On Reductions of Plane Trees

Benjamin Hackl

joint work with
Clemens Heuberger, *Sara Kropf*, *Helmut Prodinger*

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT | WIEN GRAZ

April 7, 2017

mso DK modeling simulation optimization

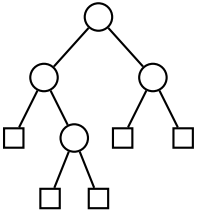
KARL POPPER KOLLEG

FWF Der Wissenschaftsfonds.

Trimming binary trees

Binary trees can be "trimmed" by the following strategy:

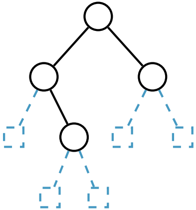
- Remove all leaves
- Merge nodes with only one descendant



Trimming binary trees

Binary trees can be “trimmed” by the following strategy:

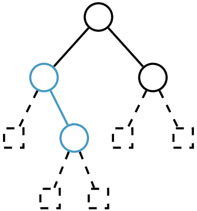
- Remove all leaves
- Merge nodes with only one descendant



Trimming binary trees

Binary trees can be "trimmed" by the following strategy:

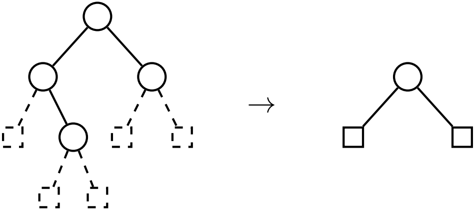
- Remove all leaves
- Merge nodes with only one descendant



Trimming binary trees

Binary trees can be “trimmed” by the following strategy:

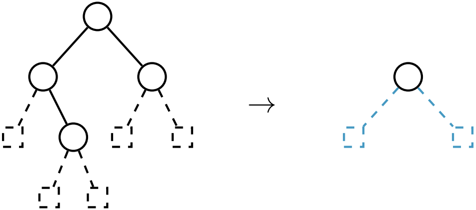
- Remove all leaves
- Merge nodes with only one descendant



Trimming binary trees

Binary trees can be “trimmed” by the following strategy:

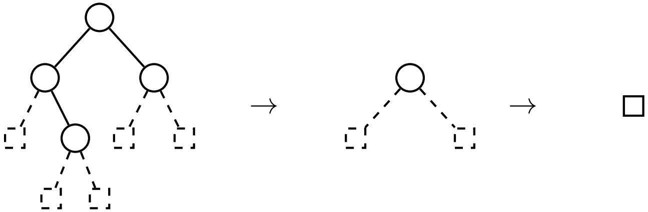
- Remove all leaves
- Merge nodes with only one descendant



Trimming binary trees

Binary trees can be "trimmed" by the following strategy:

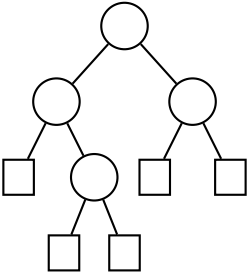
- Remove all leaves
- Merge nodes with only one descendant



"Surviving" nodes

Label all nodes in the tree by the following rules:

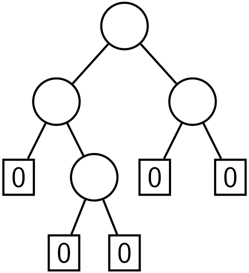
- Leaves $\rightarrow$ 0 (they do not survive a single reduction)
- val(left child) = val(right child) $\rightarrow$ increase by 1
- Otherwise: take the maximum



"Surviving" nodes

Label all nodes in the tree by the following rules:

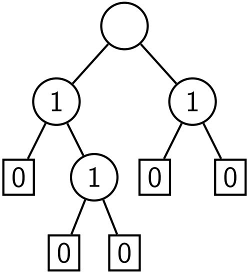
- Leaves → 0 (they do not survive a single reduction)
- val(left child) = val(right child) → increase by 1
- Otherwise: take the maximum



"Surviving" nodes

Label all nodes in the tree by the following rules:

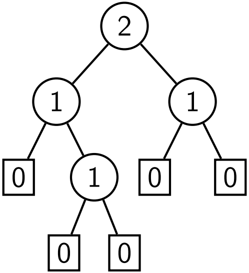
- Leaves $\to$ 0 (they do not survive a single reduction)
- val(left child) = val(right child) $\to$ increase by 1
- Otherwise: take the maximum



"Surviving" nodes

Label all nodes in the tree by the following rules:

- Leaves → 0 (they do not survive a single reduction)
- val(left child) = val(right child) → increase by 1
- Otherwise: take the maximum



The register function

Number in the root of the tree: *Register function*, a.k.a. *Horton–Strahler* number

The register function

Number in the root of the tree: *Register function*, a.k.a. *Horton–Strahler* number

- Register function = maximal number of tree trimmings

The register function

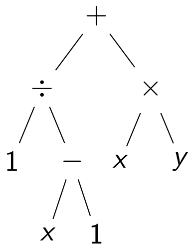
Number in the root of the tree: *Register function*, a.k.a. *Horton–Strahler* number

- Register function = maximal number of tree trimmings
- Applications:

The register function

Number in the root of the tree: *Register function*, a.k.a. *Horton–Strahler* number

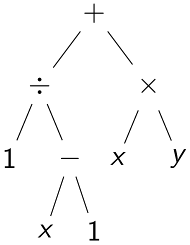
- Register function = maximal number of tree trimmings
- Applications:
  - Required stack size for evaluating an expression



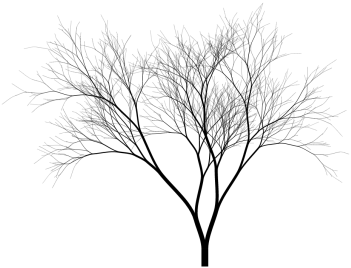
The register function

Number in the root of the tree: *Register function*, a.k.a. *Horton–Strahler* number

- Register function = maximal number of tree trimmings
- Applications:
  - Required stack size for evaluating an expression
  - Branching complexity of river networks (e.g. Danube: 9)



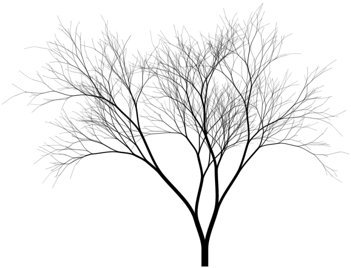
(Rooted) Plane trees



Characterization:

- unlabeled

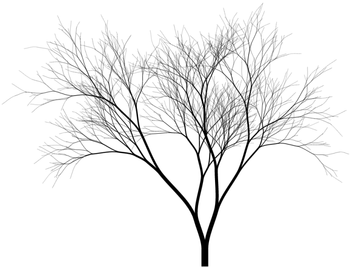
(Rooted) Plane trees



Characterization:

- unlabeled
- special node: root

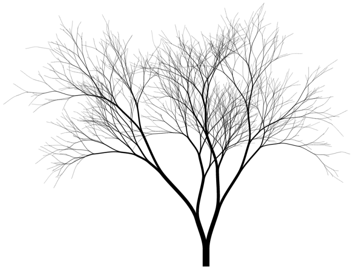
(Rooted) Plane trees



Characterization:

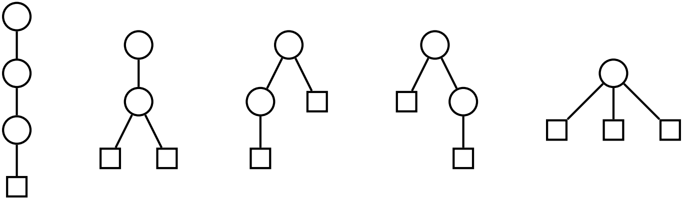
- unlabeled
- special node: root
- order of children matters

(Rooted) Plane trees

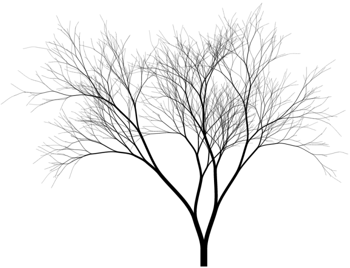


Characterization:

- unlabeled
- special node: root
- order of children matters

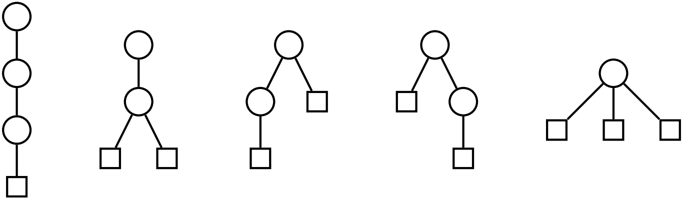


(Rooted) Plane trees



Characterization:

- unlabeled
- special node: root
- order of children matters



- $C_n = \frac{1}{n+1}\binom{2n}{n}$ plane trees of size n

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT | WIEN GRAZ

Growing plane trees

- How can we grow trees?

~~Growing~~ Trimming plane trees

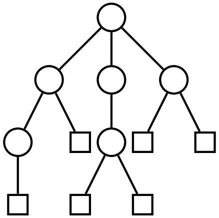
- How can we grow trees?
- Easier question: what could be the inverse operation?

~~Growing~~ Trimming plane trees

- How can we grow trees?
- Easier question: what could be the inverse operation?
  - Most straightforward: cut away all leaves!

