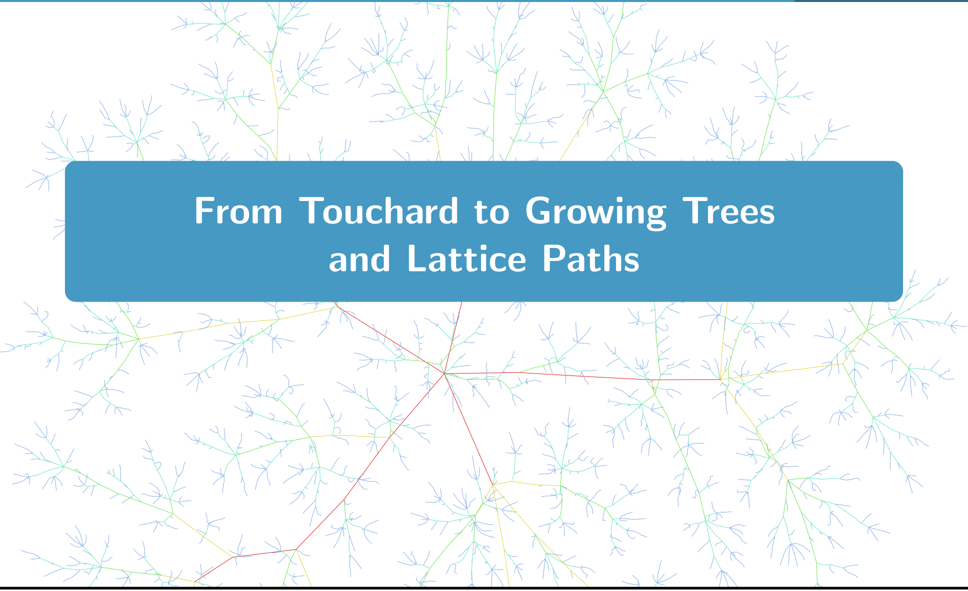
Benjamin Hackl

February 27, 2019

UNIVERSITÄT KLAGENFURT



# From Touchard to Growing Trees and Lattice Paths

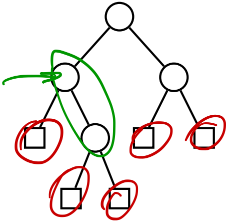


FWF
Der Wissenschaftsfonds.

# Example: Trimming Binary Trees

Cutting strategy:

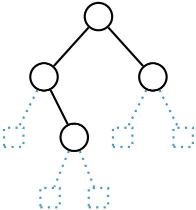
- Remove Leaves
- Merge single children with their corresponding parent



# Example: Trimming Binary Trees

Cutting strategy:

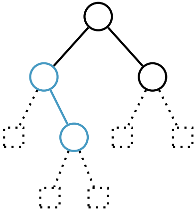
- Remove Leaves
- Merge single children with their corresponding parent



# Example: Trimming Binary Trees

Cutting strategy:

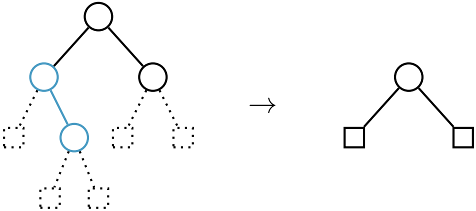
- Remove Leaves
- Merge single children with their corresponding parent



# Example: Trimming Binary Trees

Cutting strategy:

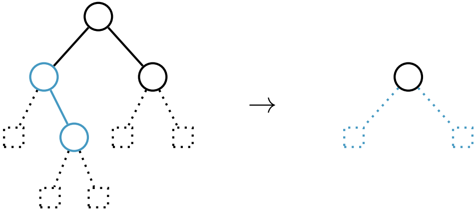
- Remove Leaves
- Merge single children with their corresponding parent



# Example: Trimming Binary Trees

Cutting strategy:

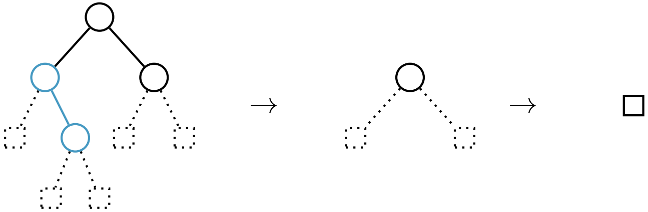
- Remove Leaves
- Merge single children with their corresponding parent



