the literature, such as

- the number of leaves,

Tree parameters

Many different parameters of trees have been studied in the literature, such as

- the number of leaves,
- the number of vertices of a given degree,

Tree parameters

Many different parameters of trees have been studied in the literature, such as

- the number of leaves,
- the number of vertices of a given degree,
- the number of fringe subtrees of a given shape,

Tree parameters

Many different parameters of trees have been studied in the literature, such as

- the number of leaves,
- the number of vertices of a given degree,
- the number of fringe subtrees of a given shape,
- the height (maximum distance of a leaf from the root),

Tree parameters

Many different parameters of trees have been studied in the literature, such as

- the number of leaves,
- the number of vertices of a given degree,
- the number of fringe subtrees of a given shape,
- the height (maximum distance of a leaf from the root),
- the path length (total distance of all vertices from the root),

Tree parameters

Many different parameters of trees have been studied in the literature, such as

- the number of leaves,
- the number of vertices of a given degree,
- the number of fringe subtrees of a given shape,
- the height (maximum distance of a leaf from the root),
- the path length (total distance of all vertices from the root),
- the Wiener index (sum of distances between all pairs of vertices),

Tree parameters

Many different parameters of trees have been studied in the literature, such as

- the number of leaves,
- the number of vertices of a given degree,
- the number of fringe subtrees of a given shape,
- the height (maximum distance of a leaf from the root),
- the path length (total distance of all vertices from the root),
- the Wiener index (sum of distances between all pairs of vertices),
- the number of automorphisms,

Tree parameters

Many different parameters of trees have been studied in the literature, such as

- the number of leaves,
- the number of vertices of a given degree,
- the number of fringe subtrees of a given shape,
- the height (maximum distance of a leaf from the root),
- the path length (total distance of all vertices from the root),
- the Wiener index (sum of distances between all pairs of vertices),
- the number of automorphisms,
- the total number of subtrees,

Tree parameters

Many different parameters of trees have been studied in the literature, such as

- the number of leaves,
- the number of vertices of a given degree,
- the number of fringe subtrees of a given shape,
- the height (maximum distance of a leaf from the root),
- the path length (total distance of all vertices from the root),
- the Wiener index (sum of distances between all pairs of vertices),
- the number of automorphisms,
- the total number of subtrees,
- the number of independent sets or matchings,

Tree parameters

Many different parameters of trees have been studied in the literature, such as

- the number of leaves,
- the number of vertices of a given degree,
- the number of fringe subtrees of a given shape,
- the height (maximum distance of a leaf from the root),
- the path length (total distance of all vertices from the root),
- the Wiener index (sum of distances between all pairs of vertices),
- the number of automorphisms,
- the total number of subtrees,
- the number of independent sets or matchings,
- the spectrum.

A general question

Given a family of trees (a random tree model) and a tree parameter, what can we say about . . .

A general question

Given a family of trees (a random tree model) and a tree parameter, what can we say about . . .

- . . . the average value of the parameter among all trees with n vertices?

A general question

Given a family of trees (a random tree model) and a tree parameter, what can we say about . . .

- . . . the average value of the parameter among all trees with n vertices?
- . . . the variance or higher moments?

A general question

Given a family of trees (a random tree model) and a tree parameter, what can we say about . . .

- . . . the average value of the parameter among all trees with n vertices?
- . . . the variance or higher moments?
- . . . the distribution?

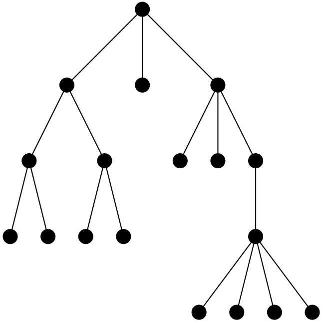
A general question

Given a family of trees (a random tree model) and a tree parameter, what can we say about . . .

- . . . the average value of the parameter among all trees with n vertices?
- . . . the variance or higher moments?
- . . . the distribution?

