Reducing Simply Generated Trees by Iterative Leaf Cutting

Joint work with Clemens Heuberger, Stephan Wagner
Example: Deterministic Plane Tree Reduction

- Remove all leaves!
Example: Deterministic Plane Tree Reduction

- Remove all leaves!
Example: Deterministic Plane Tree Reduction

- Remove all leaves!
Example: Deterministic Plane Tree Reduction

- Remove all leaves!

\[
\begin{align*}
\text{Original Tree} & \quad \Rightarrow \quad \text{Intermediate Tree} & \quad \Rightarrow \quad \text{Reduced Tree} & \quad \Rightarrow \quad \text{Final Tree}
\end{align*}
\]
Example: Deterministic Plane Tree Reduction

- Remove all leaves!

Parameter of Interest:
- Size of $r$th reduction $\leftarrow \#$ of removed nodes
Reduction → Expansion

► modelling reduction directly: not suitable
► instead: see inverse operation, growing trees
Reduction → Expansion

- modelling reduction directly: not suitable
- instead: see inverse operation, *growing trees*
Reduction $\rightarrow$ Expansion

- modelling reduction directly: not suitable
- instead: see inverse operation, **growing trees**

![Diagram of tree structures](image)
Reduction → Expansion

- modelling reduction directly: not suitable
- instead: see inverse operation, *growing trees*
Expansion operators

- $F$ ... family of plane trees; bivariate generating function $f$
- expansion operator $\Phi \Rightarrow \Phi(f)$ counts expanded trees
Expansion operators

- $F$... family of plane trees; bivariate generating function $f$
- expansion operator $\Phi \Rightarrow \Phi(f)$ counts expanded trees

Leaf expansion $\Phi$

- inverse operation to leaf reduction
Expansion operators

- $F$... family of plane trees; bivariate generating function $f$
- expansion operator $\Phi \Rightarrow \Phi(f)$ counts expanded trees

Leaf expansion $\Phi$

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

![Diagram showing leaf expansion]

$+$ $+$ $+$ $+$ $+$ ...
Expansion operators

- $F \ldots$ family of plane trees; bivariate generating function $f$
- expansion operator $\Phi \Rightarrow \Phi(f)$ counts expanded trees

Leaf expansion $\Phi$

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

\[
\begin{align*}
\square & \quad \Phi \quad \rightarrow \quad \bigoplus \quad + \\
\square & \quad \triangleq \quad t, \quad \bigcirc & \quad \triangleq \quad z
\end{align*}
\]

\[
\Phi(t) = zt + zt^2 + zt^3 + \cdots
\]
Reductions on Plane Trees

Leaves

\[ E \sim \frac{n}{r+1} \]
\[ V \sim \frac{r(r+2)}{6(r+1)^2} n \]

limit law: ✓
Reductions on Plane Trees

**Leaves**

\[ E \sim \frac{n}{r+1} \]

\[ V \sim \frac{r(r+2)}{6(r+1)^2} n \]

limit law: ✓

**Paths**

\[ E \sim \frac{n}{2r+1-1} \]

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

limit law: ✓
Reductions on Plane Trees

**Leaves**

\[ E \sim \frac{n}{r+1}, \quad V \sim \frac{r(r+2)}{6(r+1)^2} n \]

Limit law: ✓

**Paths**

\[ E \sim \frac{n}{2^{r+1}-1}, \quad V \sim \frac{2^{r+1}(2^r-1)}{3(2^{r+1}-1)^2} n \]

Limit law: ✓

**Old leaves**

\[ E \sim (2 - B_{r-1}(1/4)) n, \quad V = \Theta(n) \]

