Split Trees and Galton-Watson Trees

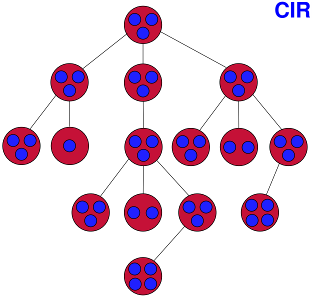
Two Important Classes of Random Trees

Cecilia Holmgren

Department of Mathematics, Uppsala University, Sweden

Dynamics on random graphs and random maps,

CIRM, October 26, 2017



Split Trees and Galton-Watson Trees

I will talk about two important classes of random trees:

- **Split trees** were introduced by Devroye (1998) for unifying many important random trees of logarithmic height. They are interesting not least because of their usefulness as models of **sorting algorithms** in computer science; for instance can the the well-known **Quicksort** algorithm (introduced by Hoare [1960]) be depicted as the **binary search tree**
- **Galton-Watson trees** were introduced already in 1875 to describe under which conditions a (noble) family name would die out or survive forever.
- The **conditioned Galton-Watson trees** (also called **simply-generated trees**) are conditioned on a given total size of the number of nodes and represent important random trees of non-logarithmic height, such as for instance the **Cayley tree**

UPPSALA
UNIVERSITET

My Research Group at Uppsala University: Postdocs and PhD students

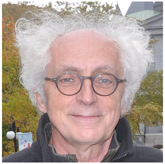
Xing Shi Cai	CH	Colin Desmarais	Tony Johansson	Fiona Skerman
Postdoc (PhD McGill, CAN)		PhD student (M.Sc Manitoba, CAN)	Postdoc (PhD Carnegie Mellon,USA)	Postdoc (PhD Oxford, UK)

Cecilia Holmgren

UPPSALA UNIVERSITET

Some Collaborators on Random Trees

Svante Janson
Uppsala Univ. SWE

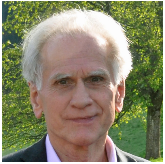


Luc Devroye
McGill Univ. CAN

Nicolas Broutin
INRIA, Paris FR

Louigi Addario-Berry
McGill Univ. CAN

Henning Sulzbach
Univ. of Birmingham UK



Béla Bollobás
Univ. of Cambridge UK
& Univ. of Memphis USA

Karen Gunderson
Univ. of Manitoba CAN

Stephan Wagner
Stellenbosch Univ.
S.Africa

Examples of Split Trees

- The class of split trees includes many important random trees of logarithmic height, e.g., **binary search trees**, **m-ary search trees**, **quadtrees**, **median of $(2k+1)$-trees**, **simplex trees** and **tries**

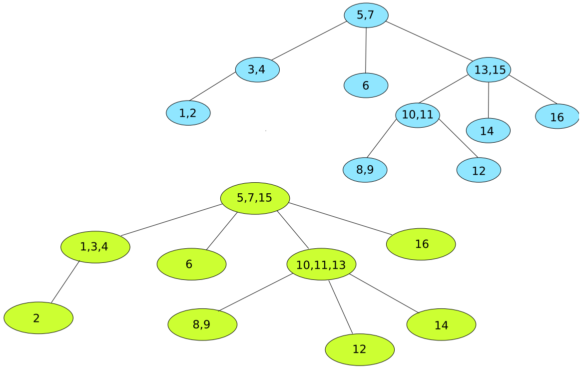
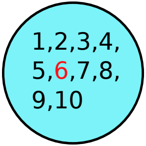
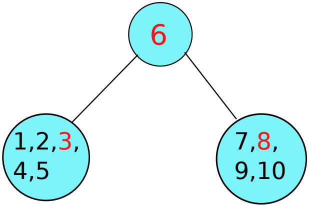


Figure: *A 3-ary and a 4-ary search tree.*

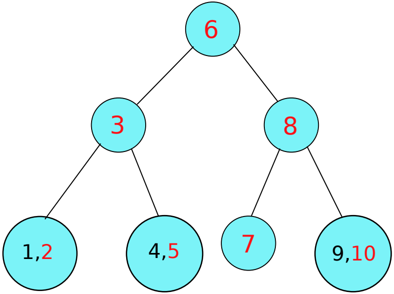
The Binary Search Tree is an Example of a Split Tree



The Binary Search Tree is an Example of a Split Tree

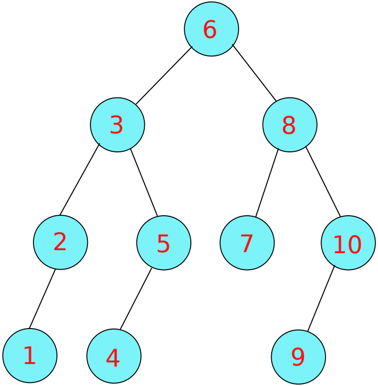


The Binary Search Tree is an Example of a Split Tree

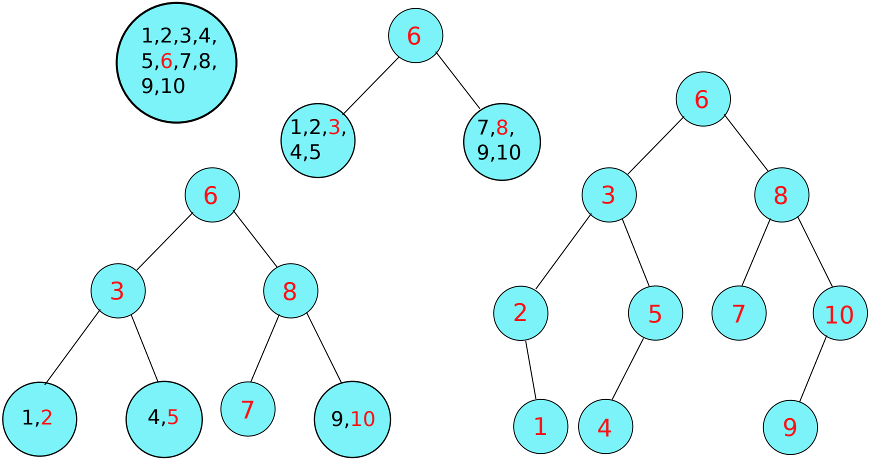


Cecilia Holmgren

The Binary Search Tree is an Example of a Split Tree

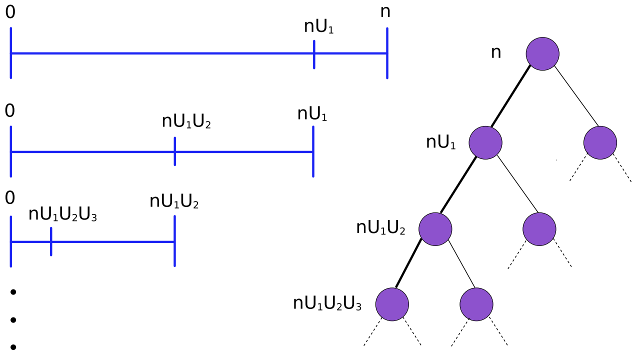


The Binary Search Tree is an Example of a Split Tree



Cecilia Holmgren

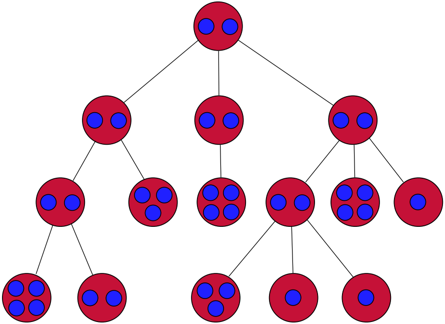
The Binary Search Tree (continued)



- Since the rank of the root's key is equally likely to be $\{1, 2, \ldots, n\}$, the size of its left subtree is distributed as $\lfloor nU \rfloor$, where U is a uniform $U(0, 1)$ random variable

Important Parameters for Split Trees

(Devroye 1998)



- Branch factor b=3
- Vertex capacity s=4>0
- The random split vector $\mathcal{V} = (V_1, V_2, \ldots, V_b)$

Figure: A split tree with $n = 35$, $b = 3$ and $s = 4$. The split vector $\mathcal{V}$ is the most important parameter and its components are probabilities

The binary search tree has $b = 2$, $s = 1$ and $\mathcal{V} = (U, 1 - U)$

Split Trees

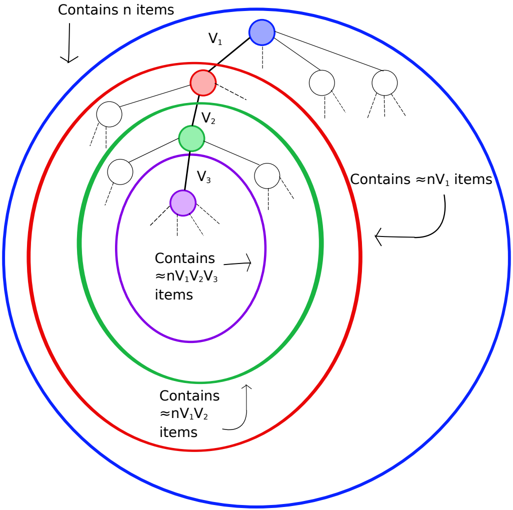


Figure: *Given all split vectors in the tree, n_v for v at depth d is close to $nL_v = n\prod_{j=1}^{d} V_j$, where the V_j's are i.i.d. random variables distributed as V*