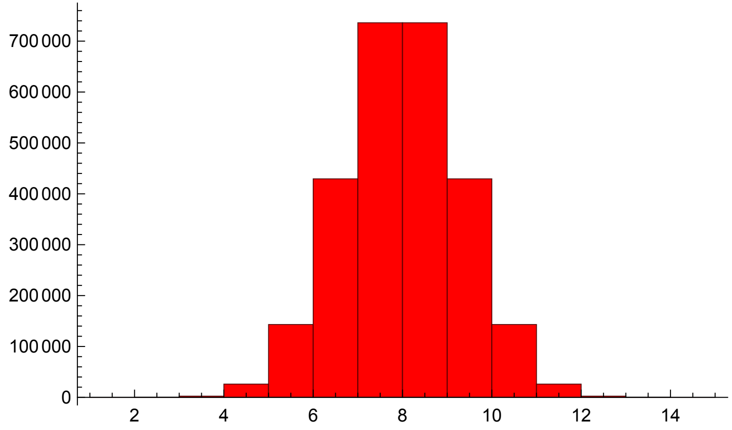
These questions become particularly relevant when n is large.

Some examples of parameters



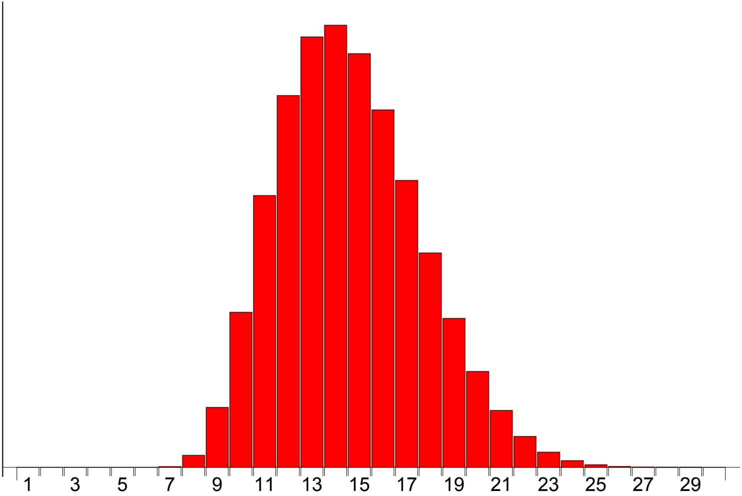
The tree above has 11 leaves, 2 "cherries", height 4, path length 44, 384 automorphisms and 3945 subtrees.

Distribution of parameters: some examples



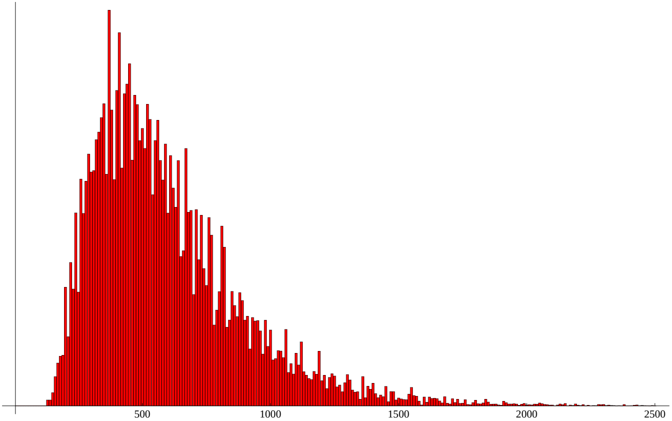
Distribution of the number of leaves in plane trees with 15 vertices. Plane trees with n vertices and k leaves are counted by the Narayana numbers $N_{n,k} = \frac{1}{n-1}\binom{n-1}{k}\binom{n-1}{k-1}$.

Distribution of parameters: some examples



Distribution of the height in binary trees with 30 internal vertices.

Distribution of parameters: some examples



Distribution of the number of subtrees in labelled trees with 15 vertices.

Distributional results

In the following, let $\mathcal{F}$ be either a simply generated family of trees or the family of unlabelled rooted trees (Pólya trees), which is not simply generated, but has similar properties.

Distributional results

In the following, let $\mathcal{F}$ be either a simply generated family of trees or the family of unlabelled rooted trees (Pólya trees), which is not simply generated, but has similar properties.

We consider a random element $\mathcal{T}_n$ of $\mathcal{F}$ with n vertices.

Distributional results

In the following, let $\mathcal{F}$ be either a simply generated family of trees or the family of unlabelled rooted trees (Pólya trees), which is not simply generated, but has similar properties.