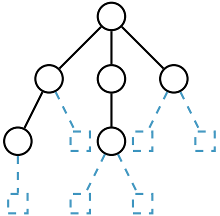
~~Growing~~ Trimming plane trees

- How can we grow trees?
- Easier question: what could be the inverse operation?
  - Most straightforward: cut away all leaves!



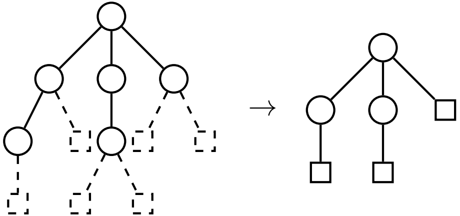
~~Growing~~ Trimming plane trees

- How can we grow trees?
- Easier question: what could be the inverse operation?
  - Most straightforward: cut away all leaves!



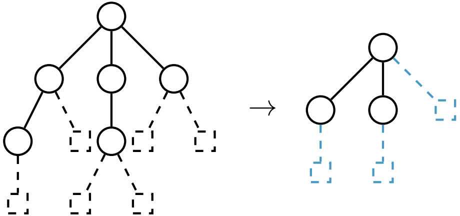
~~Growing~~ Trimming plane trees

- How can we grow trees?
- Easier question: what could be the inverse operation?
  - Most straightforward: cut away all leaves!



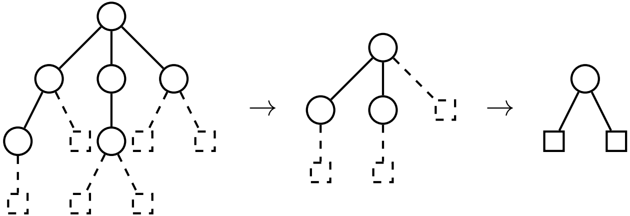
~~Growing~~ Trimming plane trees

- How can we grow trees?
- Easier question: what could be the inverse operation?
  - Most straightforward: cut away all leaves!



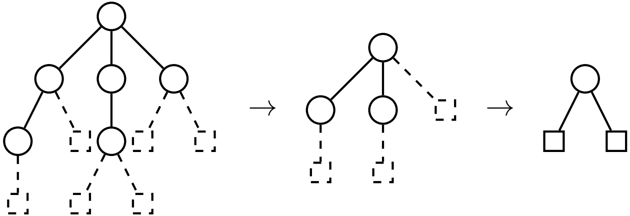
~~Growing~~ Trimming plane trees

- How can we grow trees?
- Easier question: what could be the inverse operation?
  - Most straightforward: cut away all leaves!



~~Growing~~ Trimming plane trees

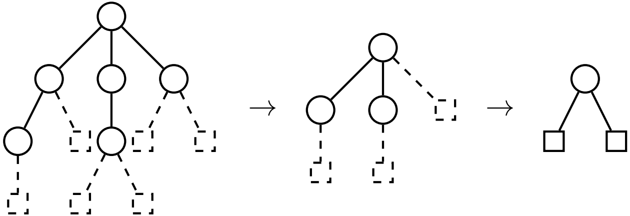
- How can we grow trees?
- Easier question: what could be the inverse operation?
  - Most straightforward: cut away all leaves!



- Growing trees:

~~Growing~~ Trimming plane trees

- How can we grow trees?
- Easier question: what could be the inverse operation?
  - Most straightforward: cut away all leaves!



- Growing trees:
  - grow new leaves out of current leaves and inner nodes

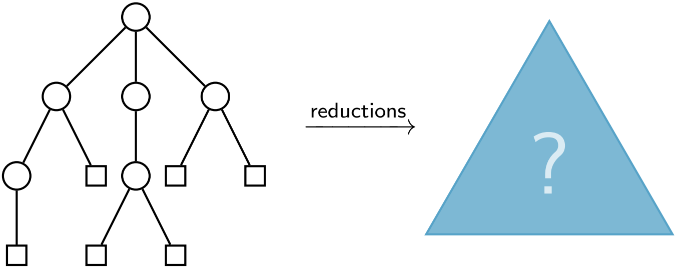
“What?” and “How?”

- **Aim:** analysis of tree structure under iterated reduction

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT | WIEN GRAZ

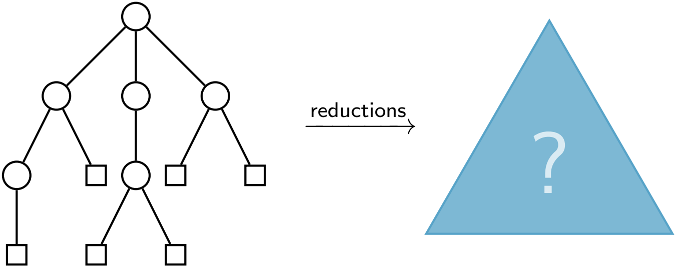
"What?" and "How?"

- **Aim:** analysis of tree structure under iterated reduction



"What?" and "How?"

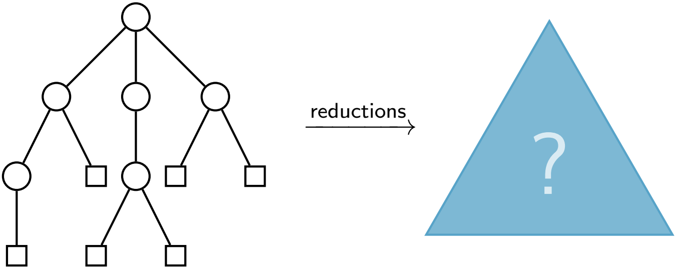
- **Aim:** analysis of tree structure under iterated reduction



- Algorithmic description

"What?" and "How?"

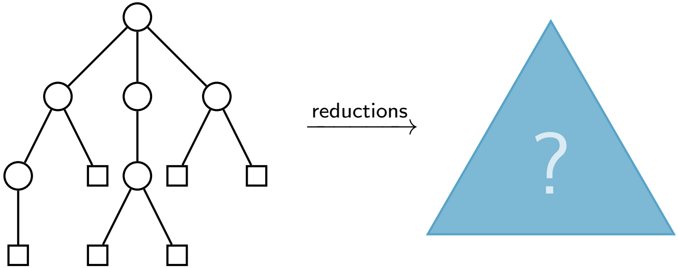
- **Aim:** analysis of tree structure under iterated reduction



- Algorithmic description
- Investigation of "tree expansion" $\rightsquigarrow$ GF

"What?" and "How?"

- **Aim:** analysis of tree structure under iterated reduction



- Algorithmic description
- Investigation of "tree expansion" $\rightsquigarrow$ GF
- Coefficient extraction; Parameter distribution

BGF for plane trees

Proposition

- $\mathcal{T} \ldots$ *plane trees*

BGF for plane trees

Proposition

- $\mathcal{T}$ ... *plane trees*
- $T(z,t)$... *BGF for* $\mathcal{T}$ *(*$z \rightsquigarrow$ *inner nodes,* $t \rightsquigarrow$ *leaves)*

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT I WIEN GRAZ

BGF for plane trees

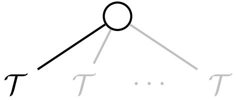
Proposition

- $\mathcal{T}\ldots$ *plane trees*
- $T(z,t)\ldots$ *BGF for* $\mathcal{T}$ *(*$z \rightsquigarrow$ *inner nodes,* $t \rightsquigarrow$ *leaves)*

$$\Rightarrow T(z,t) = \frac{1-(z-t)-\sqrt{1-2(z+t)+(z-t)^2}}{2}$$

Proof. Symbolic equation

$$\mathcal{T} \quad = \quad \square \quad +$$



translates into

$$T(z,t) = t + z \cdot \frac{T(z,t)}{1-T(z,t)}$$

which can be solved explicitly.

BGF for plane trees

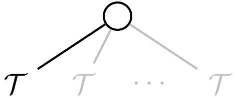
Proposition

- $\mathcal{T}$ ... *plane trees*
- $T(z,t)$ ... *BGF for* $\mathcal{T}$ *(*$z \rightsquigarrow$ *inner nodes,* $t \rightsquigarrow$ *leaves)*

$$\Rightarrow T(z,t) = \frac{1-(z-t)-\sqrt{1-2(z+t)+(z-t)^2}}{2}$$

Proof. Symbolic equation

$$\mathcal{T} = \square +$$



translates into

$$T(z,t) = t + z \cdot \frac{T(z,t)}{1-T(z,t)}$$

which can be solved explicitly.

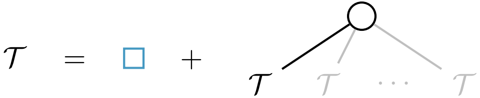
BGF for plane trees

Proposition

- $\mathcal{T}$ ... *plane trees*
- $T(z,t)$ ... *BGF for* $\mathcal{T}$ *(*$z \rightsquigarrow$ *inner nodes,* $t \rightsquigarrow$ *leaves)*

$$\Rightarrow T(z,t) = \frac{1-(z-t)-\sqrt{1-2(z+t)+(z-t)^2}}{2}$$

Proof. Symbolic equation



translates into

$$T(z,t) = t + z \cdot \frac{T(z,t)}{1-T(z,t)}$$

which can be solved explicitly.

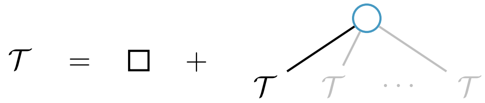
BGF for plane trees

Proposition

- $\mathcal{T}$ . . . *plane trees*
- $T(z,t)$ . . . *BGF for* $\mathcal{T}$ *(*$z \rightsquigarrow$ *inner nodes,* $t \rightsquigarrow$ *leaves)*

$$\Rightarrow T(z,t) = \frac{1-(z-t)-\sqrt{1-2(z+t)+(z-t)^2}}{2}$$

Proof. Symbolic equation



translates into

$$T(z,t) = t + z \cdot \frac{T(z,t)}{1-T(z,t)}$$

which can be solved explicitly.