Cecilia Holmgren

Split Trees: Most Nodes Close to Depth $c \ln n$

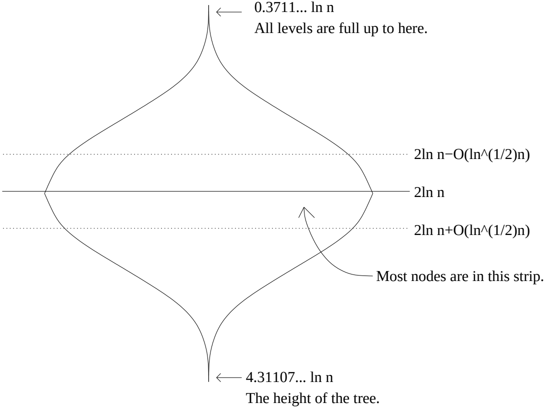


Figure: *The central limit theorem holds for the depths of the nodes. Most nodes are in a strip of width $\mathcal{O}(\sqrt{\ln n})$ around the depth $c \ln n$. The figure shows the distribution of the nodes in a binary search tree*

Galton-Watson-Trees

- A Galton-Watson tree starts with a root (a first ancestor). Children are born accordingly to some given probability distribution
- The children of the root are the first generation. Each of them give birth to new children independently and according to the same probability distribution
- The tree grows in several generations so that new ancestors give birth to children independently of each other and previous generations
- The family dies out if no children survives. If the family survives (the tree becomes infinite) or dies out (the tree becomes finite) depends on the expected number of children

Galton-Watson-Trees

- Let $\mathbf{p_i}$ be the probability that a node has i children, thus $\sum_{i \geq 0} p_i = 1$. We assume that $p_0 > 0$ and $p_0 + p_1 < 1$ to avoid trivial special cases

- Let $\mathbf{Z_n}$ be the number of individuals in generation n. It is clear that if $Z_n = 0$ for some n it holds that $Z_j = 0$ for all $j > n$

- The most important parameter is the parameter of extinction

$$q := P(Z_n = 0 \text{ for some n}) = \lim_{n \to \infty} P(Z_n = 0)$$

Galton-Watson-Trees

Theorem

Let q be the extinction probability and $m := \mathbb{E}(Z_1) = \sum_{i=0}^{\infty} ip_i$. Then it holds that $q = 1$ if $m \leq 1$ and $q < 1$ if $m > 1$.

The tree thus becomes finite with probability 1 in the sub-critical case $m < 1$ and in the critical case $m = 1$. The tree becomes infinite with positive probability in the super-critical case $m > 1$

Conditional Galton-Watson-Trees

- Interestingly, could many random trees that had been studied for a long time individually by combinatorialists be unified by critical Galton-Watson trees ($m = 1$), where one conditions on that the total number of nodes is n Kolchin (1986)
- These are in fact equivalent to ***simply generated trees*** (that is a more common name among combinatorialists) introduced by Meir and Moon (1978)

Cayley Trees

- A **labelled rooted tree**, also called a **Cayley-tree**, is a well-known combinatorial tree. By labelled it means that the nodes are marked,i.e., the order of the children is relevant

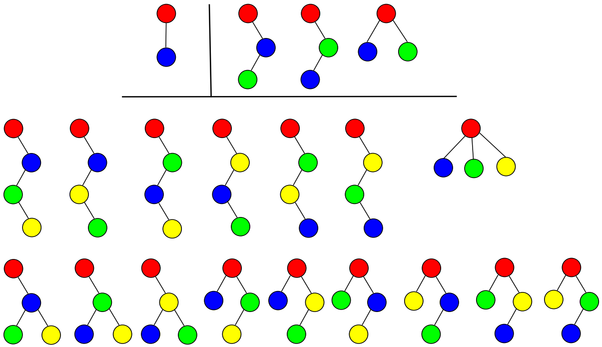


Figure: *Cayley-trees of sizes 2,3 and 4*

Cayley Trees

- Well-known result: There are n^{n-1} rooted Cayley trees with n nodes. Note that if one has an unordered tree with n nodes this number is divided by n
- A randomly chosen Cayley tree is equivalent to a conditional Galton-Watson tree with a $Po(1)$ distribution of the number of children

Depth First-Search Walk

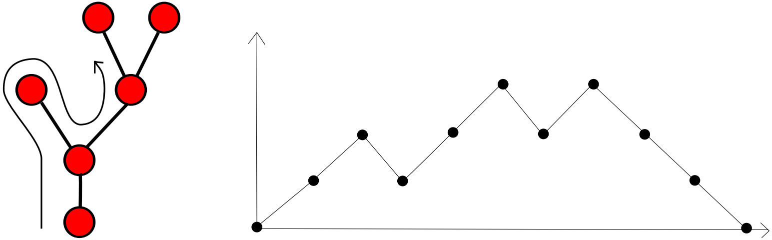


Figure: *The depth first-search walk, which can be regarded as a continuous excursion can be analyzed on any rooted tree, but is in particular useful in studies of conditional Galton-Watson trees*

Aldous Continuum Random Tree, CRT

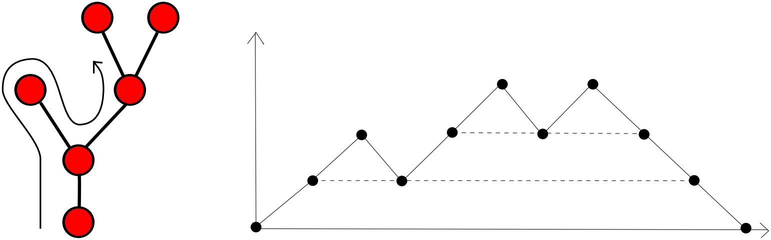


Figure: *The depth-first search (the continuous excursion) can be regarded as a function $X_n(2nt)$ for $0 \leq t \leq 1$. There is an inverse so that one can go from $X_n(2nt)$ to the tree. The tree has $n+1$ nodes.*

Aldous Continuum Random Tree, CRT

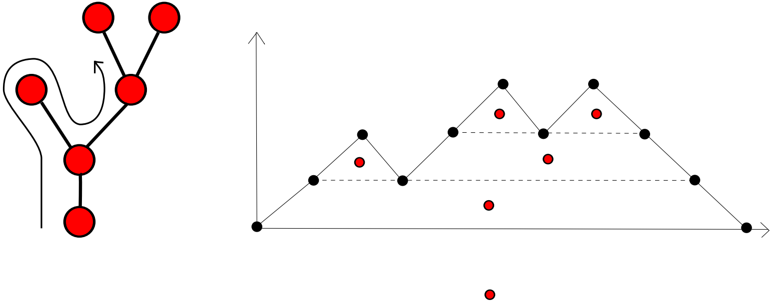


Figure: *The depth-first search (the continuous excursion) can be regarded as a function $X_n(2nt)$ for $0 \leq t \leq 1$. There is an inverse so that one can go from $X_n(2nt)$ to the tree. The tree has $n + 1$ nodes.*

Aldous Continuum Random Tree, CRT

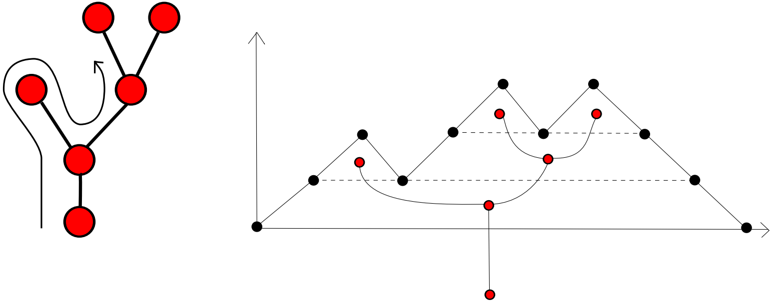


Figure: *The depth-first search (the continuous excursion) can be regarded as a function $X_n(2nt)$ for $0 \leq t \leq 1$. There is an inverse so that one can go from $X_n(2nt)$ to the tree. The tree has $n + 1$ nodes.*

Aldous Continuum Random Tree, CRT

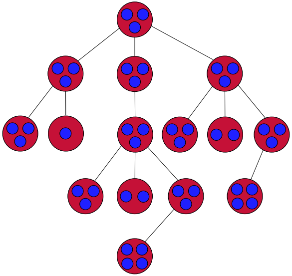
Theorem

Suppose that the conditional Galton-Watson tree $\mathcal{T}_n$ is critical ($m = 1$) with finite variance for the number of children. Then it holds that the depth first-search walk (the continuous excursion) $\frac{X_n(2nt)}{\sqrt{n}}$ converges in distribution to a Brownian excursion $e(t)$ for $0 \leq t \leq 1$.