We consider a random element $\mathcal{T}_n$ of $\mathcal{F}$ with n vertices.

For some parameter P, what can we say about the distribution of $P(\mathcal{T}_n)$?

The number of leaves

Theorem (Kolchin 1984, Drmota + Gittenberger 1999, Janson 2016)

For every family $\mathcal{F}$*, there exist constants* $\mu > 0$ *and* $\sigma^2 > 0$ *such that the number of leaves* $L(\mathcal{T}_n)$ *of a random tree* $\mathcal{T}_n$ *in* $\mathcal{F}$ *has mean* $\mu_n \sim \mu n$ *and variance* $\sigma_n^2 \sim \sigma^2 n$*.*

The number of leaves

Theorem (Kolchin 1984, Drmota + Gittenberger 1999, Janson 2016)

For every family $\mathcal{F}$, there exist constants $\mu > 0$ and $\sigma^2 > 0$ such that the number of leaves $L(\mathcal{T}_n)$ of a random tree $\mathcal{T}_n$ in $\mathcal{F}$ has mean $\mu_n \sim \mu n$ and variance $\sigma_n^2 \sim \sigma^2 n$.

Moreover, the renormalised random variable

$$X_n = \frac{L(\mathcal{T}_n) - \mu n}{\sqrt{\sigma^2 n}}$$

converges weakly to a standard normal distribution $N(0, 1)$.

The number of leaves

Theorem (Kolchin 1984, Drmota + Gittenberger 1999, Janson 2016)

For every family $\mathcal{F}$, there exist constants $\mu > 0$ and $\sigma^2 > 0$ such that the number of leaves $L(\mathcal{T}_n)$ of a random tree $\mathcal{T}_n$ in $\mathcal{F}$ has mean $\mu_n \sim \mu n$ and variance $\sigma_n^2 \sim \sigma^2 n$.

Moreover, the renormalised random variable

$$X_n = \frac{L(\mathcal{T}_n) - \mu n}{\sqrt{\sigma^2 n}}$$

converges weakly to a standard normal distribution $N(0,1)$.

The theorem generalises to the number of vertices with a given degree or the number of fringe subtrees of a given shape.

The height

Theorem (Flajolet, Gao, Odlyzko + Richmond 1993, Drmota + Gittenberger 2010)

For every family $\mathcal{F}$, there exists a constant $\mu > 0$ such that the height $H(\mathcal{T}_n)$ of a random tree $\mathcal{T}_n$ in $\mathcal{F}$ has mean $\mu_n \sim \mu\sqrt{n}$.

The height

Theorem (Flajolet, Gao, Odlyzko + Richmond 1993, Drmota + Gittenberger 2010)

For every family $\mathcal{F}$, there exists a constant $\mu > 0$ such that the height $H(\mathcal{T}_n)$ of a random tree $\mathcal{T}_n$ in $\mathcal{F}$ has mean $\mu_n \sim \mu\sqrt{n}$.

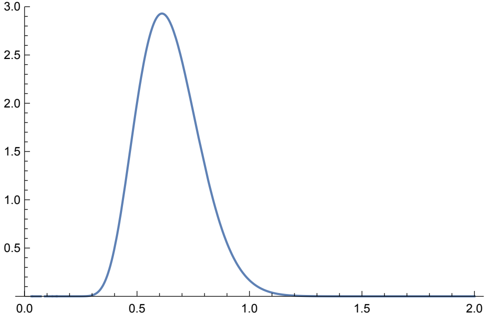
Moreover, the renormalised random variable

$$X_n = \frac{H(\mathcal{T}_n)}{c\sqrt{n}},$$

where $c = \frac{45\zeta(3)\mu}{2\pi^{5/2}}$, converges weakly to a so-called theta distribution, characterised by the density function

$$f(t) = \frac{4\pi^{5/2}}{3\zeta(3)} t^4 \sum_{m \geq 1} (m\pi)^2 (2(m\pi t)^2 - 3) \exp(-(m\pi t)^2).$$

The height



The theta distribution: limiting distribution of the height.