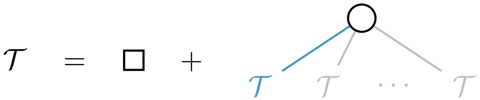
BGF for plane trees

Proposition

- $\mathcal{T}$ . . . *plane trees*
- $T(z, t)$ . . . *BGF for* $\mathcal{T}$ *(*$z \rightsquigarrow$ *inner nodes,* $t \rightsquigarrow$ *leaves)*

$$\Rightarrow T(z,t) = \frac{1-(z-t)-\sqrt{1-2(z+t)+(z-t)^2}}{2}$$

Proof. Symbolic equation



translates into

$$T(z,t) = t + z \cdot \frac{T(z,t)}{1-T(z,t)}$$

which can be solved explicitly.

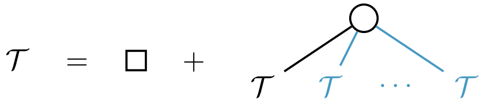
BGF for plane trees

Proposition

- *$\mathcal{T}$ ... plane trees*
- *$T(z,t)$ ... BGF for $\mathcal{T}$ ($z \rightsquigarrow$ inner nodes, $t \rightsquigarrow$ leaves)*

$$\Rightarrow T(z,t) = \frac{1-(z-t)-\sqrt{1-2(z+t)+(z-t)^2}}{2}$$

Proof. Symbolic equation



translates into

$$T(z,t) = t + z \cdot \frac{T(z,t)}{1-T(z,t)}$$

which can be solved explicitly.

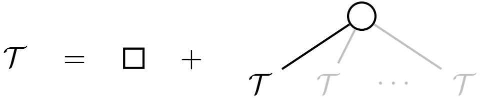
BGF for plane trees

Proposition

- $\mathcal{T}$ ... *plane trees*
- $T(z,t)$ ... *BGF for* $\mathcal{T}$ *(*$z \rightsquigarrow$ *inner nodes,* $t \rightsquigarrow$ *leaves)*

$$\Rightarrow T(z,t) = \frac{1-(z-t)-\sqrt{1-2(z+t)+(z-t)^2}}{2}$$

Proof. Symbolic equation



translates into

$$T(z,t) = t + z \cdot \frac{T(z,t)}{1-T(z,t)}$$

which can be solved explicitly.

Narayana polynomials

- Generating function $T(z, tz)$: $z \ldots$ tree size, $t \ldots$ leaves

Narayana polynomials

- Generating function $T(z, tz)$: $z \dots$ tree size, $t \dots$ leaves
- Expansion:

$$T(z, tz) = zt + z^2 t + z^3(t + t^2) + z^4(t + 3t^2 + t^3) + \dots$$

Narayana polynomials

- Generating function $T(z, tz)$: $z \ldots$ tree size, $t \ldots$ leaves
- Expansion:

$$\begin{aligned} T(z, tz) &= zt + z^2 t + z^3(t + t^2) + z^4(t + 3t^2 + t^3) + \ldots \\ &= \sum_{n \geq 1} z^n N_{n-1}(t) \end{aligned}$$

Narayana polynomials

- Generating function $T(z, tz)$: $z\ldots$ tree size, $t\ldots$ leaves
- Expansion:

$$\begin{aligned} T(z, tz) &= zt + z^2 t + z^3(t + t^2) + z^4(t + 3t^2 + t^3) + \ldots \\ &= \sum_{n\geq 1} z^n N_{n-1}(t) \end{aligned}$$

- $N_{n-1}(t)\ldots$ Narayana polynomial, counts trees of size n (i.e. $n-1$ edges) w.r.t. number of leaves

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT I WIEN GRAZ

Some combinatorial results

Proposition

$$T(z, tz) - tz = T(tz, z) - z$$

Some combinatorial results

Proposition

$$T(z, tz) - tz = T(tz, z) - z$$

Interpretation: for size $n \geq 2$, trees with k leaves are bijective to trees with k inner nodes.

Some combinatorial results

Proposition

$$T(z, tz) - tz = T(tz, z) - z$$

Interpretation: for size $n \geq 2$, trees with k leaves are bijective to trees with k inner nodes.

Proposition

$$N'_{n-1}(1) = \frac{1}{2}\binom{2n-2}{n-1}$$

Some combinatorial results

Proposition

$$T(z, tz) - tz = T(tz, z) - z$$

Interpretation: for size $n \geq 2$, trees with k leaves are bijective to trees with k inner nodes.

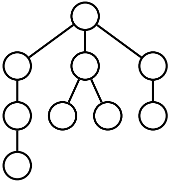
Proposition

$$N'_{n-1}(1) = \frac{1}{2}\binom{2n-2}{n-1}$$

Interpretation: half of all nodes among all trees of size n are leaves.

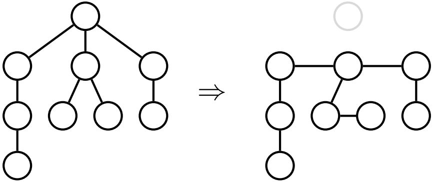
Proof – Rotation correspondence

Construction of "Left-Child Right-Sibling"-tree:



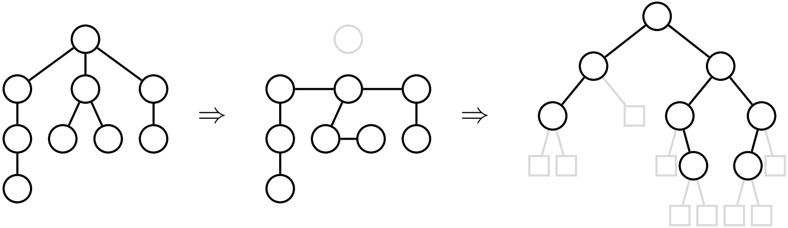
Proof – Rotation correspondence

Construction of "Left-Child Right-Sibling"-tree:



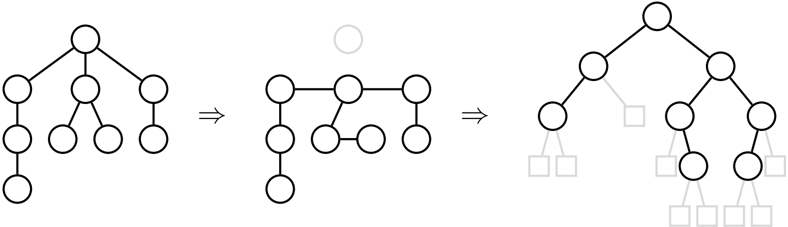
Proof – Rotation correspondence

Construction of "Left-Child Right-Sibling"-tree:



Proof – Rotation correspondence

Construction of “Left-Child Right-Sibling”-tree:

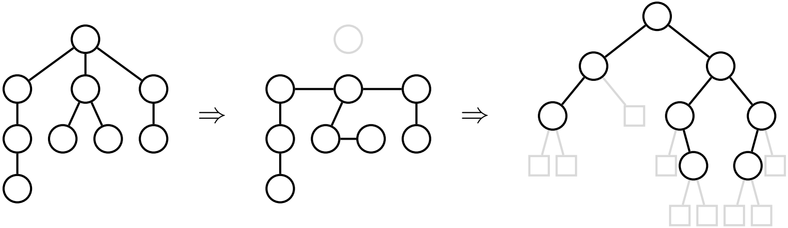


Observations:

- left leaves (binary tree) $\leftrightsquigarrow$ leaves (plane tree)

Proof – Rotation correspondence

Construction of “Left-Child Right-Sibling”-tree:

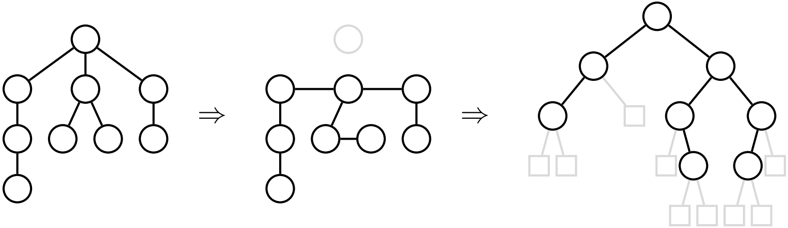


Observations:

- left leaves (binary tree) $\leftrightsquigarrow$ leaves (plane tree)
- right leaves (binary tree) $\leftrightsquigarrow$ inner nodes (plane tree)

Proof – Rotation correspondence

Construction of "Left-Child Right-Sibling"-tree:



Observations:

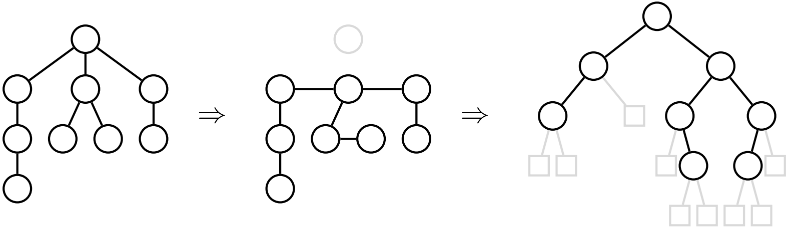
- left leaves (binary tree) ⇝ leaves (plane tree)
- right leaves (binary tree) ⇝ inner nodes (plane tree)

Proofs:

- Proof 1: bijection: mirror binary tree, transform back

Proof – Rotation correspondence

Construction of "Left-Child Right-Sibling"-tree:



Observations:

- left leaves (binary tree) ⇝ leaves (plane tree)
- right leaves (binary tree) ⇝ inner nodes (plane tree)

Proofs:

- Proof 1: bijection: mirror binary tree, transform back
- Proof 2: symmetry: equally many left as right leaves

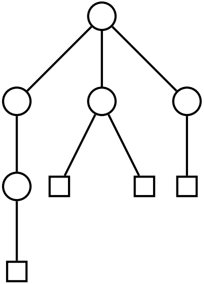
ALPEN-ADRIA UNIVERSITÄT KLAGENFURT | WIEN GRAZ

How do we cut our trees?