Limit law: ✓
**Reductions on Plane Trees**

### Leaves
- \( E \sim \frac{n}{r+1} \)
- \( V \sim \frac{r(r+2)}{6(r+1)^2} n \)
- Limit law: √

### Paths
- \( E \sim \frac{n}{2r^{1+1}} \)
- \( V \sim \frac{2^{r+1}(2^r-1)}{3(2^{r+1}-1)^2} n \)
- Limit law: √

### Old leaves
- \( E \sim (2 - B_{r-1}(1/4))n \)
- \( V = \Theta(n) \)
- Limit law: √

### Old paths
- \( E \sim \frac{2n}{r+2} \)
- \( V \sim \frac{2r(r+1)}{3(r+2)^2} n \)
- Limit law: √
Reductions on Plane Trees

**Leaves**
\[ E \sim \frac{n}{r+1} \]
\[ V \sim \frac{r(r+2)}{6(r+1)^2} n \]
limit law: ✓

**Paths**
\[ E \sim \frac{n}{2r+1-1} \]
\[ V \sim \frac{2r+1(2r-1)}{3(2r+1-1)^2} n \]
limit law: ✓

**Old leaves**
\[ E \sim (2 - B_{r-1}(1/4)) n \]
\[ V = \Theta(n) \]
limit law: ✓

**Old paths**
\[ E \sim \frac{2n}{r+2} \]
\[ V \sim \frac{2r(r+1)}{3(r+2)^2} n \]
limit law: ✓

Question
What can be done for other tree classes?
Additive Tree Parameters

$\mathcal{T}$ ... rooted trees. $F: \mathcal{T} \rightarrow \mathbb{R} ...$ additive tree parameter, if

- $\tau \in \mathcal{T}$ tree; $\tau_1, \tau_2, \ldots, \tau_k$ branches of $\tau$
Additive Tree Parameters

\(\mathcal{T}\ldots\) rooted trees. \(F: \mathcal{T} \rightarrow \mathbb{R}\ldots\) additive tree parameter, if

- \(\tau \in \mathcal{T}\) tree; \(\tau_1, \tau_2, \ldots, \tau_k\) branches of \(\tau\)
- \(F(\tau) = F(\tau_1) + F(\tau_2) + \cdots + F(\tau_k) + f(\tau)\),
  - with toll function \(f: \mathcal{T} \rightarrow \mathbb{R}\)
Additive Tree Parameters

\( T \) ... rooted trees. \( F : T \to \mathbb{R} \) ... additve tree parameter, if

- \( \tau \in T \) tree; \( \tau_1, \tau_2, \ldots, \tau_k \) branches of \( \tau \)
- \( F(\tau) = F(\tau_1) + F(\tau_2) + \cdots + F(\tau_k) + f(\tau) \)
  - with toll function \( f : T \to \mathbb{R} \)
Additive Tree Parameters

$\mathcal{T}$... rooted trees. $F: \mathcal{T} \rightarrow \mathbb{R}$ ... additive tree parameter, if

- $\tau \in \mathcal{T}$ tree; $\tau_1, \tau_2, \ldots, \tau_k$ branches of $\tau$
- $F(\tau) = F(\tau_1) + F(\tau_2) + \cdots + F(\tau_k) + f(\tau)$,
  - with toll function $f: \mathcal{T} \rightarrow \mathbb{R}$
Additive Tree Parameters

\( \mathcal{T} \ldots \) rooted trees. \( F : \mathcal{T} \to \mathbb{R} \) \ldots additive tree parameter, if

- \( \tau \in \mathcal{T} \) tree; \( \tau_1, \tau_2, \ldots, \tau_k \) branches of \( \tau \)  
- \( F(\tau) = F(\tau_1) + F(\tau_2) + \cdots + F(\tau_k) + f(\tau) \),
  - with toll function \( f : \mathcal{T} \to \mathbb{R} \)
Additive Tree Parameters

...rooted trees. $F: \mathcal{T} \rightarrow \mathbb{R}$ ...additive tree parameter, if