Path length and Wiener index

Theorem (Takács 1993, Janson 2003, SW 2012)

For every family $\mathcal{F}$, there exists a constant $\mu > 0$ such that the path length $D(\mathcal{T}_n)$ and the Wiener index $W(\mathcal{T}_n)$ of a random tree $\mathcal{T}_n$ in $\mathcal{F}$ have means $\mu_n^D \sim \mu n^{3/2}$ and $\mu_n^W \sim \frac{\mu}{2} n^{5/2}$ respectively.

Path length and Wiener index

Theorem (Takács 1993, Janson 2003, SW 2012)

For every family $\mathcal{F}$, there exists a constant $\mu > 0$ such that the path length $D(\mathcal{T}_n)$ and the Wiener index $W(\mathcal{T}_n)$ of a random tree $\mathcal{T}_n$ in $\mathcal{F}$ have means $\mu_n^D \sim \mu n^{3/2}$ and $\mu_n^W \sim \frac{\mu}{2} n^{5/2}$ respectively.

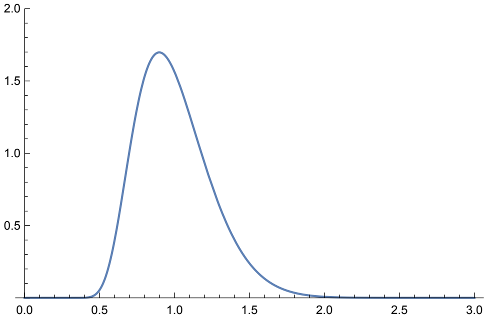
Moreover, the renormalised random variables

$$X_n = \frac{D(\mathcal{T}_n)}{\mu n^{3/2}} \qquad \text{and} \qquad Y_n = \frac{W(\mathcal{T}_n)}{\mu n^{5/2}}$$

converge weakly to random variables given in terms of a normalised Brownian excursion $e(t)$ on $[0,1]$:

$$\sqrt{\frac{8}{\pi}} \int_0^1 e(t)\,dt \quad \text{and} \quad \sqrt{\frac{8}{\pi}} \iint_{0<s<t<1} \Big(e(s) + e(t) - 2 \min_{s \le u \le t} e(u)\Big)\,ds\,dt.$$

The path length



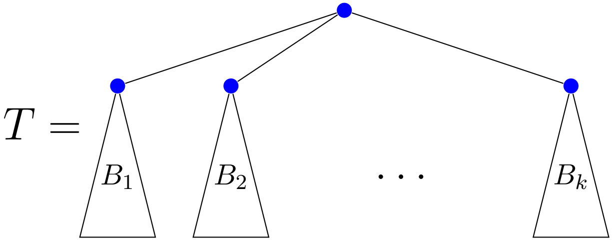
The Airy distribution: limiting distribution of the path length.

Additive functionals: a general concept

A tree parameter is called an *additive functional* if it can be computed by adding its values for all the branches and adding a "toll function" that also depends on the tree.

Additive functionals: a general concept

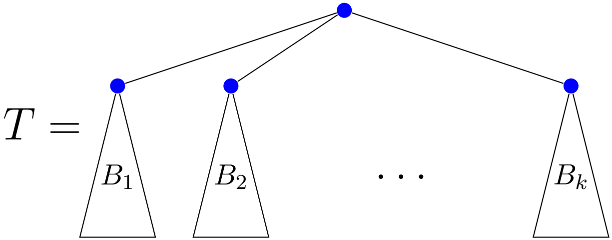
A tree parameter is called an *additive functional* if it can be computed by adding its values for all the branches and adding a “toll function” that also depends on the tree.



$$F(T) = F(B_1) + F(B_2) + \cdots + F(B_k) + f(T).$$

Additive functionals: a general concept

A tree parameter is called an *additive functional* if it can be computed by adding its values for all the branches and adding a "toll function" that also depends on the tree.



$$F(T) = F(B_1) + F(B_2) + \cdots + F(B_k) + f(T).$$

Remark

The recursion remains true for the tree $T = \bullet$ of order 1 if we assume without loss of generality that $f(\bullet) = F(\bullet)$.

An equivalent definition

The fringe subtree T_v associated with a vertex v of a tree T is the subtree consisting of v and all its descendants.

An equivalent definition

The fringe subtree T_v associated with a vertex v of a tree T is the subtree consisting of v and all its descendants.