- Remove all leaves!

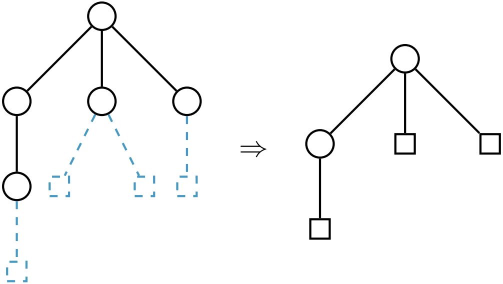
How do we cut our trees?

- Remove all leaves!



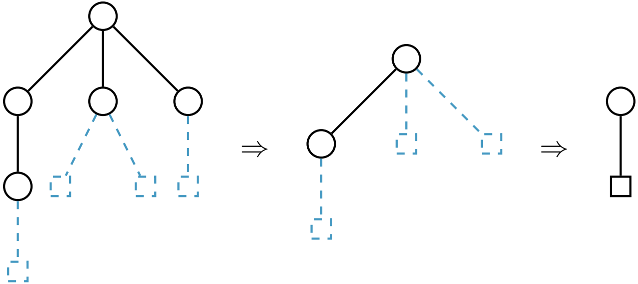
How do we cut our trees?

- Remove all leaves!



How do we cut our trees?

- Remove all leaves!



Expansion operators

- $F\dots$ family of plane trees; BGF $f(z, t)$

Expansion operators

- $F \dots$ family of plane trees; BGF $f(z, t)$
- expansion operator $\Phi \Rightarrow \Phi(f(z, t))$ counts expanded trees

Expansion operators

- $F \dots$ family of plane trees; BGF $f(z, t)$
- expansion operator $\Phi \Rightarrow \Phi(f(z, t))$ counts expanded trees

Leaf expansion Φ_L

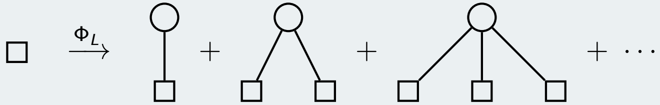
- inverse operation to leaf reduction

Expansion operators

- $F \ldots$ family of plane trees; BGF $f(z, t)$
- expansion operator $\Phi \Rightarrow \Phi(f(z, t))$ counts expanded trees

Leaf expansion Φ_L

- inverse operation to leaf reduction
  - attach leaves to all current leaves (necessary)
  - attach leaves to inner nodes (optional)



$$\Rightarrow \quad \Phi_L(t) = zt + zt^2 + zt^3 + \cdots$$

Leaf expansion operator Φ_L

Proposition

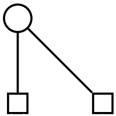
$$\Phi_L(f(z,t)) = (1-t)f\left(\frac{z}{(1-t)^2}, \frac{zt}{(1-t)^2}\right)$$

Leaf expansion operator Φ_L

Proposition

$$\Phi_L(f(z,t)) = (1-t)f\left(\frac{z}{(1-t)^2}, \frac{zt}{(1-t)^2}\right)$$

- Tree with n inner nodes and k leaves $\rightsquigarrow z^n t^k$
- **Expansion:**



- In total:

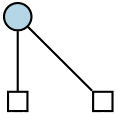
$$\Phi_L(z^n t^k) =$$

Leaf expansion operator Φ_L

Proposition

$$\Phi_L(f(z,t)) = (1-t)f\left(\frac{z}{(1-t)^2}, \frac{zt}{(1-t)^2}\right)$$

- Tree with n inner nodes and k leaves $\rightsquigarrow z^n t^k$
- **Expansion:**
  - inner nodes stay inner nodes



- In total:

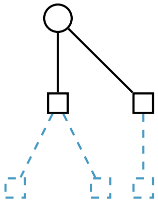
$$\Phi_L(z^n t^k) = z^n \cdot$$

Leaf expansion operator Φ_L

Proposition

$$\Phi_L(f(z,t)) = (1-t)f\left(\frac{z}{(1-t)^2}, \frac{zt}{(1-t)^2}\right)$$

- Tree with n inner nodes and k leaves $\rightsquigarrow z^n t^k$
- **Expansion:**
  - inner nodes stay inner nodes
  - attach a non-empty sequence of leaves to all current leaves



- In total:

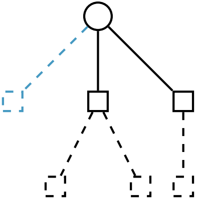
$$\Phi_L(z^n t^k) = z^n \cdot \left(\frac{zt}{1-t}\right)^k.$$

Leaf expansion operator Φ_L

Proposition

$$\Phi_L(f(z,t)) = (1-t)f\left(\frac{z}{(1-t)^2}, \frac{zt}{(1-t)^2}\right)$$

- Tree with n inner nodes and k leaves $\rightsquigarrow z^n t^k$
- **Expansion:**
  - inner nodes stay inner nodes
  - attach a non-empty sequence of leaves to all current leaves
  - there are $2n+k-1$ positions where sequences of leaves can be inserted



- In total:

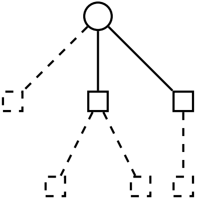
$$\Phi_L(z^n t^k) = z^n \cdot \left(\frac{zt}{1-t}\right)^k \cdot \frac{1}{(1-t)^{2n+k-1}}$$

Leaf expansion operator Φ_L

Proposition

$$\Phi_L(f(z,t)) = (1-t) f\left(\frac{z}{(1-t)^2}, \frac{zt}{(1-t)^2}\right)$$

- Tree with n inner nodes and k leaves $\rightsquigarrow z^n t^k$
- **Expansion:**
  - inner nodes stay inner nodes
  - attach a non-empty sequence of leaves to all current leaves
  - there are $2n+k-1$ positions where sequences of leaves can be inserted



- In total:

$$\Phi_L(z^n t^k) = z^n \cdot \left(\frac{zt}{1-t}\right)^k \cdot \frac{1}{(1-t)^{2n+k-1}}$$

- As Φ_L is linear, this proves the proposition.

Properties of Φ_L

- Functional equation: $T(z, t) = \Phi_L(T(z, t)) + t$

Properties of Φ_L

- Functional equation: $T(z,t) = \Phi_L(T(z,t)) + t$
- With $z = u/(1+u)^2$ and by some manipulations

$$\Phi_L^r(z^n t^k)|_{t=z} = \frac{1-u^{r+2}}{(1-u^{r+1})(1+u)} \left(\frac{u(1-u^{r+1})^2}{(1-u^{r+2})^2}\right)^n \left(\frac{u^{r+1}(1-u)^2}{(1-u^{r+2})^2}\right)^k$$

Properties of Φ_L

- Functional equation: $T(z,t) = \Phi_L(T(z,t)) + t$
- With $z = u/(1+u)^2$ and by some manipulations

$$\Phi_L^r(z^n t^k)|_{t=z} = \frac{1-u^{r+2}}{(1-u^{r+1})(1+u)}\Big(\frac{u(1-u^{r+1})^2}{(1-u^{r+2})^2}\Big)^n\Big(\frac{u^{r+1}(1-u)^2}{(1-u^{r+2})^2}\Big)^k$$

- BGF $G_r(z,v)$ for size comparison: z tracks original size, v size of r-fold reduced tree

Properties of Φ_L

- Functional equation: $T(z,t) = \Phi_L(T(z,t)) + t$
- With $z = u/(1+u)^2$ and by some manipulations

$$\Phi_L^r(z^n t^k)|_{t=z} = \frac{1-u^{r+2}}{(1-u^{r+1})(1+u)}\Bigl(\frac{u(1-u^{r+1})^2}{(1-u^{r+2})^2}\Bigr)^n\Bigl(\frac{u^{r+1}(1-u)^2}{(1-u^{r+2})^2}\Bigr)^k$$

- BGF $G_r(z,v)$ for size comparison: z tracks original size, v size of r-fold reduced tree
- Intuition: v “remembers” size while tree family is expanded

Properties of Φ_L

- Functional equation: $T(z,t) = \Phi_L(T(z,t)) + t$
- With $z = u/(1+u)^2$ and by some manipulations

$$\Phi_L^r(z^n t^k)|_{t=z} = \frac{1-u^{r+2}}{(1-u^{r+1})(1+u)} \Big(\frac{u(1-u^{r+1})^2}{(1-u^{r+2})^2}\Big)^n \Big(\frac{u^{r+1}(1-u)^2}{(1-u^{r+2})^2}\Big)^k$$