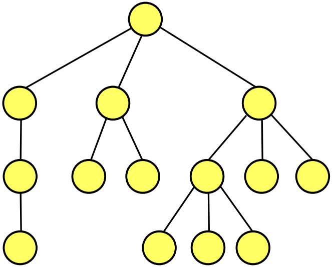
- (A Brownian excursion is a stochastic process which has important applications in e.g., physics and financial mathematics)
- Hence, $\frac{\mathcal{T}_n}{\sqrt{n}}$ converges in distribution to Aldous continuum random tree, CRT. The tree is scaled by $\sqrt{n}$ since the height of a conditional Galton-Watson-tree (and thus the height of $X_n(2nt)$) is of order $\sqrt{n}$

UPPSALA UNIVERSITET

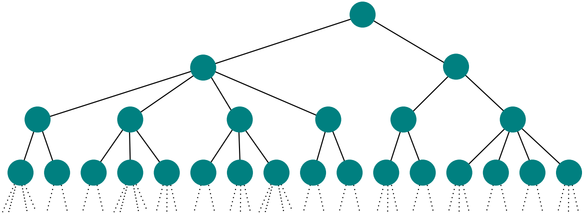
Some of My Own Results



Split trees



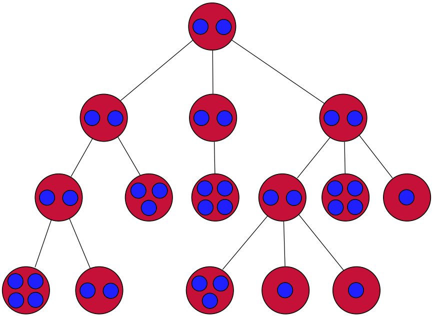
Conditional Galton-Watson trees



Infinite Galton-Watson trees

Some of My Own Results:

Renewal Theory to Study Split Trees



Renewal Theory

Study sums $S_k = \sum_i^k X_i$ of $i.i.d$ random variables

- A light bulb has a random life time X_1 with some distribution and when it breaks it has to be replaced by a new light bulb with a life time X_2 of the same distribution
- How many light bulbs are needed say in 1 year time?
- This number is a random variable called **counting process** $\mathcal{N}(t) := \max\{k : S_k \leq t\}$ in renewal theory
- Let $F(t) = \mathbf{P}(X_1 \leq t)$. The **renewal function** is defined as $V(t) := \mathbf{E}(\mathcal{N}(t)$ and satisfies the **renewal equation**

$$\begin{aligned} V(t) &= \sum_{k=1}^{\infty} \mathbf{P}(S_k \leq t) = F(t) + \int_0^t V(t-s)dF(s) \\ &= F(t) + (V * dF)(t) \end{aligned}$$

- Law of large numbers suggests $V(t) = \frac{t}{\mathbf{E}(X)}$+o(t)

UPPSALA UNIVERSITET

Applying Renewal Theory

- Recall that the subtree sizes n_v for v at depth d are approximated by $n\prod_{j=1}^{d} V_j$
- Let $S_d := -\sum_{j=1}^{d} \ln V_j$. Note that $n\prod_{j=1}^{d} V_j = ne^{-S_d}$
- Define the renewal function

$$U(t) := \sum_{k=1}^{\infty} b^k \mathbf{P}(S_k \leq t),$$

  and let $F(t) := b\mathbf{P}(-\ln V \leq t)$
- For $U(t)$ we obtain the following renewal equation

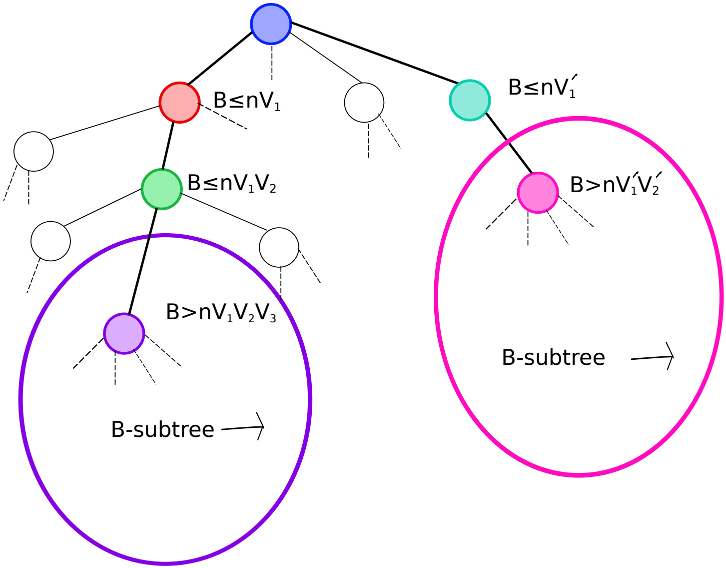
$$U(t) = F(t) + (U * dF)(t)$$

- As $t \to \infty$, $U(t)$ satisfies

$$U(t) = (c + o(1))e^t$$

  for some constant c

B-Subtrees



Cecilia Holmgren

UPPSALA UNIVERSITET

The Random Number of Nodes in a Split Tree

Theorem

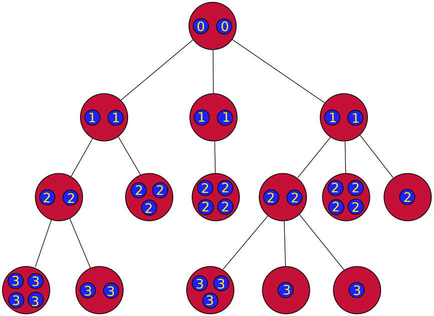
Let N be the random number of nodes in a split tree with n items. Then it holds that there is a constant C depending on the type of the split tree such that

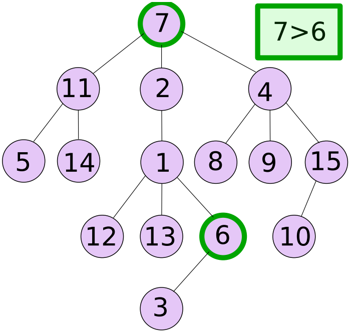
$$\mathbf{E}[N] = Cn + o(n) \text{ and } \mathrm{Var}(N) = o(n^2)$$

C. Holmgren, *Electronic Journal of Probability* (2012)

Some of My Own Results:

Total Path Length and Number of Inversions



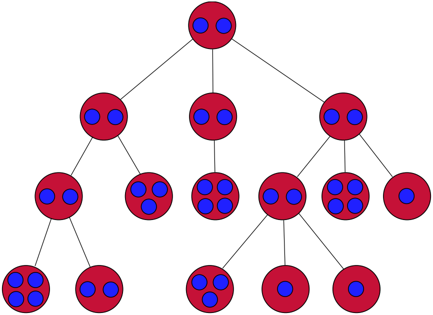


The Total Path Length/Running-Time of Sorting Algorithms

- Sorting algorithms sort a collection of data items (often called keys) by comparisons of the input data
- The number of comparisons for a certain key is given by its depth in the tree
- **The total number of comparisons is the total path length, which therefore represents a natural cost measure or running time of these algorithms**
- Effective sorting algorithms are represented by $\log n$-trees with total path length, i.e., running time $\mathcal{O}(n \log n)$

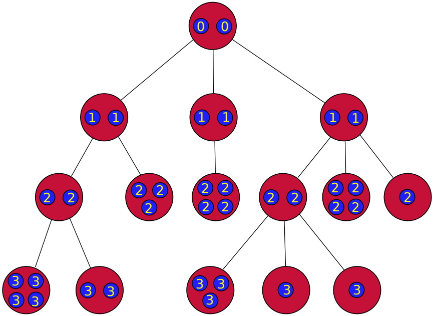
The Total Path Length of A Rooted Tree

- The total path length is the sum of the depths of all items (often represented by keys in tree data structures) in the tree

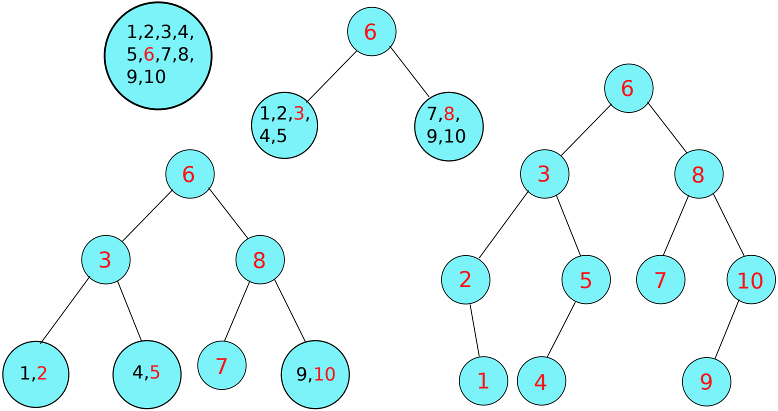


The Total Path Length of A Rooted Tree

- The total path length is the sum of the depths of all items (often represented by keys in tree data structures) in the tree

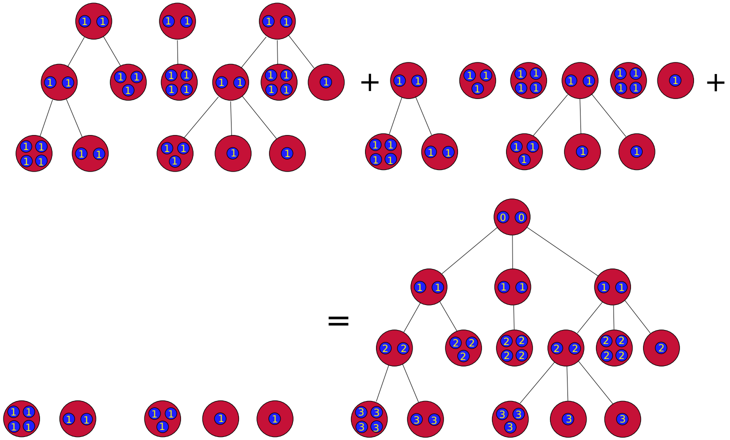


The Running Time of Quicksort



An Equivalent Definition of the Total Path Length

- The total path length can also be defined as the sum of all proper subtree sizes, i.e., $\Psi(T^n) = \sum_{v \neq \sigma} n_v$



The Total Path Length of Split Trees

Theorem

Let $\Psi(T^n)$ be the total path length in a split tree with split vector $\mathcal{V} = (V_1, \ldots, V_b)$. Then $\mathbf{E}\left[\Psi(T^n)\right] = Cn \ln n + n\omega(\ln n) + o(n)$, where C is a constant (depending on $\mathcal{V}$) and ω is a continuous periodic function. Let $X_n := \frac{\Psi(T^n) - \mathbf{E}[\Psi(T^n)]}{n}$. Then $X_n \to X$ in distribution, where X is the unique solution of the fixed point equation

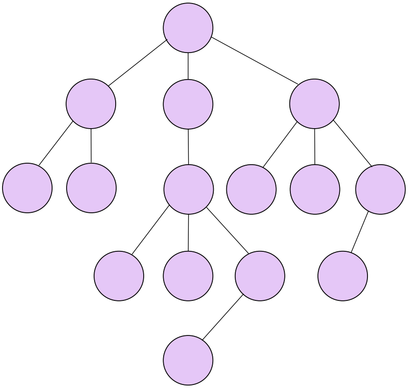
$$X \stackrel{d}{=} \sum_{k=1}^{b} V_k X^{(k)} + C(\mathcal{V}),$$

where $X^{(k)}$ are independent and identically distributed copies of X, satisfying $\mathbf{E}[X] = 0$ and $\mathrm{Var}(X) < \infty$ and $C(\mathcal{V})$ is a function of $\mathcal{V}$.