One can see by induction that the recursion

$$F(T) = F(B_1) + F(B_2) + \cdots + F(B_k) + f(T)$$

is equivalent to the formula

$$F(T) = \sum_v f(T_v).$$

Some examples

- The number of leaves, corresponding to the toll function

$$f(T) = \begin{cases} 1 & |T| = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Some examples

- The number of leaves, corresponding to the toll function

$$f(T) = \begin{cases} 1 & |T| = 1, \\ 0 & \text{otherwise.} \end{cases}$$

- More generally, the number of occurrences of a fixed rooted tree H:

$$f(T) = \begin{cases} 1 & T \simeq H, \\ 0 & \text{otherwise.} \end{cases}$$

Some examples

- The number of leaves, corresponding to the toll function

$$f(T) = \begin{cases} 1 & |T| = 1, \\ 0 & \text{otherwise.} \end{cases}$$

- More generally, the number of occurrences of a fixed rooted tree H:

$$f(T) = \begin{cases} 1 & T \simeq H, \\ 0 & \text{otherwise.} \end{cases}$$

- The number of vertices whose outdegree is a fixed number k:

$$f(T) = \begin{cases} 1 & \text{if the root of } T \text{ has outdegree } k, \\ 0 & \text{otherwise.} \end{cases}$$

Some more examples

- The path length, i.e., the sum of the distances from the root to all vertices, can be obtained from the toll function $f(T) = |T| - 1$:

$$P(T) = \sum_{i=1}^{k} (P(B_i) + |B_i|) = |T| - 1 + \sum_{i=1}^{k} P(B_i).$$

Some more examples

- The path length, i.e., the sum of the distances from the root to all vertices, can be obtained from the toll function $f(T) = |T| - 1$:

$$P(T) = \sum_{i=1}^{k} (P(B_i) + |B_i|) = |T| - 1 + \sum_{i=1}^{k} P(B_i).$$

- The log-product of the subtree sizes, also called the "shape functional", corresponds to $f(T) = \log |T|$. It is related to the number of linear extensions:

$$\mathrm{LE}(T) = \binom{|T| - 1}{|B_1|, |B_2|, \ldots, |B_k|} \prod_{i=1}^{k} \mathrm{LE}(B_i),$$

Some more examples

- The path length, i.e., the sum of the distances from the root to all vertices, can be obtained from the toll function $f(T) = |T| - 1$:

$$P(T) = \sum_{i=1}^{k} (P(B_i) + |B_i|) = |T| - 1 + \sum_{i=1}^{k} P(B_i).$$

- The log-product of the subtree sizes, also called the "shape functional", corresponds to $f(T) = \log |T|$. It is related to the number of linear extensions:

$$\mathrm{LE}(T) = \binom{|T| - 1}{|B_1|, |B_2|, \ldots, |B_k|} \prod_{i=1}^{k} \mathrm{LE}(B_i),$$

thus

$$\log \frac{|T|!}{\mathrm{LE}(T)} = \log |T| + \sum_{i=1}^{n} \log \frac{|B_i|!}{\mathrm{LE}(B_i)}.$$

Even more examples

- The size of the automorphism group: if $c_1, c_2, \ldots, c_r$ are the multiplicities of the different isomorphism classes of branches, we have

$$|\operatorname{Aut}(T)| = \prod_{i=1}^{k} |\operatorname{Aut}(B_i)| \cdot \prod_{j=1}^{r} c_j!,$$

Even more examples

- The size of the automorphism group: if $c_1, c_2, \ldots, c_r$ are the multiplicities of the different isomorphism classes of branches, we have

$$|\operatorname{Aut}(T)| = \prod_{i=1}^{k} |\operatorname{Aut}(B_i)| \cdot \prod_{j=1}^{r} c_j!,$$

thus

$$\log|\operatorname{Aut}(T)| = \sum_{i=1}^{k} \log|\operatorname{Aut}(B_i)| + \sum_{j=1}^{r} \log(c_j!).$$

Even more examples

- The size of the automorphism group: if $c_1, c_2, \ldots, c_r$ are the multiplicities of the different isomorphism classes of branches, we have

$$|\operatorname{Aut}(T)| = \prod_{i=1}^{k} |\operatorname{Aut}(B_i)| \cdot \prod_{j=1}^{r} c_j!,$$

thus

$$\log|\operatorname{Aut}(T)| = \sum_{i=1}^{k} \log|\operatorname{Aut}(B_i)| + \sum_{j=1}^{r} \log(c_j!).$$