- $\tau \in \mathcal{T}$ tree; $\tau_1, \tau_2, \ldots, \tau_k$ branches of $\tau$
- $F(\tau) = F(\tau_1) + F(\tau_2) + \cdots + F(\tau_k) + f(\tau)$,
  with toll function $f: \mathcal{T} \rightarrow \mathbb{R}$
Additive Tree Parameters

\[ \mathcal{T} \ldots \text{rooted trees. } F : \mathcal{T} \to \mathbb{R} \ldots \text{additive tree parameter, if} \]

\[ \begin{align*}
&\quad \tau \in \mathcal{T} \text{ tree; } \tau_1, \tau_2, \ldots, \tau_k \text{ branches of } \tau \\
&\quad F(\tau) = F(\tau_1) + F(\tau_2) + \cdots + F(\tau_k) + f(\tau), \\
&\quad \text{with toll function } f : \mathcal{T} \to \mathbb{R}
\end{align*} \]
Additive Tree Parameters

\( T \) \ldots rooted trees. \( F: T \rightarrow \mathbb{R} \) \ldots additive tree parameter, if

- \( T \in T \) tree; \( \tau_1, \tau_2, \ldots, \tau_k \) branches of \( \tau \)
- \( F(\tau) = F(\tau_1) + F(\tau_2) + \cdots + F(\tau_k) + f(\tau) \),
  - with toll function \( f: T \rightarrow \mathbb{R} \)

- Wagner (2015), Janson (2016), Wagner et al. (2018)\ldots:
  - \( \tau_n \) random tree, size \( n \); \( f \) suitable
    \( \sim F(\tau_n) \) asymptotically Gaussian
Example: Removed Leaves

\[ a_r(\tau) \sim \# \text{ of removed nodes when cutting leaves } r \text{ times} \]
Example: Removed Leaves

- $a_r(\tau) \sim \# \text{ of removed nodes when cutting leaves } r \text{ times}$
- **Example:** $r = 3$

![Tree Diagram](image.png)
Example: Removed Leaves

- $a_r(\tau) \sim \# \text{ of removed nodes when cutting leaves } r \text{ times}$
- **Example:** $r = 3$
Example: Removed Leaves

- $a_r(\tau) \sim \# \text{ of removed nodes when cutting leaves } r \text{ times}$
- **Example:** $r = 3$
Example: Removed Leaves

- \( a_r(\tau) \sim \# \text{ of removed nodes when cutting leaves } r \text{ times} \)
- **Example:** \( r = 3 \)
Example: Removed Leaves

- $a_r(\tau) \sim \#$ of removed nodes when cutting leaves $r$ times
- **Example:** $r = 3$

```
0 if \( \tau \) has height \( \geq r \),
1 if \( \tau \) has height \( < r \).
```

“Cutting Leaves” – Toll Function
Simply Generated Trees

Plane Trees:

- *rooted*: designated root node
- *ordered*: order of children matters
Simply Generated Trees

Plane Trees:

- **rooted**: designated root node
- **ordered**: order of children matters

Generalization:

- **weight sequence**: \( w = (w_k)_{k \geq 0} \subseteq \mathbb{R}_{>0} \)
Simply Generated Trees

**Plane Trees:**

- *rooted:* designated root node
- *ordered:* order of children matters

**Generalization:**

- *weight sequence:* \( w = (w_k)_{k \geq 0} \subseteq \mathbb{R}_{\geq 0} \)
- *weight of tree* \( \tau \):

\[
w(\tau) := \prod_{\nu \text{ node in } \tau} w(\# \text{ children of } \nu)
\]
Simply Generated Trees

Plane Trees:

- **rooted**: designated root node
- **ordered**: order of children matters

Generalization:

- **weight sequence**: \( w = (w_k)_{k \geq 0} \subseteq \mathbb{R}_{\geq 0} \)
- **weight of tree** \( \tau \):

\[
w(\tau) := \prod_{\nu \text{ node in } \tau} w(\# \text{ children of } \nu)
\]

\[
w(\tau) = w_0^4 w_1 w_2 w_3
\]
Simply Generated Trees

Plane Trees:
- *rooted*: designated root node
- *ordered*: order of children matters