- BGF $G_r(z,v)$ for size comparison: z tracks original size, v size of r-fold reduced tree
- Intuition: v "remembers" size while tree family is expanded

$$G_r(z,v) = \frac{1-u^{r+2}}{(1-u^{r+1})(1+u)} T\Big(\frac{u(1-u^{r+1})^2}{(1-u^{r+2})^2} v, \frac{u^{r+1}(1-u)^2}{(1-u^{r+2})^2} v\Big)$$

Cutting leaves

Theorem (H.–Heuberger–Kropf–Prodinger, 2016)

- *$r \ldots$ number of reductions, fixed*

Cutting leaves

Theorem (H.–Heuberger–Kropf–Prodinger, 2016)

- *r . . . number of reductions, fixed*
- *$X_{n,r}$ . . . RV for size of r-fold leaf-reduced tree with originally n nodes*

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT | WIEN GRAZ

Cutting leaves

Theorem (H.–Heuberger–Kropf–Prodinger, 2016)

- *r ... number of reductions, fixed*
- *$X_{n,r}$ ... RV for size of r-fold leaf-reduced tree with originally n nodes*

Then the expected size of the reduced tree and the corresponding variance are

$$\mathbb{E}X_{n,r} = \frac{n}{r+1} - \frac{r(r-1)}{6(r+1)} + O(n^{-1}),$$

Cutting leaves

Theorem (H.–Heuberger–Kropf–Prodinger, 2016)

- *$r \ldots$ number of reductions, fixed*
- *$X_{n,r} \ldots$ RV for size of r-fold leaf-reduced tree with originally n nodes*

Then the expected size of the reduced tree and the corresponding variance are

$$\mathbb{E}X_{n,r} = \frac{n}{r+1} - \frac{r(r-1)}{6(r+1)} + O(n^{-1}),$$

$$\mathbb{V}X_{n,r} = \frac{r(r+2)}{6(r+1)^2} n + O(1),$$

Cutting leaves

Theorem (H.–Heuberger–Kropf–Prodinger, 2016)

- *$r \ldots$ number of reductions, fixed*
- *$X_{n,r} \ldots$ RV for size of r-fold leaf-reduced tree with originally n nodes*

Then the expected size of the reduced tree and the corresponding variance are

$$\mathbb{E}X_{n,r} = \frac{n}{r+1} - \frac{r(r-1)}{6(r+1)} + O(n^{-1}),$$

$$\mathbb{V}X_{n,r} = \frac{r(r+2)}{6(r+1)^2} n + O(1),$$

and $X_{n,r}$ is asymptotically normally distributed.

Cutting leaves – Some insights

- $\mathbb{E}X_{n,r}$ and $\mathbb{V}X_{n,r}$ follow via singularity analysis

Cutting leaves – Some insights

- $\mathbb{E}X_{n,r}$ and $\mathbb{V}X_{n,r}$ follow via singularity analysis
- Asymptotic normality: $X_{n,r}$ is a tree parameter with small toll function, limit law by Wagner (2015)

Cutting leaves – Some insights

- $\mathbb{E}X_{n,r}$ and $\mathbb{V}X_{n,r}$ follow via singularity analysis
- Asymptotic normality: $X_{n,r}$ is a tree parameter with small toll function, limit law by Wagner (2015)
- We can even get all factorial moments:

$$\mathbb{E}X_{n,r}^{\underline{d}} = \frac{1}{(r+1)^d} n^d + O(n^{d-1})$$

Cutting leaves – Some insights

- $\mathbb{E}X_{n,r}$ and $\mathbb{V}X_{n,r}$ follow via singularity analysis
- Asymptotic normality: $X_{n,r}$ is a tree parameter with small toll function, limit law by Wagner (2015)
- We can even get all factorial moments:

$$\mathbb{E}X_{n,r}^{\underline{d}} = \frac{1}{(r+1)^d} n^d + O(n^{d-1})$$

Cutting leaves – factorial moments

- $C_{n-1}\mathbb{E}X_{n,r}^{\underline{d}}$ is extracted from $\frac{\partial^d}{\partial v^d} G_r(z, v)|_{v=1}$

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT | WIEN GRAZ

Cutting leaves – factorial moments

- $C_{n-1}\mathbb{E}X_{n,r}^{\underline{d}}$ is extracted from $\frac{\partial^d}{\partial v^d} G_r(z, v)|_{v=1}$
- **Problem:** general derivative unknown

Cutting leaves – factorial moments

- $C_{n-1}\mathbb{E}X_{n,r}^{\underline{d}}$ is extracted from $\frac{\partial^d}{\partial v^d} G_r(z,v)|_{v=1}$
- **Problem:** general derivative unknown
- **Solution:**

$$\sum_{d\geq 0} \frac{q^d}{d!} \frac{\partial^d}{\partial v^d} G_r(z,v)|_{v=1}$$

Cutting leaves – factorial moments

- $C_{n-1}\mathbb{E}X_{n,r}^{\underline{d}}$ is extracted from $\frac{\partial^d}{\partial v^d} G_r(z, v)|_{v=1}$
- **Problem:** general derivative unknown
- **Solution:**

$$\sum_{d\geq 0} \frac{q^d}{d!} \frac{\partial^d}{\partial v^d} G_r(z, v)|_{v=1} = G_r(z, 1+q)$$

Cutting leaves – factorial moments

- $C_{n-1}\mathbb{E}X_{n,r}^{\underline{d}}$ is extracted from $\frac{\partial^d}{\partial v^d} G_r(z,v)|_{v=1}$
- **Problem:** general derivative unknown
- **Solution:**

$$\sum_{d\geq 0} \frac{q^d}{d!} \frac{\partial^d}{\partial v^d} G_r(z,v)|_{v=1} = G_r(z, 1+q)$$

$$= c \cdot T(a(1+q), b(1+q))$$

Cutting leaves – factorial moments

- $C_{n-1}\mathbb{E}X_{n,r}^{\underline{d}}$ is extracted from $\frac{\partial^d}{\partial v^d} G_r(z,v)|_{v=1}$
- **Problem:** general derivative unknown
- **Solution:**

$$\sum_{d\geq 0} \frac{q^d}{d!} \frac{\partial^d}{\partial v^d} G_r(z,v)|_{v=1} = G_r(z, 1+q)$$

$$= c \cdot T(a(1+q), b(1+q))$$

$$= \delta + \Delta \cdot T(\alpha q, \beta q)$$

Cutting leaves – factorial moments

- $C_{n-1}\mathbb{E}X_{n,r}^{\underline{d}}$ is extracted from $\frac{\partial^d}{\partial v^d} G_r(z,v)|_{v=1}$
- **Problem:** general derivative unknown
- **Solution:**

$$\sum_{d\geq 0} \frac{q^d}{d!} \frac{\partial^d}{\partial v^d} G_r(z,v)|_{v=1} = G_r(z, 1+q)$$

$$= c \cdot T(a(1+q), b(1+q))$$

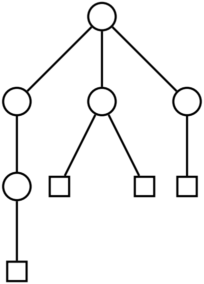
$$= \delta + \Delta \cdot T(\alpha q, \beta q)$$

- This allows extracting the coefficient of $z^n q^d$

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT I WIEN GRAZ

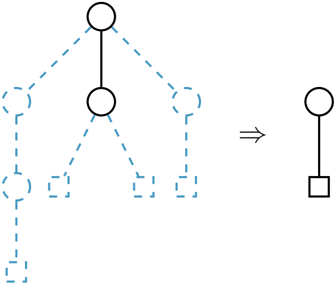
How do we cut our trees? (2)

- Remove all paths that end in a leaf!



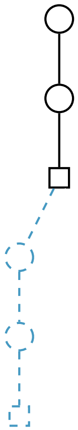
How do we cut our trees? (2)

- Remove all paths that end in a leaf!



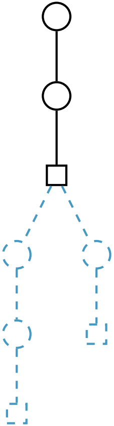
Path expansions

- Append one path to leaf $\rightsquigarrow$ longer path $\lightning$



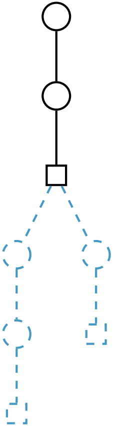
Path expansions

- Append one path to leaf $\rightsquigarrow$ longer path $\lightning$
- $\Rightarrow$ at least two paths need to be appended



Path expansions

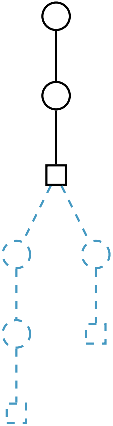
- Append one path to leaf $\rightsquigarrow$ longer path ↯
- $\Rightarrow$ at least two paths need to be appended
- Write $p = \frac{t}{1-z}$ … BGF for paths



Path expansions

- Append one path to leaf ⇝ longer path ↯
- ⇒ at least two paths need to be appended
- Write $p = \frac{t}{1-z}$ ... BGF for paths
- Similar to before we obtain

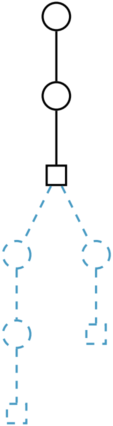
$$\Phi_P(z^n t^k) = z^n \cdot \frac{z^k p^{2k}}{(1-p)^k} \cdot \frac{1}{(1-p)^{2n+k-1}}$$



Path expansions

- Append one path to leaf $\rightsquigarrow$ longer path $\lightning$
- $\Rightarrow$ at least two paths need to be appended
- Write $p = \frac{t}{1-z}$ ... BGF for paths
- Similar to before we obtain

$$\Phi_P(z^n t^k) = z^n \cdot \frac{z^k p^{2k}}{(1-p)^k} \cdot \frac{1}{(1-p)^{2n+k-1}}$$



Proposition

The linear operator given by

$$\Phi_P(f(z,t)) = (1-p) f\left(\frac{z}{(1-p)^2}, \frac{zp^2}{(1-p)^2}\right)$$

is the path expansion operator.