- The multiplicity of some eigenvalue λ:

Even more examples

- The size of the automorphism group: if $c_1, c_2, \dots, c_r$ are the multiplicities of the different isomorphism classes of branches, we have

$$|\operatorname{Aut}(T)| = \prod_{i=1}^{k} |\operatorname{Aut}(B_i)| \cdot \prod_{j=1}^{r} c_j!,$$

  thus

$$\log|\operatorname{Aut}(T)| = \sum_{i=1}^{k} \log|\operatorname{Aut}(B_i)| + \sum_{j=1}^{r} \log(c_j!).$$

- The multiplicity of some eigenvalue λ:

$$N_\lambda(T) = \sum_{i=1}^{k} N_\lambda(B_i) + \epsilon_\lambda(T),$$

  where $\epsilon_\lambda(T) \in \{-1, 0, 1\}$.

Yet another example

- The number of subtrees: it is somewhat more convenient to work with the number $s_1(T)$ of subtrees that contain the root (the difference turns out to be asymptotically negligible).

Yet another example

- The number of subtrees: it is somewhat more convenient to work with the number $s_1(T)$ of subtrees that contain the root (the difference turns out to be asymptotically negligible).

The following recursion in terms of the branches $B_1, B_2, \dots, B_k$ holds:

$$s_1(T) = \prod_{i=1}^{k} (1 + s_1(B_i)).$$

Yet another example

- The number of subtrees: it is somewhat more convenient to work with the number $s_1(T)$ of subtrees that contain the root (the difference turns out to be asymptotically negligible).

The following recursion in terms of the branches $B_1, B_2, \ldots, B_k$ holds:

$$s_1(T) = \prod_{i=1}^{k} (1 + s_1(B_i)).$$

Hence

$$\log(1 + s_1(T)) = \sum_{i=1}^{k} \log(1 + s_1(B_i)) + \log(1 + s_1(T)^{-1}).$$

This means that $\log(1 + s_1(T))$ is additive with toll function $f(T) = \log(1 + s_1(T)^{-1})$.

General results

Theorem (SW 2015, Janson 2016, Ralaivaosaona + Šileikis + SW 2019)

Under suitable technical conditions, an additive functional F *on a family* $\mathcal{F}$ *of trees satisfies a central limit theorem:*

General results

Theorem (SW 2015, Janson 2016, Ralaivaosaona + Šileikis + SW 2019)

Under suitable technical conditions, an additive functional F *on a family* $\mathcal{F}$ *of trees satisfies a central limit theorem:*

There exist constants μ *and* σ^2 *such that mean and variance of* $F(\mathcal{T}_n)$ *for a random tree* $\mathcal{T}_n$ *in* $\mathcal{F}$ *are* $\mu_n \sim \mu n$ *and* $\sigma_n^2 \sim \sigma^2 n$*.*

General results

Theorem (SW 2015, Janson 2016, Ralaivaosaona + Šileikis + SW 2019)

Under suitable technical conditions, an additive functional F on a family $\mathcal{F}$ of trees satisfies a central limit theorem:

There exist constants μ and σ^2 such that mean and variance of $F(\mathcal{T}_n)$ for a random tree $\mathcal{T}_n$ in $\mathcal{F}$ are $\mu_n \sim \mu n$ and $\sigma_n^2 \sim \sigma^2 n$.

Moreover, the renormalised random variable

$$X_n = \frac{F(\mathcal{T}_n) - \mu n}{\sqrt{\sigma^2 n}}$$

converges weakly to a standard normal distribution.

General results

What are “suitable technical conditions”?

General results

What are “suitable technical conditions”?

In a nutshell, there are two types of conditions:

- The toll function f is “small” (at least on average) for large trees.
- The toll function f is “local” (only depends on a small neighbourhood of the root), at least approximately.