N. Broutin and **C. Holmgren**, *Annals of Applied Probability* (2012)

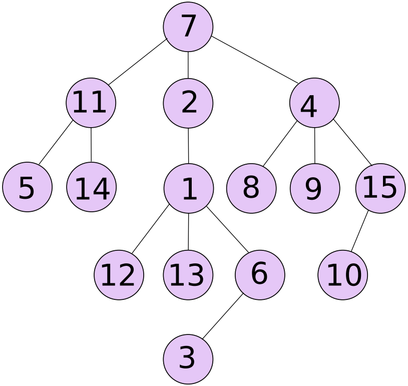
What is an Inversion of a Rooted Tree?

- Write $u < v$ if u is an ancestor of v. Given a bijection $\lambda : V \to \{1, \ldots, |V|\}$ (a **node labelling**), define an **inversion** as a pair (u, v) so that $\lambda(u) > \lambda(v)$



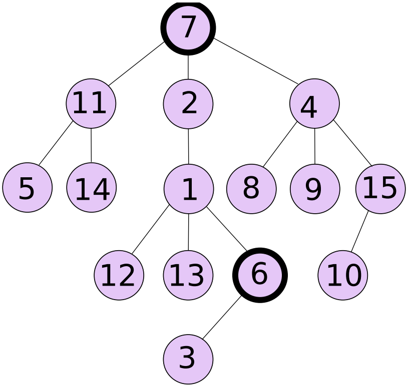
What is an Inversion of a Rooted Tree?

- Write $u < v$ if u is an ancestor of v. Given a bijection $\lambda : V \to \{1, \ldots, |V|\}$ (a **node labelling**), define an **inversion** as a pair (u, v) so that $\lambda(u) > \lambda(v)$



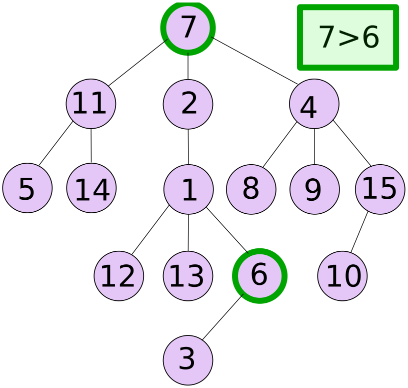
What is an Inversion of a Rooted Tree?

- Write $u < v$ if u is an ancestor of v. Given a bijection $\lambda : V \to \{1, \ldots, |V|\}$ (a **node labelling**), define an **inversion** as a pair (u, v) so that $\lambda(u) > \lambda(v)$



What is an Inversion of a Rooted Tree?

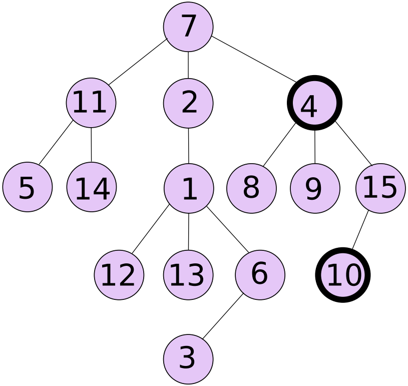
- Write $u < v$ if u is an ancestor of v. Given a bijection $\lambda : V \to \{1, \dots, |V|\}$ (a **node labelling**), define an **inversion** as a pair (u, v) so that $\lambda(u) > \lambda(v)$



Cecilia Holmgren

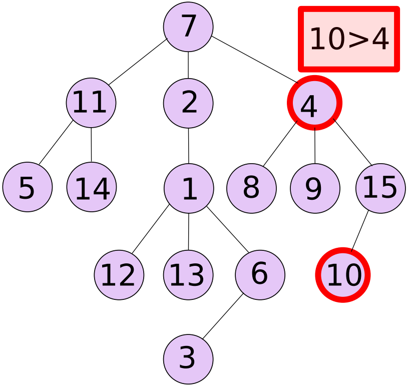
What is an Inversion of a Rooted Tree?

- Write $u < v$ if u is an ancestor of v. Given a bijection $\lambda : V \to \{1, \ldots, |V|\}$ (a **node labelling**), define an **inversion** as a pair (u, v) so that $\lambda(u) > \lambda(v)$



What is an Inversion of a Rooted Tree?

- Write $u < v$ if u is an ancestor of v. Given a bijection $\lambda : V \to \{1, \ldots, |V|\}$ (a **node labelling**), define an **inversion** as a pair (u, v) so that $\lambda(u) > \lambda(v)$



The Number of Inversions of Split Trees

Let $I(T_n)$ be the number of inversions in a split tree. Let $\mathcal{V} = (V_1, \dots, V_b)$ be the split vector and assume that $-\ln V_i$ is non-lattice. Then $\mathbf{E}[I(T_n)] = \frac{1}{2}\mathbf{E}[\Psi(T^n)] = \frac{C_1}{2} n \ln n + \frac{C_2}{2} n + o(n)$.

Theorem

Let $X_n = \frac{I(T_n) - \mathbf{E}[I(T_n)]}{n}$. Then $X_n \to X$ in distribution, where X is the unique solution of the fixed point equation

$$X \overset{d}{=} \alpha U_0 + \sum_{i=1}^{b} V_i X^{(i)} + C(\mathcal{V})$$

where $U_0 \sim \mathrm{Unif}(0,1)$, each $X^{(i)}$ is an independent copy of X and $C(\mathcal{V})$ is a function of the split vector $\mathcal{V}$. The variables are independent except for the V_i's.

X. S. Cai, **C. Holmgren**, T. Johansson, S. Janson & F. Skerman *arXiv:1709.00216 (2017)*

The Number of Inversions of Conditional Galton-Watson Trees

Let $e(s), s \in [0,1]$ be the random path of a standard Brownian excursion, and define $C(s,t) = 2\min_{s \le u \le t} e(u)$. We define a random variable $\eta = \int_{[0,1]^2} C(s,t) ds\, dt$

Theorem

Suppose T_n is a conditional Galton–Watson tree with offspring distribution ξ such that $\mathbf{E}[\xi] = 1$, $\mathrm{Var}(\xi) = \sigma^2 \in (0,\infty)$, and $\mathbf{E}\left[e^{\alpha\xi}\right] < \infty$ for some $\alpha > 0$, and define $Y_n = \frac{I(T_n) - \frac{1}{2}\Upsilon(T_n)}{n^{5/4}}$. Then,

$$Y_n \xrightarrow{d} Y \stackrel{\text{def}}{=} \frac{1}{\sqrt{12\sigma}}\sqrt{\eta}\,\mathcal{N},$$

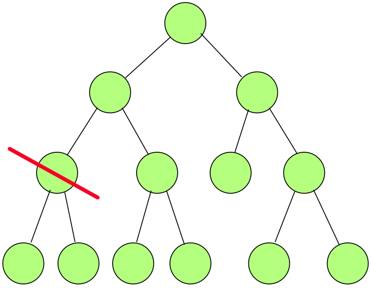
where $\mathcal{N}$ is a standard normal random variable, independent from the random variable η.

X. S. Cai, **C. Holmgren**, T. Johansson, S. Janson & F. Skerman *arXiv:1709.00216*

UPPSALA UNIVERSITET

Some of My Own Results:

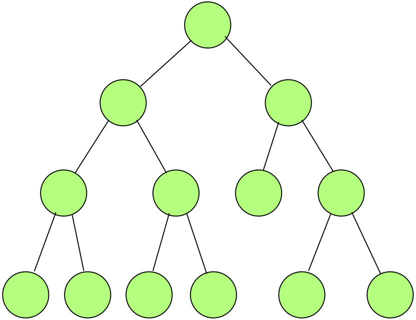
Cutting Down Split Trees and Conditonal Galton-Watson Trees



Cecilia Holmgren

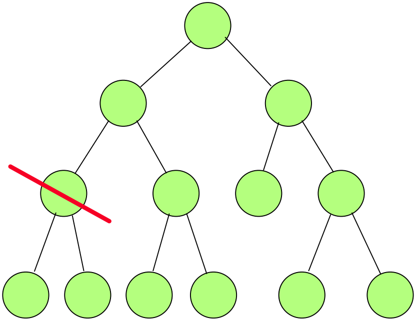
What is a Cutting in a Rooted Tree?

- Choose a random node



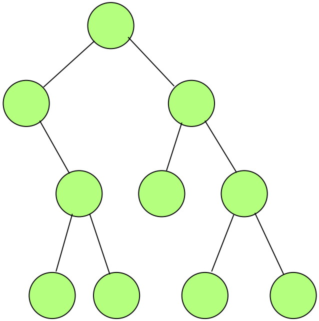
What is a Cutting in a Rooted Tree?