Generating function for path reductions

Proposition

BGF for size comparison ($z \rightsquigarrow$ original size, $v \rightsquigarrow$ r-fold path reduced size) is

$$\frac{1 - u^{2^{r+1}}}{(1 - u^{2^{r+1}-1})(1+u)} T\Big(\frac{u(1 - u^{2^{r+1}-1})^2}{(1 - u^{2^{r+1}})^2} v, \frac{u^{2^{r+1}-1}(1-u)^2}{(1 - u^{2^{r+1}})^2} v\Big),$$

where $z = u/(1+u)^2$.

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT | WIEN GRAZ

Generating function for path reductions

Proposition

BGF for size comparison ($z \rightsquigarrow$ original size, $v \rightsquigarrow$ r-fold path reduced size) is

$$\frac{1-u^{2^{r+1}}}{(1-u^{2^{r+1}-1})(1+u)}T\Big(\frac{u(1-u^{2^{r+1}-1})^2}{(1-u^{2^{r+1}})^2}v, \frac{u^{2^{r+1}-1}(1-u)^2}{(1-u^{2^{r+1}})^2}v\Big),$$

where $z = u/(1+u)^2$.

Observation. This is the BGF for leaf reductions

$$\frac{1-u^{r+2}}{(1-u^{r+1})(1+u)}T\Big(\frac{u(1-u^{r+1})^2}{(1-u^{r+2})^2}v, \frac{u^{r+1}(1-u)^2}{(1-u^{r+2})^2}v\Big)$$

with $r \mapsto 2^{r+1}-2$.

Cutting paths – Pruning

Theorem (H.–Heuberger–Kropf–Prodinger, 2016)

- *$r \dots$ number of reductions, fixed*

Cutting paths – Pruning

Theorem (H.–Heuberger–Kropf–Prodinger, 2016)

- *$r \dots$ number of reductions, fixed*
- *$X_{n,r} \dots$ RV for size of r-fold path-reduced tree with originally n nodes*

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT | WIEN GRAZ

Cutting paths – Pruning

Theorem (H.–Heuberger–Kropf–Prodinger, 2016)

- *r . . . number of reductions, fixed*
- *$X_{n,r}$ . . . RV for size of r-fold path-reduced tree with originally n nodes*

Then the expected size of the reduced tree and the corresponding variance are

$$\mathbb{E}X_{n,r} = \frac{n}{2^{r+1} - 1} - \frac{(2^r - 1)(2^{r+1} - 3)}{3(2^{r+1} - 1)} + O(n^{-1}),$$

Cutting paths – Pruning

Theorem (H.–Heuberger–Kropf–Prodinger, 2016)

- *r . . . number of reductions, fixed*
- *$X_{n,r}$ . . . RV for size of r-fold path-reduced tree with originally n nodes*

Then the expected size of the reduced tree and the corresponding variance are

$$\mathbb{E}X_{n,r} = \frac{n}{2^{r+1}-1} - \frac{(2^r-1)(2^{r+1}-3)}{3(2^{r+1}-1)} + O(n^{-1}),$$

$$\mathbb{V}X_{n,r} = \frac{2^{r+1}(2^r-1)}{3(2^{r+1}-1)^2}n + O(1).$$

Cutting paths – Pruning

Theorem (H.–Heuberger–Kropf–Prodinger, 2016)

- *r . . . number of reductions, fixed*
- *$X_{n,r}$ . . . RV for size of r-fold path-reduced tree with originally n nodes*

Then the expected size of the reduced tree and the corresponding variance are

$$\mathbb{E}X_{n,r} = \frac{n}{2^{r+1}-1} - \frac{(2^r-1)(2^{r+1}-3)}{3(2^{r+1}-1)} + O(n^{-1}),$$

$$\mathbb{V}X_{n,r} = \frac{2^{r+1}(2^r-1)}{3(2^{r+1}-1)^2}n + O(1).$$

Furthermore, $X_{n,r}$ is asymptotically normally distributed.

Cutting paths – Pruning

Theorem (H.–Heuberger–Kropf–Prodinger, 2016)

- *r … number of reductions, fixed*
- *$X_{n,r}$ … RV for size of r-fold path-reduced tree with originally n nodes*

Then the expected size of the reduced tree and the corresponding variance are

$$\mathbb{E}X_{n,r} = \frac{n}{2^{r+1}-1} - \frac{(2^r-1)(2^{r+1}-3)}{3(2^{r+1}-1)} + O(n^{-1}),$$

$$\mathbb{V}X_{n,r} = \frac{2^{r+1}(2^r-1)}{3(2^{r+1}-1)^2}n + O(1).$$

Furthermore, $X_{n,r}$ is asymptotically normally distributed.

- Factorial moments are known as well

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT | WIEN GRAZ

Cutting paths – Pruning

Theorem (H.–Heuberger–Kropf–Prodinger, 2016)

- *$r \dots$ number of reductions, fixed*
- *$X_{n,r} \dots$ RV for size of r-fold path-reduced tree with originally n nodes*

Then the expected size of the reduced tree and the corresponding variance are

$$\mathbb{E}X_{n,r} = \frac{n}{2^{r+1}-1} - \frac{(2^r-1)(2^{r+1}-3)}{3(2^{r+1}-1)} + O(n^{-1}),$$

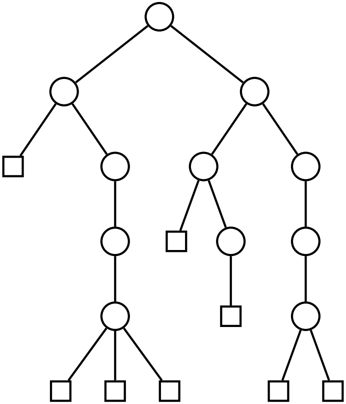
$$\mathbb{V}X_{n,r} = \frac{2^{r+1}(2^r-1)}{3(2^{r+1}-1)^2} n + O(1).$$

Furthermore, $X_{n,r}$ is asymptotically normally distributed.

- Factorial moments are known as well
- Proof: subsequence of RV's from cutting leaves

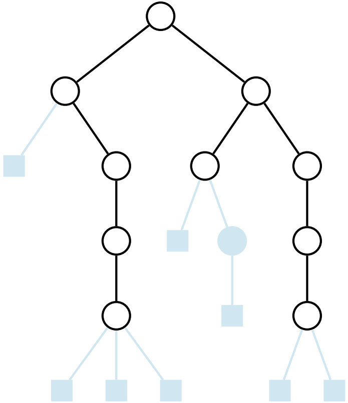
Counting total number of paths

- Trees can be partitioned into paths ($\rightsquigarrow$ branches)!



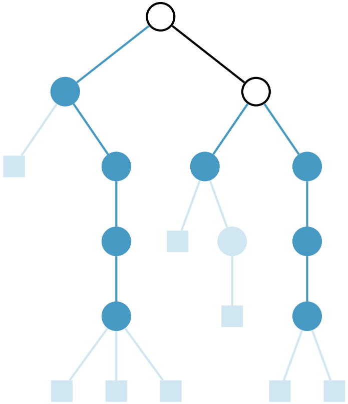
Counting total number of paths

- Trees can be partitioned into paths ($\rightsquigarrow$ branches)!



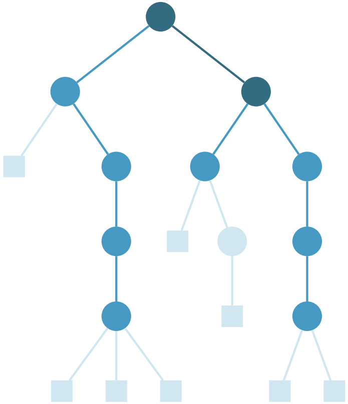
Counting total number of paths

- Trees can be partitioned into paths ($\rightsquigarrow$ branches)!



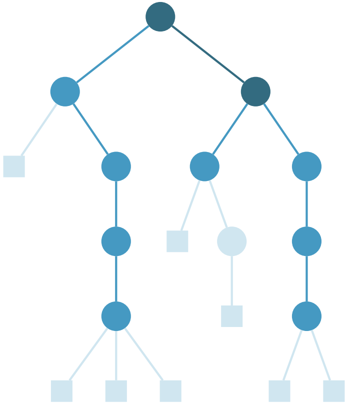
Counting total number of paths

- Trees can be partitioned into paths ($\rightsquigarrow$ branches)!



Counting total number of paths

- Trees can be partitioned into paths ($\rightsquigarrow$ branches)!



- Average number of paths?