Generalization:
- *weight sequence*: \( w = (w_k)_{k \geq 0} \subseteq \mathbb{R}_{\geq 0} \)
- *weight of tree* \( \tau \):
  \[
  w(\tau) := \prod_{\nu \text{ node in } \tau} w(\# \text{ children of } \nu)
  \]
  \[
  w(\tau) = w_0^4 w_1 w_2 w_3
  \]
- *partition function* and *probability distribution*:
  \[
  Z_n = Z_n(w) := \sum_{\tau \in \mathcal{T}_n} w(\tau) \quad \Rightarrow \quad P(\tau_n = \tau) = \frac{w(\tau)}{Z_n}
  \]
Simply Generated Trees: Examples

Plane Trees

\[ w_0 = w_1 = \cdots = 1 \]
Simply Generated Trees: Examples

**Plane Trees**

\[ w_0 = w_1 = \cdots = 1 \]

**Unary-Binary Trees**

\[ w_0 = w_1 = w_2 = 1 \]
Simply Generated Trees: Examples

**Plane Trees**

```
w_0 = w_1 = \cdots = 1
```

**Unary-Binary Trees**

```
w_0 = w_1 = w_2 = 1
```

**d-ary Trees**

```
w_0 = w_d = 1
```
Simply Generated Trees: Examples

**Plane Trees**

\[ w_0 = w_1 = \cdots = 1 \]

**Unary-Binary Trees**

\[ w_0 = w_1 = w_2 = 1 \]

**d-ary Trees**

\[ w_0 = w_d = 1 \]

**k-colored leaves**

\[ w_0 = k, \ w_1 = \cdots = 1 \]
Recursive Characterization

Proposition

- $T(z, u) \ldots \text{BWGF} (z \ldots \text{tree size, } u \ldots \text{removed nodes})$
Recursive Characterization

**Proposition**

- $T(z, u) \ldots \text{BWGF (z \ldots \text{tree size}, u \ldots \text{removed nodes})}$
- $\Phi(t) \ldots \text{GF of } w$, $\Phi(t) = \sum_{k \geq 0} w_k t^k$
Proposition

- \( T(z, u) \) \( \ldots \) BWGF (\( z \ldots \) tree size, \( u \ldots \) removed nodes)
- \( \Phi(t) \) \( \ldots \) GF of \( w \), \( \Phi(t) = \sum_{k \geq 0} w_k t^k \)
- \( T_r(z) \) \( \ldots \) WGF of trees of height \( < r \)
Recursive Characterization

**Proposition**

- $T(z, u)\ldots$ BWGF ($z\ldots$ tree size, $u\ldots$ removed nodes)
- $\Phi(t)\ldots$ GF of $w$, $\Phi(t) = \sum_{k \geq 0} w_k t^k$
- $T_r(z)\ldots$ WGF of trees of height $< r$

Then:

$$T(z, u) = z \Phi(T(z, u)) + \left(1 - \frac{1}{u}\right) T_r(zu)$$
Recursive Characterization

**Proposition**

- \( T(z, u) \) \( \ldots \) BWGF (\( z \ldots \) tree size, \( u \ldots \) removed nodes)
- \( \Phi(t) \) \( \ldots \) GF of \( w \), \( \Phi(t) = \sum_{k \geq 0} w_k t^k \)
- \( T_r(z) \) \( \ldots \) WGF of trees of height \( < r \)

Then:

\[
T(z, u) = z\Phi(T(z, u)) + \left(1 - \frac{1}{u}\right)T_r(zu)
\]

**Observation.** \( T \ldots w \)-simply generated trees, GF \( T(z) = T(z, 1) \)
Recursive Characterization

**Proposition**

- $T(z, u)$... BWGF (z... tree size, u... removed nodes)
- $\Phi(t)$... GF of $w$, $\Phi(t) = \sum_{k \geq 0} w_k t^k$
- $T_r(z)$... WGF of trees of height $< r$

Then:

$$T(z, u) = z\Phi(T(z, u)) + \left(1 - \frac{1}{u}\right)T_r(zu)$$

**Observation.** $T$... $w$-simply generated trees, GF $T(z) = T(z, 1)$

\[
T = \sum_{k \geq 0, w_k \neq 0} \left( T \cdots T \right) \quad k \text{ branches}
\]