General results

Similar results are known for other tree models, specifically:

- increasing tree families: recursive trees, d-ary increasing trees, (generalised) plane-oriented recursive trees (Holmgren + Janson 2015, Holmgren + Janson + Šileikis 2017, Ralaivaosaona + SW 2019)
- d-ary search trees (Holmgren + Janson + Šileikis 2017)

General results

Similar results are known for other tree models, specifically:

- increasing tree families: recursive trees, d-ary increasing trees, (generalised) plane-oriented recursive trees (Holmgren + Janson 2015, Holmgren + Janson + Šileikis 2017, Ralaivaosaona + SW 2019)
- d-ary search trees (Holmgren + Janson + Šileikis 2017)

Proofs involve:

- combinatorial techniques (generating functions, analytic combinatorics, . . .)
- probabilistic techniques (growth processes, urn models, . . .)

Examples covered

Many different examples are covered by one or more of the technical conditions, in particular:

(N) = normal   (L) = lognormal

Examples covered

Many different examples are covered by one or more of the technical conditions, in particular:

- the number of leaves (N),

(N) = normal (L) = lognormal

Examples covered

Many different examples are covered by one or more of the technical conditions, in particular:

- the number of leaves (N),
- the number of vertices of degree k (N),

(N) $=$ normal (L) $=$ lognormal

Examples covered

Many different examples are covered by one or more of the technical conditions, in particular:

- the number of leaves (N),
- the number of vertices of degree k (N),
- the number of fringe subtrees of a given type (N),

(N) = normal (L) = lognormal

Examples covered

Many different examples are covered by one or more of the technical conditions, in particular:

- the number of leaves (N),
- the number of vertices of degree k (N),
- the number of fringe subtrees of a given type (N),
- the number of subtrees (L),

(N) = normal (L) = lognormal

Examples covered

Many different examples are covered by one or more of the technical conditions, in particular:

- the number of leaves (N),
- the number of vertices of degree k (N),
- the number of fringe subtrees of a given type (N),
- the number of subtrees (L),
- the number of independent sets (L),

(N) $=$ normal (L) $=$ lognormal

Examples covered

Many different examples are covered by one or more of the technical conditions, in particular:

- the number of leaves (N),
- the number of vertices of degree k (N),
- the number of fringe subtrees of a given type (N),
- the number of subtrees (L),
- the number of independent sets (L),
- the number of matchings (L),

(N) $=$ normal   (L) $=$ lognormal

Examples covered

Many different examples are covered by one or more of the technical conditions, in particular:

- the number of leaves (N),
- the number of vertices of degree k (N),
- the number of fringe subtrees of a given type (N),
- the number of subtrees (L),
- the number of independent sets (L),
- the number of matchings (L),
- the independence number (N),

(N) = normal (L) = lognormal

Examples covered

Many different examples are covered by one or more of the technical conditions, in particular:

- the number of leaves (N),
- the number of vertices of degree k (N),
- the number of fringe subtrees of a given type (N),
- the number of subtrees (L),
- the number of independent sets (L),
- the number of matchings (L),
- the independence number (N),
- the matching number (N),

(N) = normal   (L) = lognormal

Examples covered

Many different examples are covered by one or more of the technical conditions, in particular:

- the number of leaves (N),
- the number of vertices of degree k (N),
- the number of fringe subtrees of a given type (N),
- the number of subtrees (L),
- the number of independent sets (L),
- the number of matchings (L),
- the independence number (N),
- the matching number (N),
- the average subtree size (N).

(N) = normal   (L) = lognormal

Future work

- Random tree models that have not been covered yet,

Future work

- Random tree models that have not been covered yet,
- Tree-like graph classes,

Future work

- Random tree models that have not been covered yet,
- Tree-like graph classes,
- Parameters that are not covered by any of the existing general conditions,

Future work

- Random tree models that have not been covered yet,
- Tree-like graph classes,
- Parameters that are not covered by any of the existing general conditions,
- General schemes for additive parameters whose limit distributions are not normal,

Future work

- Random tree models that have not been covered yet,
- Tree-like graph classes,
- Parameters that are not covered by any of the existing general conditions,
- General schemes for additive parameters whose limit distributions are not normal,
- Parameters that follow different types of recursion (e.g.: $\max$ instead of $\sum$),

Future work

- Random tree models that have not been covered yet,
- Tree-like graph classes,
- Parameters that are not covered by any of the existing general conditions,
- General schemes for additive parameters whose limit distributions are not normal,
- Parameters that follow different types of recursion (e.g.: $\max$ instead of $\sum$),
- ...