Cutting paths – total number of paths

Theorem (H.–Heuberger–Kropf–Prodinger, 2017)

- P_n . . . RV for number of paths in tree of size n

Cutting paths – total number of paths

Theorem (H.–Heuberger–Kropf–Prodinger, 2017)

- P_n ... RV for number of paths in tree of size n

The expected number of paths is

$$\mathbb{E}P_n = (\alpha - 1)n + \frac{1}{6}\log_4 n + \delta(\log_4 n) + c + O(n^{-1/2}).$$

Cutting paths – total number of paths

Theorem (H.–Heuberger–Kropf–Prodinger, 2017)

- P_n ... RV for number of paths in tree of size n

The expected number of paths is

$$\mathbb{E}P_n = (\alpha - 1)n + \frac{1}{6}\log_4 n + \delta(\log_4 n) + c + O(n^{-1/2}).$$

- $\delta(x) := \frac{1}{\log 2}\sum_{k\neq 0}(-1+\chi_k)\Gamma(\chi_k/2)\zeta(-1+\chi_k)e^{2k\pi ix}$,
- $\alpha := \sum_{k\geq 1} 1/(2^k - 1) \approx 1.606695$,
- $c \approx -0.118105$.

Cutting paths – total number of paths

Theorem (H.–Heuberger–Kropf–Prodinger, 2017)

- P_n . . . RV for number of paths in tree of size n

The expected number of paths is

$$\mathbb{E}P_n = (\alpha - 1)n + \frac{1}{6}\log_4 n + \delta(\log_4 n) + c + O(n^{-1/2}).$$

- $\delta(x) := \frac{1}{\log 2}\sum_{k\neq 0}(-1+\chi_k)\Gamma(\chi_k/2)\zeta(-1+\chi_k)e^{2k\pi i x}$,
- $\alpha := \sum_{k\geq 1} 1/(2^k - 1) \approx 1.606695$,
- $c \approx -0.118105$.

- **Proof:** Sum of leaves in all reductions, Mellin-transform, singularity analysis.

How do we cut our trees? (3)

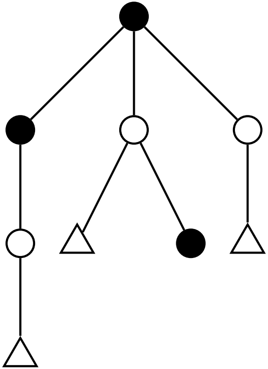
- Introduced by Chen, Deutsch, Elizalde (2006)

How do we cut our trees? (3)

- Introduced by Chen, Deutsch, Elizalde (2006)

Old leaves

- Remove all leaves that are leftmost children

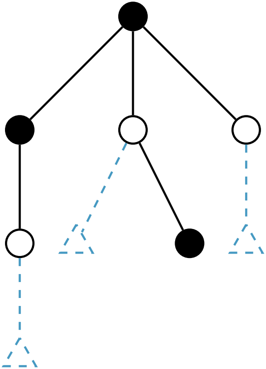


How do we cut our trees? (3)

- Introduced by Chen, Deutsch, Elizalde (2006)

Old leaves

- Remove all leaves that are leftmost children

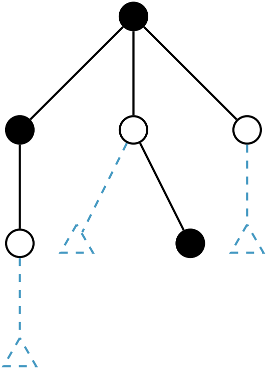


How do we cut our trees? (3)

- Introduced by Chen, Deutsch, Elizalde (2006)

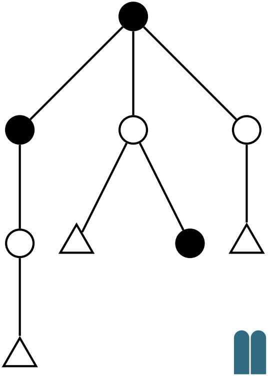
Old leaves

- Remove all leaves that are leftmost children



Old paths

- Remove all paths consisting of leftmost children

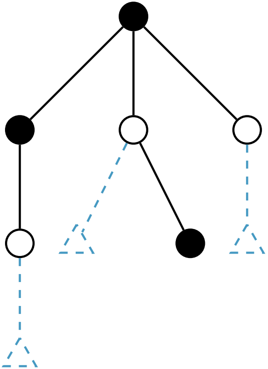


How do we cut our trees? (3)

- Introduced by Chen, Deutsch, Elizalde (2006)

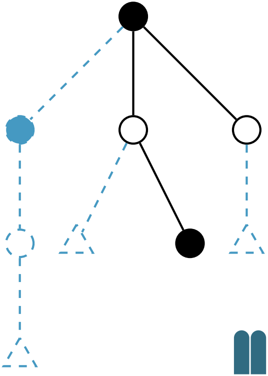
Old leaves

- Remove all leaves that are leftmost children



Old paths

- Remove all paths consisting of leftmost children



Preliminaries

Proposition

- $\mathcal{L}\ldots$ *plane trees*

Preliminaries

Proposition

- *$\mathcal{L}$... plane trees*
- *$L(z, w)$... BGF ($w \rightsquigarrow$ old leaves,*
  $z \rightsquigarrow$ all nodes that are neither old leaves nor parents thereof)

Preliminaries

Proposition

- *$\mathcal{L}$... plane trees*
- *$L(z, w)$... BGF ($w \rightsquigarrow$ old leaves,*
  $z \rightsquigarrow$ all nodes that are neither old leaves nor parents thereof)

Then

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

Preliminaries

Proposition

- *$\mathcal{L}$... plane trees*
- *$L(z,w)$... BGF ($w \rightsquigarrow$ old leaves,*
  $z \rightsquigarrow$ all nodes that are neither old leaves nor parents thereof)

Then

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

and there are $C_{k-1}\binom{n-2}{n-2k}2^{n-2k}$ trees of size n with k old leaves.

Proof. Symbolic equation

$$\mathcal{L} = \bullet$$

Preliminaries

Proposition

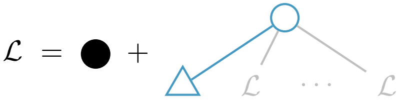
- *$\mathcal{L}\ldots$ plane trees*
- *$L(z, w)\ldots$ BGF ($w \rightsquigarrow$ old leaves,*
  $z \rightsquigarrow$ all nodes that are neither old leaves nor parents thereof)

Then

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

and there are $C_{k-1}\binom{n-2}{n-2k}2^{n-2k}$ trees of size n with k old leaves.

Proof. Symbolic equation



Preliminaries

Proposition

- *$\mathcal{L}$… plane trees*
- *$L(z, w)$… BGF ($w \rightsquigarrow$ old leaves, $z \rightsquigarrow$ all nodes that are neither old leaves nor parents thereof)*

Then

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

and there are $C_{k-1}\binom{n-2}{n-2k}2^{n-2k}$ trees of size n with k old leaves.

Proof. Symbolic equation

translation; series expansion of the root.

Expansion operators

Proposition

The operators for "old leaf"- and "old path"-expansions are given by

$$\Phi_{OL}(f(z,w)) = f(z+w,(2z+w)w)$$

Expansion operators

Proposition

The operators for "old leaf"- and "old path"-expansions are given by

$$\Phi_{OL}(f(z,w)) = f(z+w, (2z+w)w)$$

and

$$\Phi_{OP}(f(z,w)) = f\left(z + \frac{w}{1-z}, \left(z + \frac{w}{1-z}\right)\frac{w}{1-z}\right),$$

respectively.

Expansion operators

Proposition

The operators for "old leaf"- and "old path"-expansions are given by

$$\Phi_{OL}(f(z,w)) = f(z+w, (2z+w)w)$$

and

$$\Phi_{OP}(f(z,w)) = f\left(z + \frac{w}{1-z}, \left(z + \frac{w}{1-z}\right)\frac{w}{1-z}\right),$$

respectively.

Proof for old leaves.

$w \triangleq$ 

Expansion operators

Proposition

The operators for "old leaf"- and "old path"-expansions are given by

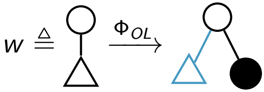
$$\Phi_{OL}(f(z,w)) = f(z+w,(2z+w)w)$$

and

$$\Phi_{OP}(f(z,w)) = f\left(z+\frac{w}{1-z},\left(z+\frac{w}{1-z}\right)\frac{w}{1-z}\right),$$

respectively.

Proof for old leaves.



Expansion operators

Proposition

The operators for “old leaf”- and “old path”-expansions are given by

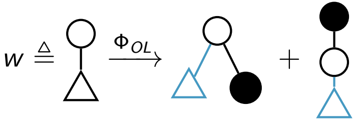
$$\Phi_{OL}(f(z,w)) = f(z+w, (2z+w)w)$$

and

$$\Phi_{OP}(f(z,w)) = f\left(z + \frac{w}{1-z}, \left(z + \frac{w}{1-z}\right)\frac{w}{1-z}\right),$$

respectively.

Proof for old leaves.



Expansion operators

Proposition

The operators for "old leaf"- and "old path"-expansions are given by

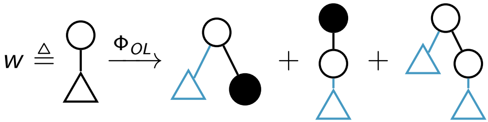
$$\Phi_{OL}(f(z,w)) = f(z+w,(2z+w)w)$$

and

$$\Phi_{OP}(f(z,w)) = f\left(z+\frac{w}{1-z},\left(z+\frac{w}{1-z}\right)\frac{w}{1-z}\right),$$

respectively.

Proof for old leaves.