$T(z) = z \cdot \sum_{w_k \neq 0} w_k \cdot T(z)^k$
Recursive Characterization

**Proposition**

- $T(z, u) \ldots$ BWGF ($z \ldots$ tree size, $u \ldots$ removed nodes)
- $\Phi(t) \ldots$ GF of $w$, $\Phi(t) = \sum_{k \geq 0} w_k t^k$
- $T_r(z) \ldots$ WGF of trees of height $< r$

Then:

$$T(z, u) = z\Phi(T(z, u)) + \left(1 - \frac{1}{u}\right)T_r(zu)$$

**Observation.** $T \ldots$ $w$-simply generated trees, GF $T(z) = T(z, 1)$

$$T = \sum_{k \geq 0, w_k \neq 0} T \quad \Rightarrow \quad T(z) = z\Phi(T(z))$$
Motivation / Preliminaries
Simply Generated Trees
Outlook – Pólya Trees

Recursive Characterization – Proof

**Functional Equation**

\[ T(z, u) = z\Phi(T(z, u)) + \left(1 - \frac{1}{u}\right)T_r(zu) \]

**Proof.** \( T_r \) ... trees with height less than \( r \)
Recursive Characterization – Proof

Functional Equation

\[ T(z, u) = z\Phi(T(z, u)) + \left(1 - \frac{1}{u}\right)T_r(zu) \]

Proof. \( T_r \ldots \text{trees with height less than } r \)

\[ T(z, u) = \sum_{\tau \in T} w(\tau)z^{|\tau|}u^{a_r(\tau)} = \]

\[ = \sum_{k\geq 0} z \cdot \sum_{\tau_1} \sum_{\tau_2} \cdots \sum_{\tau_k} \left( \prod w(\tau_j) \right) z^{\tau_1} u^{\tau_2} \cdots u^{\tau_k} \]
Recursive Characterization – Proof

**Functional Equation**

\[ T(z, u) = z\Phi(T(z, u)) + \left(1 - \frac{1}{u}\right)T_r(zu) \]

**Proof.** \(T_r\)... trees with height less than \(r\)

\[
T(z, u) = \sum_{\tau \in T} w(\tau)z^{|\tau|}u^{a_r(\tau)}
\]

\[
= \sum_{k \geq 0} \sum_{\tau_1} \cdots \sum_{\tau_k} \left( \prod_{j=1}^{k} w(\tau_j) \right) z^{1+\sum|\tau_j|}u^{\sum a_r(\tau_j)}
\]

\[ + \sum_{\tau \in T_r} w(\tau)z^{\tau}(u^{\tau} - u^{\tau-1}) \]
Recursive Characterization – Proof

**Functional Equation**

\[
T(z, u) = z\Phi(T(z, u)) + \left(1 - \frac{1}{u}\right) T_r(zu)
\]

**Proof.** \(T_r\)…trees with height less than \(r\)

\[
T(z, u) = \sum_{\tau \in T} w(\tau) z^{|\tau|} u^{a_r(\tau)}
\]

\[
= \sum_{k \geq 0} w_k \sum_{\tau_1} \cdots \sum_{\tau_k} \left( \prod_{j=1}^{k} w(\tau_j) \right) z^{1+\sum |\tau_j|} u^{\sum a_r(\tau_j)}
\]

\[
+ \sum_{\tau \in T_r} w(\tau) z^{|\tau|} (u^{|\tau|} - u^{|\tau|-1})
\]

\[
= \cdots = z\Phi(T(z, u)) + \left(1 - \frac{1}{u}\right) T_r(zu).
\]
Singular Expansion

**Goal:** extract information from functional equation
Singular Expansion

- **Goal:** extract information from functional equation
- In particular: partial derivatives $\partial_u T(z, 1)$, $\partial_{uu} T(z, 1)$
Singular Expansion