- Choose a random node
- Cut in this node so that the tree separates into two parts and keep only the part which contains the root



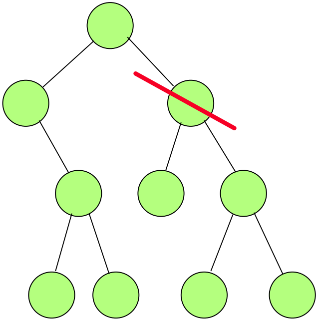
What is a Cutting in a Rooted Tree?

- Choose a random node
- Cut in this node so that the tree separates into two parts and keep only the part which contains the root



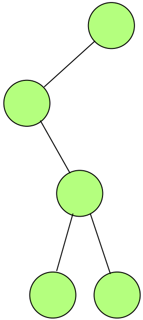
What is a Cutting in a Rooted Tree?

- Choose a random node
- Cut in this node so that the tree separates into two parts and keep only the part which contains the root



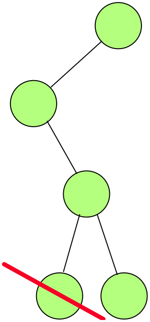
What is a Cutting in a Rooted Tree?

- Choose a random node
- Cut in this node so that the tree separates into two parts and keep only the part which contains the root



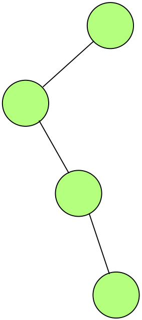
What is a Cutting in a Rooted Tree?

- Choose a random node
- Cut in this node so that the tree separates into two parts and keep only the part which contains the root



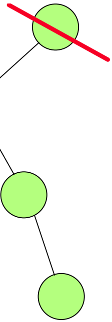
What is a Cutting in a Rooted Tree?

- Choose a random node
- Cut in this node so that the tree separates into two parts and keep only the part which contains the root



What is a Cutting in a Rooted Tree?

- Choose a random node
- Cut in this node so that the tree separates into two parts and keep only the part which contains the root
- What is the maximum number of cuts required to cut the root?



The Number of Cuts to Cut Down a Split Tree

Theorem

Let $X(T_n)$ be the random number of cuts that is required to cut down a split tree with n items. Then it holds that $n \to \infty$,

$$\left(X(T^n) - f(n)\right) \Big/ \frac{c_1 n}{c^2 \log^2 n} \xrightarrow{d} -W, \quad \text{where}$$

$$f(n) := \frac{c_1 n}{c \log n} + \frac{c_1 n \log\log n}{c \log^2 n} - \frac{c_2 n}{c \log^2 n}$$

for constants c, c_1 and c_2 depending on the type of the split tree, where W has a weakly 1-stable distribution.

C. Holmgren, *Combinatorics Probability and Computing* (2010) (for the specific case of the binary search tree)
C. Holmgren, *Advances in Applied Probability* (2011) (for all split trees)

UPPSALA UNIVERSITET

The Number of Cuts to Cut Down a Conditional Galton-Watson Tree

Theorem

Let $X(\mathcal{T}^n)$ be the random number of cuts that is required to cut down a (critical) conditional Galton-Watson tree $\mathcal{T}^n$.

$$\lim_{n\to\infty} P\left(\frac{X(\mathcal{T}^n)}{\sigma\sqrt{n}} \geq x\right) = e^{-x^2/2},$$

with the constant $\sigma^2 = Var(\xi)$ where ξ is the offspring distribution. .

L. Addario-Berry, N. Broutin and **C. Holmgren**, *Annals of Applied Probability* (2014)

Number of Cuts to Cut Down a Cayley Tree

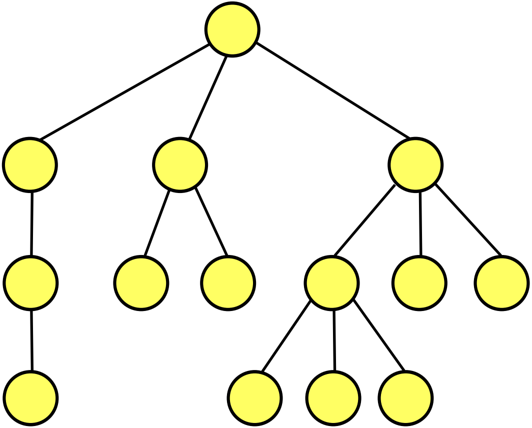


Figure: *A Cayley tree.*

Number of Cuts to Cut Down a Cayley Tree

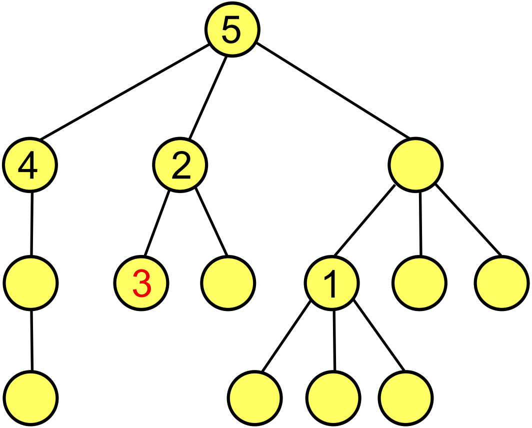


Figure: *We choose nodes to cut in randomly until the root is cut*

Number of Cuts to Cut Down a Cayley Tree

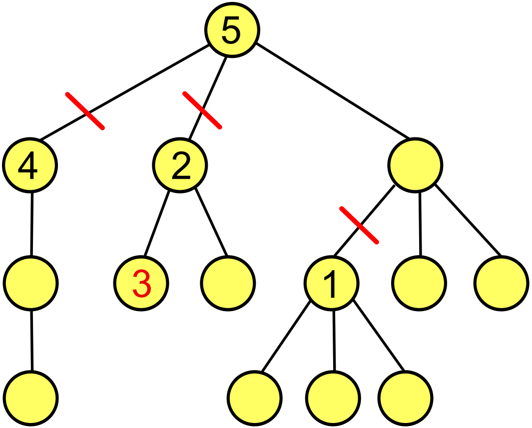


Figure: *We cut in a node by deleting the edge over that node. It is not allowed to continue to cut in subtrees that we have already cut*

Number of Cuts to Cut Down a Cayley Tree

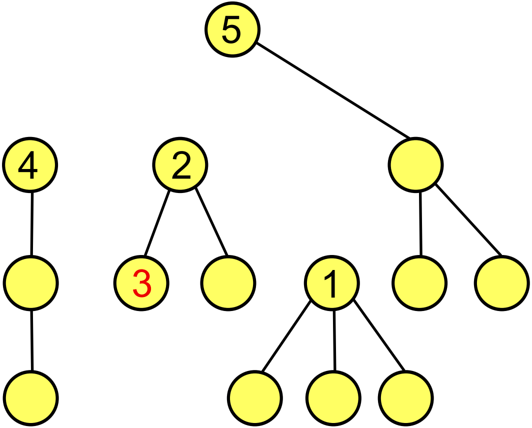


Figure: *We have cut down all these four subtrees, when the root of the first tree, which is the node labelled 5, finally is cut*

Number of Cuts to Cut Down a Cayley Tree

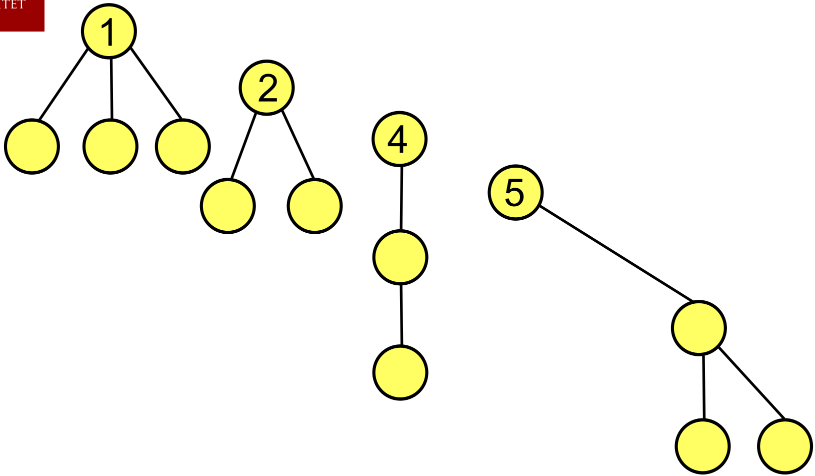


Figure: *The subtrees that are cut down are placed in a path after each other, where the first cut node becomes the new root of a tree. A coupling argument shows that the new tree is also a Cayley tree*

Number of Cuts to Cut Down a Cayley Tree

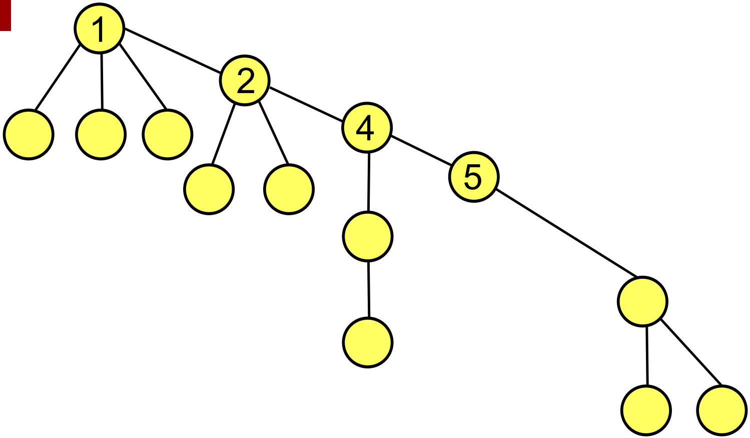


Figure: *Thus, the new tree is distributed as a Cayley-tree and the path between the node labelled 1 and the node labelled 5 represents a path between the root and a random node in the tree. This distance scaled by* $\sqrt{n}$ *has a Rayleigh distribution*