Expansion operators

Proposition

The operators for "old leaf"- and "old path"-expansions are given by

$$\Phi_{OL}(f(z,w)) = f(z+w,(2z+w)w)$$

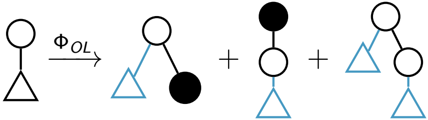
and

$$\Phi_{OP}(f(z,w)) = f\left(z+\frac{w}{1-z},\left(z+\frac{w}{1-z}\right)\frac{w}{1-z}\right),$$

respectively.

Proof for old leaves.

$w \triangleq$



$\triangleq zw + zw + w^2$

Expansion operators

Proposition

The operators for "old leaf"- and "old path"-expansions are given by

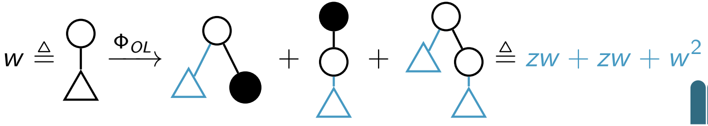
$$\Phi_{OL}(f(z,w)) = f(z+w, (2z+w)w)$$

and

$$\Phi_{OP}(f(z,w)) = f\left(z + \frac{w}{1-z}, \left(z + \frac{w}{1-z}\right)\frac{w}{1-z}\right),$$

respectively.

Proof for old leaves.



Cutting old leaves

Theorem (H.–Heuberger–Kropf–Prodinger, 2016)

- *$r\ldots$ number of reductions, fixed*

Cutting old leaves

Theorem (H.–Heuberger–Kropf–Prodinger, 2016)

- *$r \ldots$ number of reductions, fixed*
- *$B_h(z) \ldots$ polynomial enumerating binary trees of height $\leq h$*

Cutting old leaves

Theorem (H.–Heuberger–Kropf–Prodinger, 2016)

- *r ... number of reductions, fixed*
- *$B_h(z)$ ... polynomial enumerating binary trees of height $\leq h$*

Then the expected reduced tree size after r "old leaf"-reductions and the corresponding variance are given by

$$\mathbb{E}X_{n,r} = (2 - B_r(1/4))n - \frac{B_r'(1/4)}{8} + O(n^{-1}),$$

Cutting old leaves

Theorem (H.–Heuberger–Kropf–Prodinger, 2016)

- *r . . . number of reductions, fixed*
- *$B_h(z)$ . . . polynomial enumerating binary trees of height $\leq h$*

Then the expected reduced tree size after r "old leaf"-reductions and the corresponding variance are given by

$$\mathbb{E}X_{n,r} = (2 - B_r(1/4))n - \frac{B_r'(1/4)}{8} + O(n^{-1}),$$

$$\mathbb{V}X_{n,r} = \Big(B_r(1/4) - B_r(1/4)^2 + \frac{(2 - B_r(1/4))B_r'(1/4)}{2}\Big)n + O(1).$$

In addition, $X_{n,r}$ is asymptotically normally distributed.

Cutting old leaves – Details

$$\mathbb{E}X_{n,r} \sim (2 - B_r(1/4))n$$

Cutting old leaves – Details

$$\mathbb{E}X_{n,r} \sim (2 - B_r(1/4))n$$

- **Note.** Via Flajolet, Odlyzko (1982):

$$B_r(1/4) = 2 - \frac{4}{r} + \frac{4\log r}{r^2} + O(r^{-3}), \quad r \to \infty$$

Cutting old leaves – Details

$$\mathbb{E}X_{n,r} \sim (2 - B_r(1/4))n$$

- **Note.** Via Flajolet, Odlyzko (1982):
$$B_r(1/4) = 2 - \frac{4}{r} + \frac{4\log r}{r^2} + O(r^{-3}), \quad r \to \infty$$
- Limiting distribution:
  - $n - X_{n,r}$ is a *local tree functional*

Cutting old leaves – Details

$$\mathbb{E}X_{n,r} \sim (2 - B_r(1/4))n$$

- **Note.** Via Flajolet, Odlyzko (1982):
$$B_r(1/4) = 2 - \frac{4}{r} + \frac{4\log r}{r^2} + O(r^{-3}), \quad r \to \infty$$
- Limiting distribution:
  - $n - X_{n,r}$ is a *local tree functional*
  - toll function can be evaluated from a fixed part of the tree

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT | WIEN GRAZ

Cutting old leaves – Details

$$\mathbb{E}X_{n,r} \sim (2 - B_r(1/4))n$$

- **Note.** Via Flajolet, Odlyzko (1982):

$$B_r(1/4) = 2 - \frac{4}{r} + \frac{4\log r}{r^2} + O(r^{-3}), \quad r \to \infty$$

- Limiting distribution:
  - $n - X_{n,r}$ is a *local tree functional*
  - toll function can be evaluated from a fixed part of the tree
  - limit law then follows from a result by Janson (2016)

Cutting old paths

Theorem (H.–Heuberger–Kropf–Prodinger, 2016)

- *$r \ldots$ number of reductions, fixed*

Cutting old paths

Theorem (H.–Heuberger–Kropf–Prodinger, 2016)

- *$r \dots$ number of reductions, fixed*
- *$X_{n,r} \dots$ RV for size of r-fold "old path"-reduced tree with originally n nodes*

Cutting old paths

Theorem (H.–Heuberger–Kropf–Prodinger, 2016)

- *$r \dots$ number of reductions, fixed*
- *$X_{n,r} \dots$ RV for size of r-fold "old path"-reduced tree with originally n nodes*

Then the expected size of the reduced tree and the corresponding variance are

$$\mathbb{E}X_{n,r} = \frac{2n}{r+2} - \frac{r(r+1)}{3(r+2)} + O(n^{-1}),$$

Cutting old paths

Theorem (H.–Heuberger–Kropf–Prodinger, 2016)

- *r . . . number of reductions, fixed*
- *$X_{n,r}$ . . . RV for size of r-fold "old path"-reduced tree with originally n nodes*

Then the expected size of the reduced tree and the corresponding variance are

$$\mathbb{E}X_{n,r} = \frac{2n}{r+2} - \frac{r(r+1)}{3(r+2)} + O(n^{-1}),$$

$$\mathbb{V}X_{n,r} = \frac{2r(r+1)}{3(r+2)^2}n + O(1).$$

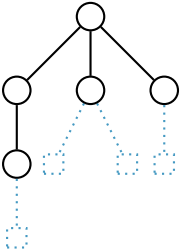
Summary

Leaves

$\mathbb{E} \sim \frac{n}{r+1}$

$\mathbb{V} \sim \frac{r(r+2)}{6(r+1)^2} n$

limit law: ✓



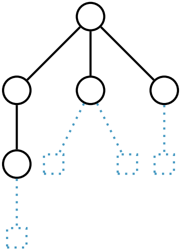
Summary

Leaves

$\mathbb{E} \sim \frac{n}{r+1}$

$\mathbb{V} \sim \frac{r(r+2)}{6(r+1)^2} n$

limit law: ✓

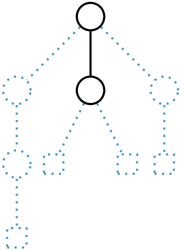


Paths

$\mathbb{E} \sim \frac{n}{2^{r+1}-1}$

$\mathbb{V} \sim \frac{2^{r+1}(2^r-1)}{3(2^{r+1}-1)^2} n$

limit law: ✓



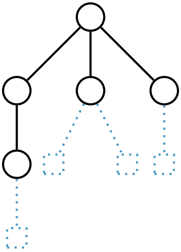
Summary

Leaves

$\mathbb{E} \sim \frac{n}{r+1}$

$\mathbb{V} \sim \frac{r(r+2)}{6(r+1)^2} n$

limit law: ✓

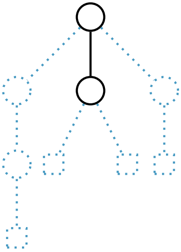


Paths

$\mathbb{E} \sim \frac{n}{2^{r+1}-1}$

$\mathbb{V} \sim \frac{2^{r+1}(2^r-1)}{3(2^{r+1}-1)^2} n$

limit law: ✓

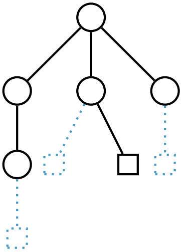


Old leaves

$\mathbb{E} \sim (2 - B_r(1/4))n$

$\mathbb{V} = \Theta(n)$

limit law: ✓



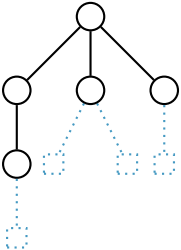
Summary

Leaves

$\mathbb{E} \sim \frac{n}{r+1}$

$\mathbb{V} \sim \frac{r(r+2)}{6(r+1)^2} n$

limit law: ✓

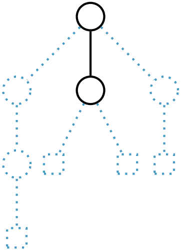


Paths

$\mathbb{E} \sim \frac{n}{2^{r+1}-1}$

$\mathbb{V} \sim \frac{2^{r+1}(2^r-1)}{3(2^{r+1}-1)^2} n$

limit law: ✓

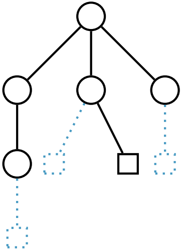


Old leaves

$\mathbb{E} \sim (2 - B_r(1/4))n$

$\mathbb{V} = \Theta(n)$

limit law: ✓



Old paths

$\mathbb{E} \sim \frac{2n}{r+2}$

$\mathbb{V} \sim \frac{2r(r+1)}{3(r+2)^2} n$

limit law: ???