**Goal:** extract information from functional equation

- In particular: partial derivatives $\partial_u T(z,1)$, $\partial_{uu} T(z,1)$
- Via implicit differentiation:

$$\partial_u T(z,1) = \frac{zT'(z)T_r(z)}{T(z)}$$
Singular Expansion

- **Goal**: extract information from functional equation
- In particular: partial derivatives $\partial_u T(z, 1)$, $\partial_{uu} T(z, 1)$
- Via implicit differentiation:
  \[
  \partial_u T(z, 1) = \frac{zT'(z)T_r(z)}{T(z)}
  \]
- $T(z)$ satisfies $T(z) = z\Phi(T(z)) \leadsto$ singular inversion!
Singular Expansion

- **Goal:** extract information from functional equation
- In particular: partial derivatives $\partial_u \, T(z, 1), \; \partial_{uu} \, T(z, 1)$
- Via implicit differentiation:
  \[
  \partial_u \, T(z, 1) = \frac{z T'(z) T_r(z)}{T(z)}
  \]
- $T(z)$ satisfies $T(z) = z \Phi(T(z)) \leadsto$ singular inversion!
  - *fundamental constant* $\tau > 0$: solution of $\tau \Phi'(\tau) - \Phi(\tau) = 0$
Singular Expansion

- **Goal:** extract information from functional equation
- In particular: partial derivatives $\partial_u T(z, 1), \partial_{uu} T(z, 1)$
- Via implicit differentiation:

$$\partial_u T(z, 1) = \frac{z T'(z) T_r(z)}{T(z)}$$

- $T(z)$ satisfies $T(z) = z \Phi(T(z)) \leadsto$ singular inversion!
  - *fundamental constant* $\tau > 0$: solution of $\tau \Phi'() - \Phi(\tau) = 0$
  - $T$ has square root singularity at $\rho = \tau / \Phi(\tau)$
Singular Expansion

- **Goal:** extract information from functional equation
- In particular: partial derivatives $\partial_u T(z, 1)$, $\partial_{uu} T(z, 1)$
- Via implicit differentiation:
  \[
  \partial_u T(z, 1) = \frac{zT'(z)Tr(z)}{T(z)}
  \]
- $T(z)$ satisfies $T(z) = z\Phi(T(z)) \leadsto$ singular inversion!
  - *fundamental constant* $\tau > 0$: solution of $\tau \Phi'(\tau) - \Phi(\tau) = 0$
  - $T$ has square root singularity at $\rho = \tau/\Phi(\tau)$
  \[
  T(z) \underset{z \rightarrow \rho}{\equiv} \tau - \sqrt{\frac{2\tau}{\rho\Phi''(\tau)}} \sqrt{1 - z/\rho} + O(1 - z/\rho)
  \]
Result – Simply Generated Trees

**Theorem (H.–Heuberger–Wagner, ’18+)**

- weight sequence \( w \), fundamental constant \( \tau \), \( \rho = \tau / \Phi(\tau) \)
- \( X_{n,r} \ldots \) # of removed nodes after cutting leaves \( r \) times

Then:
Result – Simply Generated Trees

Theorem (H.–Heuberger–Wagner, ’18+)

- weight sequence \( w \), fundamental constant \( \tau \), \( \rho = \tau / \Phi(\tau) \)
- \( X_{n,r} \ldots \) # of removed nodes after cutting leaves \( r \) times

Then:

- \( X_{n,r} \) is asymptotically normally distributed for \( n \to \infty \)
Result – Simply Generated Trees

Theorem (H.–Heuberger–Wagner, ’18+)

- weight sequence $w$, fundamental constant $\tau$, $\rho = \tau / \Phi(\tau)$
- $X_{n,r}$ is the number of removed nodes after cutting leaves $r$ times

Then:

- $X_{n,r}$ is asymptotically normally distributed for $n \to \infty$ with mean and variance
  $$\mathbb{E} X_{n,r} = \frac{T_r(\rho)}{\tau} \cdot n + O(1), \quad \forall X_{n,r} = \sigma_r^2 \cdot n + O(1)$$
Theorem (H.–Heuberger–Wagner, ’18+)