The Number of Cuts to Cut Down a Conditional Galton-Watson Tree

- By using a coupling argument we have shown that the number of cuts to cut down a Cayley-tree is distributed as the distance between the root of the Cayley tree and a random node in the tree. A well-known result is that this distance scaled by $\sqrt{n}$ is Rayleigh distributed

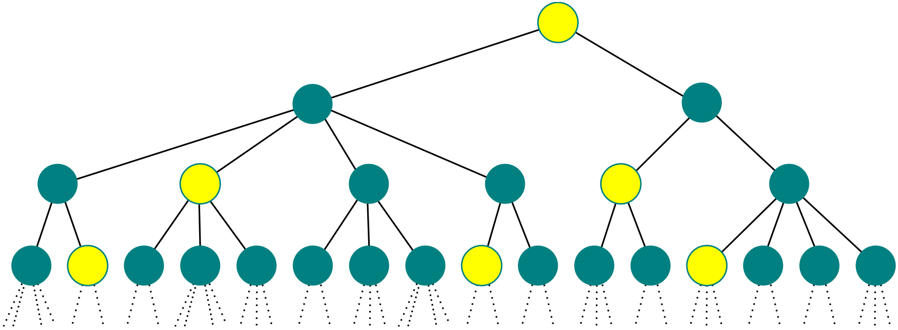
- We can then show the same result for **general conditional Galton-Watson trees** $\mathcal{T}^n$ by using the **Aldous continuum random tree CRT,** which is the "limiting tree" of all conditional Galton-Watson-trees (scaled by $\sqrt{n}$)

L. Addario-Berry, N. Broutin and **C. Holmgren**, *Annals of Applied Probability* (2014)

UPPSALA UNIVERSITET

Some of My Own Results:

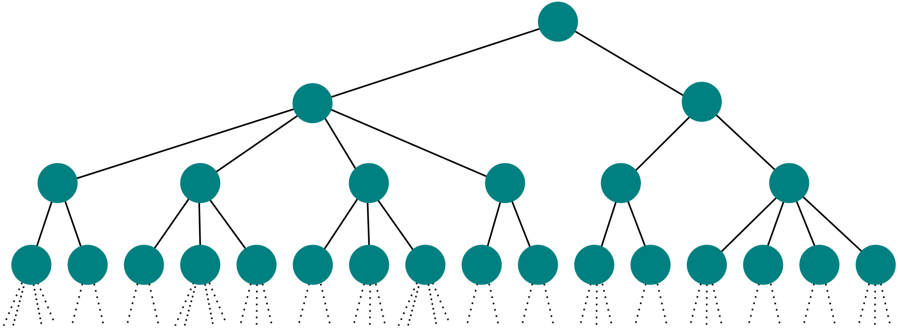
Bootstrap Percolation on Galton-Watson Trees



Cecilia Holmgren

Bootstrap Percolation on Galton-Watson Trees

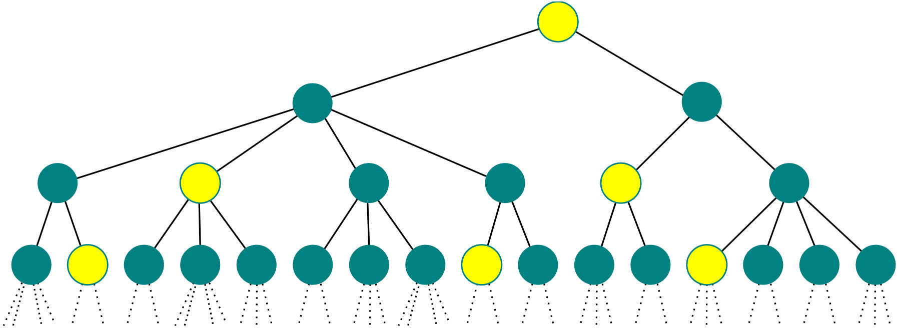
(1) Start with an arbitrary infinite Galton-Watson tree



Cecilia Holmgren

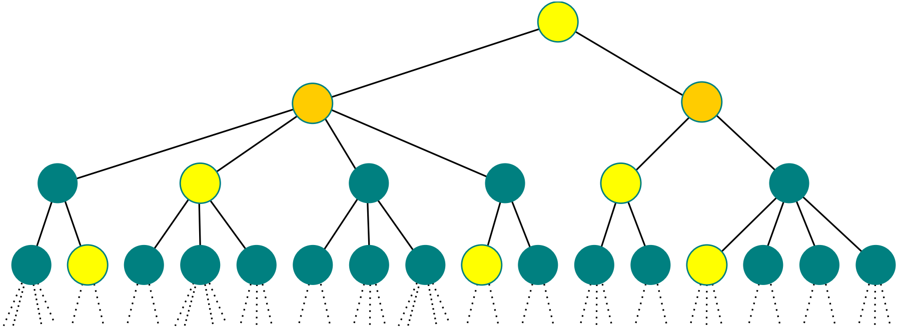
Bootstrap Percolation on Galton-Watson Trees

(2) Let the nodes be infected with probability *p*



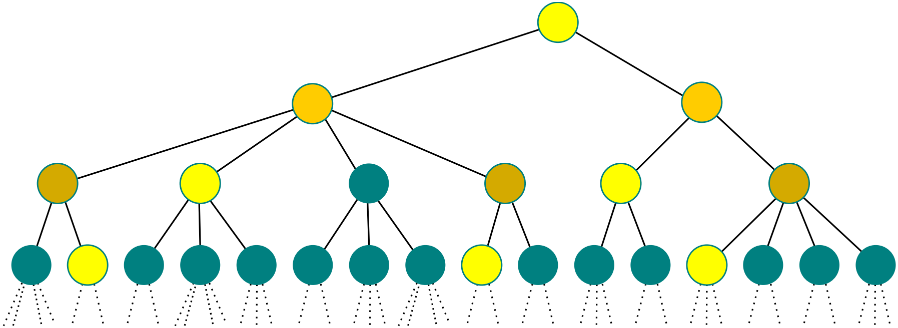
Bootstrap Percolation on Galton-Watson Trees

(3) Nodes with 2 infected neighbors get infected



Bootstrap Percolation on Galton-Watson Trees

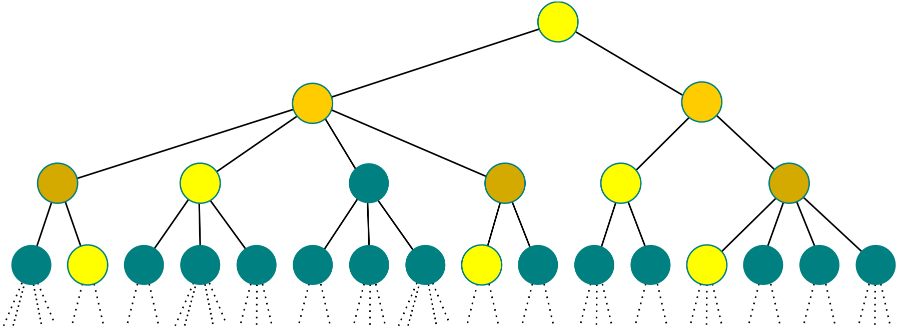
(4) Infection continues until no more nodes get infected



Bootstrap Percolation on Galton-Watson Trees

(4) Infection continues until no more nodes get infected

Aim: **To find the threshold for the infection probability *p* that determines when the final infected cluster occupies the whole tree (i.e., the tree percolates)**



Critical Probabilities for Galton-Watson Trees

Let T_ξ be the Galton-Watson tree with offspring distribution ξ.
Let the ***critical probability*** be

$$\mathbf{p_C}(\mathbf{T}_\xi) := \inf\{\mathbf{p} : \mathbf{P}(\mathbf{T}_\xi \textbf{ percolates}) > \mathbf{0}\}.$$

Assume that $P(\xi = 0) = 0$ and define

$$\mathbf{f}(\mathbf{b}) := \inf\{\mathbf{p_C}(\mathbf{T}_\xi) \mid \mathbf{E}(\xi) = \mathbf{b}\}$$

Theorem

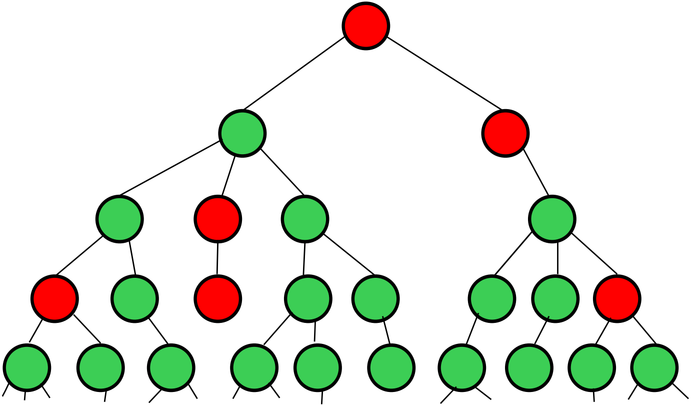
1. If $b = 1$ then $\mathbf{f}(\mathbf{1}) = \mathbf{1}$.
2. There are constants c and C such that if $\mathbf{b} \geq \mathbf{2}$ then

$$\frac{\mathbf{c}}{\mathbf{b}}\mathbf{e}^{-\mathbf{b}} \leq \mathbf{f}(\mathbf{b}) \leq \mathbf{C}\mathbf{e}^{-\mathbf{b}}.$$