- weight sequence $w$, fundamental constant $\tau$, $\rho = \tau / \Phi(\tau)$
- $X_{n,r} \ldots \#$ of removed nodes after cutting leaves $r$ times

Then:
- $X_{n,r}$ is asymptotically normally distributed for $n \to \infty$ with mean and variance
  \[
  \mathbb{E}X_{n,r} = \frac{T_r(\rho)}{\tau} \cdot n + O(1), \quad \forall X_{n,r} = \sigma_r^2 \cdot n + O(1)
  \]
- and for $r \to \infty$ we have
  \[
  F_r(\rho) = 1 - \frac{2}{\rho \tau \Phi''(\tau)} r^{-1} + o(r^{-1}), \quad \sigma_r^2 = \frac{1}{3 \rho \tau \Phi''(\tau)} + o(1).
  \]
Theorem (H.–Heuberger–Wagner, ’18+)

- weight sequence \( w \), fundamental constant \( \tau, \rho = \tau / \Phi(\tau) \)
- \( X_{n,r} \ldots \) # of removed nodes after cutting leaves \( r \) times

Then:

- \( X_{n,r} \) is asymptotically normally distributed for \( n \to \infty \) with mean and variance

\[
\mathbb{E}X_{n,r} = \frac{T_r(\rho)}{\tau} \cdot n + O(1), \quad \forall X_{n,r} = \sigma_r^2 \cdot n + O(1)
\]

- and for \( r \to \infty \) we have

\[
\frac{F_r(\rho)}{\tau} = 1 - \frac{2}{\rho \tau \Phi''(\tau)} r^{-1} + o(r^{-1}), \quad \sigma_r^2 = \frac{1}{3 \rho \tau \Phi''(\tau)} + o(1).
\]

Technical Detail. \( w \) periodic \( \sim \) parity!
Pólya Trees

- Rooted trees, order of children is irrelevant!

\[
T(z) = z \exp(\sum \frac{1}{k} T(z^k))
\]
Pólya Trees

- Rooted trees, order of children is **irrelevant**!

**Proposition**

- \( T \ldots \text{Pólya trees}, T(z) \ldots \text{generating function} \)

\[
T(z) = z \exp \left( \sum_{k \geq 1} \frac{1}{k} T(z^k) \right)
\]
Recursive Characterization / Functional Equation

**Proposition**

- $T(z, u) \ldots BGF (z \ldots tree\ size, u \ldots removed\ nodes)$
Proposition

- $T(z, u)$... BGF ($z$... tree size, $u$... removed nodes)
- $T_r(z)$... GF of trees of height $< r$
Recursion Characterization / Functional Equation

**Proposition**

- \( T(z, u) \) \( \ldots \) BGF (\( z \ldots \) tree size, \( u \ldots \) removed nodes)
- \( T_r(z) \) \( \ldots \) GF of trees of height \( < r \)

Then:

\[
T(z, u) = z \exp \left( \sum_{k \geq 1} \frac{1}{k} T(z^k, u^k) \right) + \left( 1 - \frac{1}{u} \right) T_r(zu)
\]
## Recursive Characterization / Functional Equation

### Proposition

- \( T(z, u) \) ... BGF (\( z \)... tree size, \( u \)... removed nodes)
- \( T_r(z) \) ... GF of trees of height < \( r \)

Then:

\[
T(z, u) = z \exp \left( \sum_{k \geq 1} \frac{1}{k} T(z^k, u^k) \right) + \left( 1 - \frac{1}{u} \right) T_r(zu)
\]

### Key Observations.

- \( R(z, u) := \sum_{k \geq 2} \frac{1}{k} T(z^k, u^k) \). Then: \( R(z) = R(z, 1) \) has larger RoC than \( T(z) \)
## Recursive Characterization / Functional Equation

### Proposition

- \( T(z, u) \) ... BGF (\( z \)... tree size, \( u \)... removed nodes)
- \( T_r(z) \) ... GF of trees of height < \( r \)

Then:

\[
T(z, u) = z \exp \left( \sum_{k \geq 1} \frac{1}{k} T(z^k, u^k) \right) + \left( 1 - \frac{1}{u} \right) T_r(zu)
\]

### Key Observations.

- \( R(z, u) := \sum_{k \geq 2} \frac{1}{k} T(z^k, u^k) \). Then: \( R(z) = R(z, 1) \) has larger RoC than \( T(z) \)
- Implicit differentiation: \( \partial_u T(z, 1) = \frac{T(z) \partial_u R(z, 1) + T_r(z)}{1 - T(z)} \)
Proposition

- $T(z, u) \ldots$ BGF ($z \ldots$ tree size, $u \ldots$ removed nodes)
- $T_r(z) \ldots$ GF of trees of height $< r$

Then:

$$T(z, u) = z \exp \left( \sum_{k \geq 1} \frac{1}{k} T(z^k, u^k) \right) + \left( 1 - \frac{1}{u} \right) T_r(zu)$$

Key Observations.

- $R(z, u) := \sum_{k \geq 2} \frac{1}{k} T(z^k, u^k)$. Then: $R(z) = R(z, 1)$ has larger RoC than $T(z)$
- implicit differentiation: $\partial_u T(z, 1) = \frac{T(z) \partial_u R(z, 1) + T_r(z)}{1 - T(z)}$
- $T(z) \rightsquigarrow$ dominant singularity for $z \to \rho \approx 0.338322$
Result – Pólya Trees

Theorem (H.–Heuberger–Wagner, ’18+)

\[ \rho \approx 0.338322 \ldots \text{radius of convergence of } T(z) \]

\[ X_{n,r} \ldots \# \text{ of removed nodes after cutting leaves } r \text{ times} \]

Then:
Result – Pólya Trees

**Theorem (H.–Heuberger–Wagner, ’18+)**

- $\rho \approx 0.338322 \ldots$ radius of convergence of $T(z)$
- $X_{n,r} \ldots$ # of removed nodes after cutting leaves $r$ times

**Then:**

- For $n \to \infty$, $X_{n,r}$ has **mean** and **variance**

\[
\mathbb{E}X_{n,r} = \mu_r \cdot n + O(1), \quad \nabla X_{n,r} = \sigma_r^2 \cdot n + O(1)
\]
Result – Pólya Trees

Theorem (H.–Heuberger–Wagner, ’18+)

$\rho \approx 0.338322 \ldots$ radius of convergence of $T(z)$

$X_{n,r}\ldots$ # of removed nodes after cutting leaves $r$ times

Then:

$\mathbb{E}X_{n,r} = \mu_r \cdot n + O(1), \quad \forall X_{n,r} = \sigma_r^2 \cdot n + O(1)$

$\mu_r = 1 - \frac{2}{(1 + \rho R'(\rho))} r^{-1} + o(r^{-1}), \quad \sigma_r^2 = \frac{1}{3(1 + \rho R'(\rho))} + o(1).$
Theorem (H.–Heuberger–Wagner, ’18+)

\( \rho \approx 0.338322 \ldots \text{radius of convergence of } T(z) \)

\( X_{n,r} \ldots \# \text{ of removed nodes after cutting leaves } r \text{ times} \)

Then:

\( \text{for } n \to \infty, X_{n,r} \text{ has mean and variance} \)

\[ \mathbb{E} X_{n,r} = \mu_r \cdot n + O(1), \quad \forall X_{n,r} = \sigma_r^2 \cdot n + O(1) \]

\( \text{and for } r \to \infty \text{ we have} \)

\[ \mu_r = 1 - \frac{2}{(1 + \rho R'(\rho))} r^{-1} + o(r^{-1}), \quad \sigma_r^2 = \frac{1}{3(1 + \rho R'(\rho))} + o(1). \]

Note. \( \frac{2}{1 + \rho R'(\rho)} \approx 1.644731, \quad \frac{1}{3(1 + \rho R'(\rho))} \approx 0.274122 \)