B. Bollobás, K. Gunderson, **C. Holmgren**, S. Janson and M. Przykucki
Electronic Journal of Probability (2014)

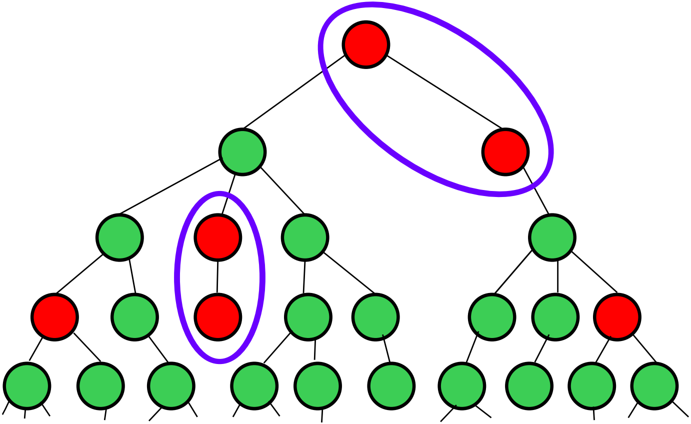
UPPSALA UNIVERSITET

Healthy 1-Forts Stops Infection



Cecilia Holmgren

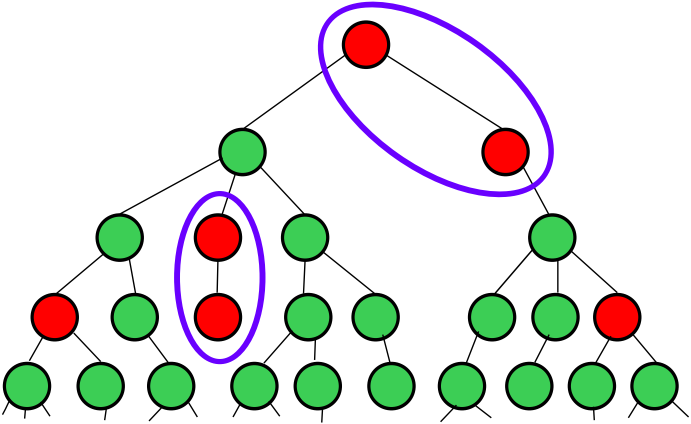
Healthy 1-Forts Stops Infection



Cecilia Holmgren

Healthy 1-Forts Stops Infection

By using a branching process to analyze a fixed point-equation we could determine the probability that the root stays healthy forever, i.e., that it belongs to a healthy 1-fort



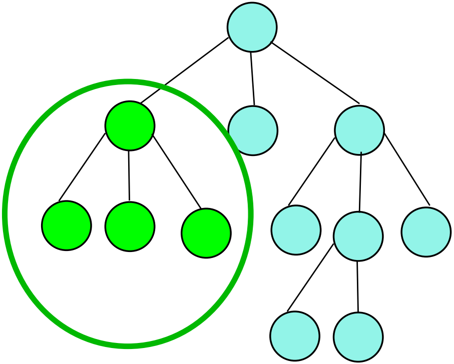
Bootstrap Percolation on Galton-Watson Trees

- ***For Galton-Watson trees with offspring distribution*** ξ ***we have shown sharp bounds for the function*** $\mathbf{f}(\mathbf{b})$, which is the infimum of the critical probabilities over all Galton-Watson trees with $E(\xi) = b$. (This function is never 0, but of order e^{-b} and thus is much smaller than the critical probability for the complete b-ary tree which is $\frac{1}{2b^2}$)
- For any offspring distribution ξ, we have also shown ***upper and lower bounds of the critical probabilities*** $\mathbf{p_C}(\mathbf{T}_\xi)$. These bounds explain that offspring distributions highly concentrated around their mean b yield much higher values for the critical probability than offsprings with larger variations from the mean b. (The complete b-ary tree is a special case of a Galton-Watson tree, and has a large critical value)

B. Bollobás, K. Gunderson, **C. Holmgren**, S. Janson and M. Przykucki
Electronic Journal of Probability (2014)

Some of My Own Results:

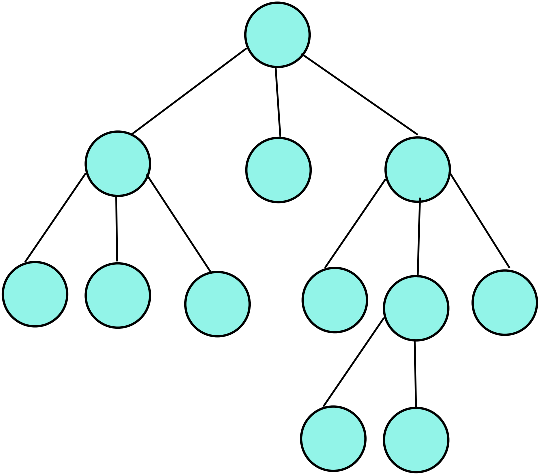
Fringe Subtrees to Study Random Trees



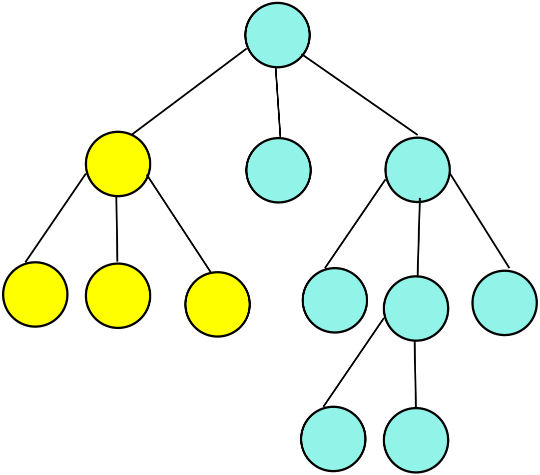
Cecilia Holmgren

UPPSALA UNIVERSITET

What is a Fringe Subtree?

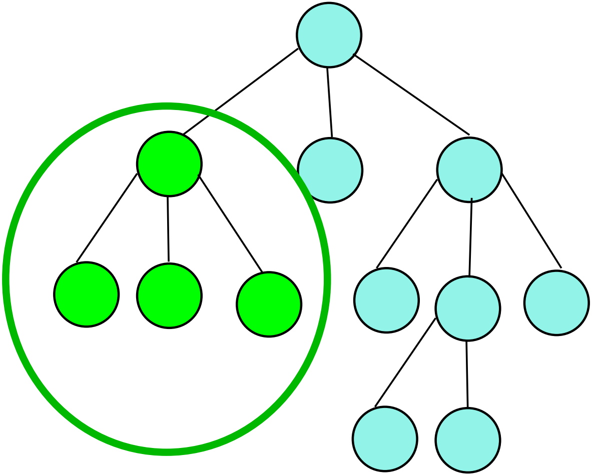


What is a Fringe Subtree?



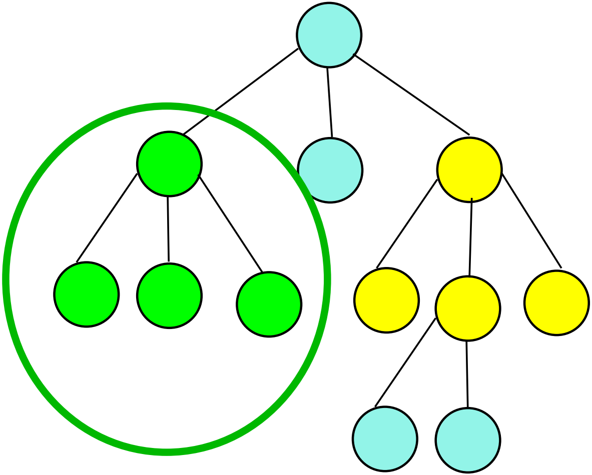
Cecilia Holmgren

What is a Fringe Subtree?



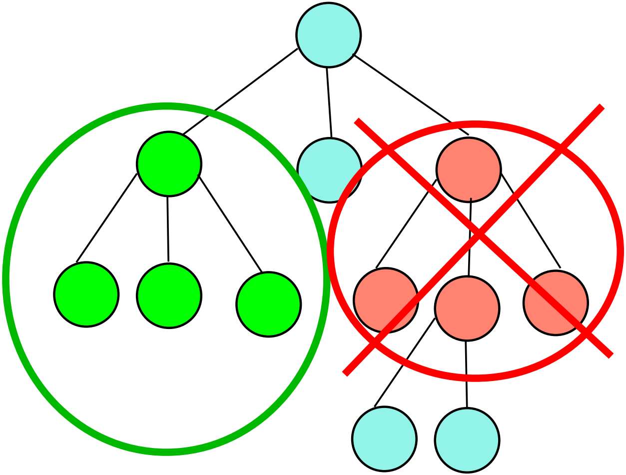
UPPSALA UNIVERSITET

What is a Fringe Subtree?



Cecilia Holmgren

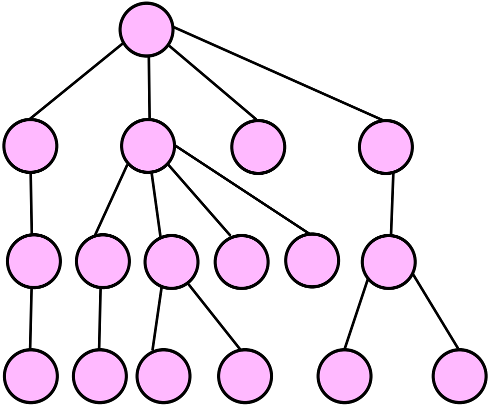
What is a Fringe Subtree?



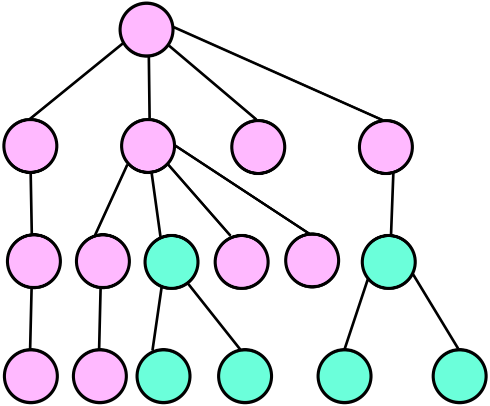
Cecilia Holmgren

UPPSALA UNIVERSITET

Counting Fringe Subtrees



Counting Fringe Subtrees



UPPSALA UNIVERSITET

Functions of Subtrees

Let T be a tree and $f(T)$ be a function to $\mathbb{R}$. In a random tree T_n. Set

$$\mathbf{X_n} = \sum_{\mathbf{u}} \mathbf{f(T_n(u))},$$

summing over all nodes in T_n, where $T_n(u)$ is the subtree rooted at u. We can thus use X_n to calculate the number of subtrees with properties that interest us by choosing appropriate functions f

Examples:

- Let $f(T_n(u)) = 1\{T_n(u) \approx T\}$. Then X_n is ***the number of subtrees that are equal to* T**
- Let $f(T_n(u)) = 1\{| T_n(u) |= k\}$. Then X_n is ***the number of subtrees with exactly* k *nodes***
- Let $f(T_n(u)) = 1\{| T_n(u) |= 1\}$. Then X_n is ***the number of leaves***

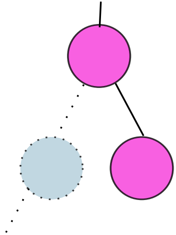
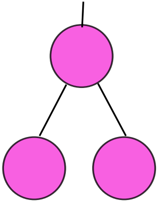
Fringe Subtrees in some Split Trees

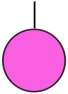
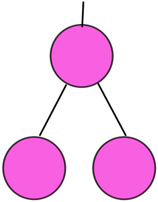
- By using Stein's method we could prove general limit theorems for sums of functions of fringe subtrees in the case of the binary search tree and the random recursive tree. When the fringe subtrees are not too large we get normal limit laws and when they are large (tending to infinity) we get Poisson limit laws
  C. Holmgren and S. Janson, Electr. Journ. Probab. 2015a

- We could then use Pólya urns to show normal limit laws of fringe subtrees in the more general classes of *m*-ary search trees (including the binary search tree) and preferential attachment trees (including the random recursive tree). These results could for example be applied to show a central limit theorem for the number of "2-protected" nodes in *m*-ary search trees
  C. Holmgren and S. Janson, Electr. Journ. Probab. 2015b;
  C. Holmgren, S. Janson and M. Sileikis Electr. Journ. Combin. 2017

Example: Protected Nodes in the Binary Search Tree

We consider the number of so-called *2-protected nodes* in binary search trees. A node is 2-protected if the shortest distance to a leaf is at least two, i.e., it is neither a leaf or the parent of a leaf

$n-X_n=2\#$  $+3\#$ 

$=2\#$  $-\#$ 

Protected Nodes in the Binary Search Tree

The following theorem was shown by Mahmoud and Ward using generating functions and the contraction method.

Theorem

Let X_n be the number of protected nodes in a binary search tree. Then

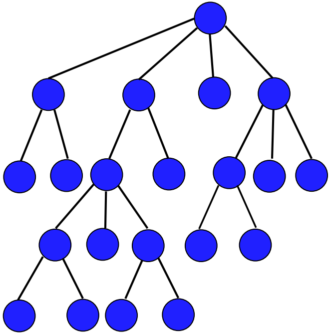
$$\frac{X_n - \frac{11}{30}n}{\sqrt{n}} \xrightarrow{d} \mathcal{N}\left(0, \frac{29}{225}\right).$$

- We provide a simple proof of this theorem using that the number of unprotected nodes equals twice the number of leaves minus the number of cherry subtrees

  C. Holmgren and S. Janson, Electr. Journ. Probab. 2015a

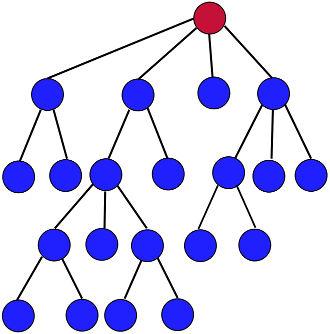
Embedded Heavy Trees

- The embedded 2-heavy tree in a conditional Galton-Watson tree



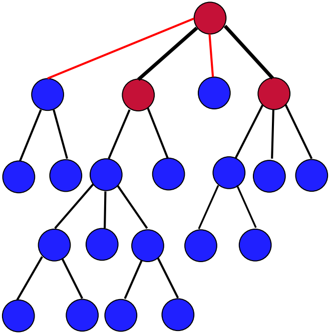
Embedded Heavy Trees

- The embedded 2-heavy tree in a conditional Galton-Watson tree



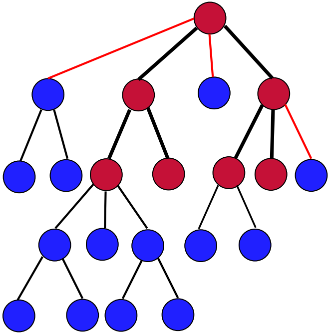
Embedded Heavy Trees

- The embedded 2-heavy tree in a conditional Galton-Watson tree



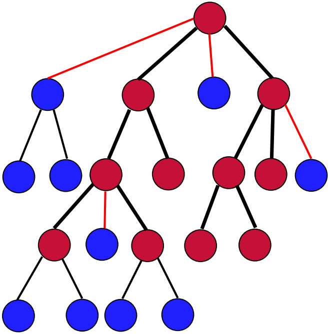
Embedded Heavy Trees

- The embedded 2-heavy tree in a conditional Galton-Watson tree



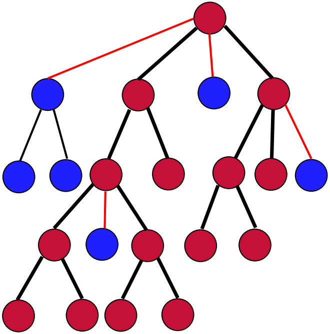
Embedded Heavy Trees

- The embedded 2-heavy tree in a conditional Galton-Watson tree



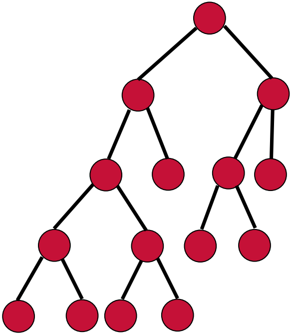
Embedded Heavy Trees

- The embedded 2-heavy tree in a conditional Galton-Watson tree



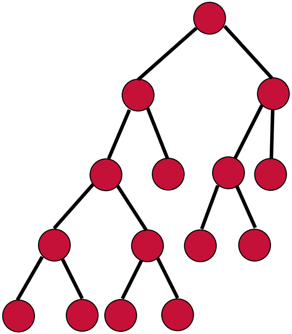
Embedded Heavy Trees

- The embedded 2-heavy tree in a conditional Galton-Watson tree



Embedded Heavy Trees

- The embedded 2-heavy tree in a conditional Galton-Watson tree
- We have applied fringe subtrees to analyze **embedded heavy subtrees** and in the special case of the ”heavy path” we instead used the Aldous continuum random tree



L. Devroye, **C. Holmgren** and H. Sulzbach, 2017 arXiv: 1701.02527

Summary

- We have studied two classes of trees: **Splittrees** (many of these are used as **sorting algorithms** e.g., the binary search tree/Quicksort and **Galton-Watson-trees** (that describe family trees). **Conditional Galton-Watson-trees** contain many combinatorial trees e.g., the Cayley-tree
- By the introduction of split trees that are logarithmic (the height is $c \cdot logn$) and conditional Galton-Watson-trees (height is $C \cdot \sqrt{n}$) one can show general results for all trees at once
- My own results use **renewal theory** as a general method to analyze spit trees. For conditional Galton-Watson trees I have used **Aldous continuum random tree CRT**
- Using a **branching process** to study a **fixed-point equation** we found sharp results for bootstrap percolation in all infinite Galton-Watson trees

UPPSALA UNIVERSITET

Summary

- We have also shown general **normal and Poisson limit laws** for **fringe subtrees** in some split trees that could be directly applied for solving many other problems such as for example the number of protected nodes in such trees
- We have also applied fringe subtrees and the Aldous continuum random tree to analyze **embedded heavy subtrees** in conditional Galton-Watson trees

UPPSALA UNIVERSITET

Xing Shi Cai
Postdoc
(PhD McGill, CAN)

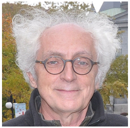
CH

Colin Desmarais
PhD student
(M.Sc Manitoba, CAN)

Tony Johansson
Postdoc
(PhD Carnegie Mellon,USA)

Fiona Skerman
Postdoc
(PhD Oxford, UK)

Svante Janson
Uppsala Univ. SWE

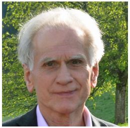


Luc Devroye
McGill Univ. CAN

Nicolas Broutin
INRIA, Paris FR

Louigi Addario-Berry
McGill Univ. CAN

Henning Sulzbach
Univ. of Birmingham UK



Béla Bollobás
Univ. of Cambridge UK
& Univ. of Memphis USA

Karen Gunderson
Univ. of Manitoba CAN

Stephan Wagner
Stellenbosch Univ.
S.Africa

Cecilia Holmgren