# Example: Trimming Binary Trees

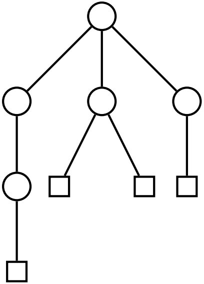
Cutting strategy:

- Remove Leaves
- Merge single children with their corresponding parent



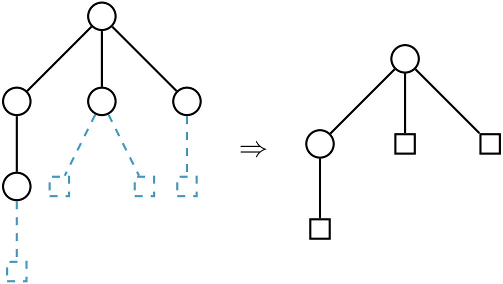
# Example: Cutting Down Plane Trees

- Remove all leaves!



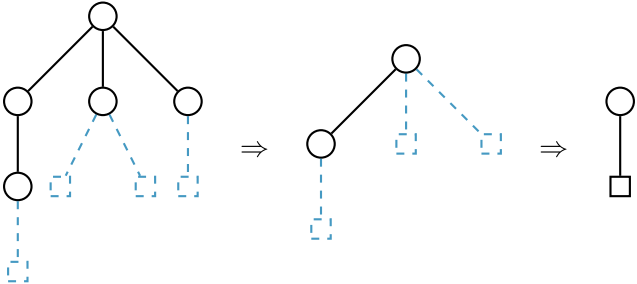
# Example: Cutting Down Plane Trees

- Remove all leaves!



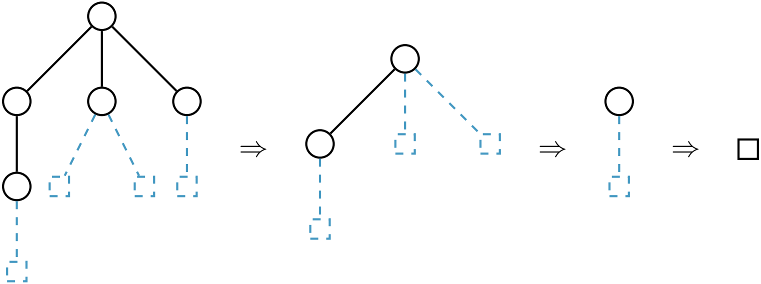
# Example: Cutting Down Plane Trees

- Remove all leaves!



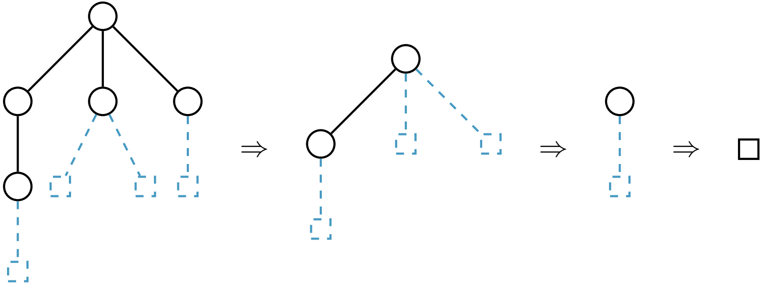
# Example: Cutting Down Plane Trees

- Remove all leaves!



# Example: Cutting Down Plane Trees

- Remove all leaves!



**Parameters of Interest:**

- Size of $r$th reduction
- Age: # of possible reductions

UNIVERSITÄT KLAGENFURT

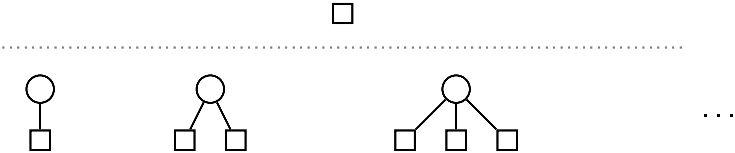
# Reduction → Expansion

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

$\square$

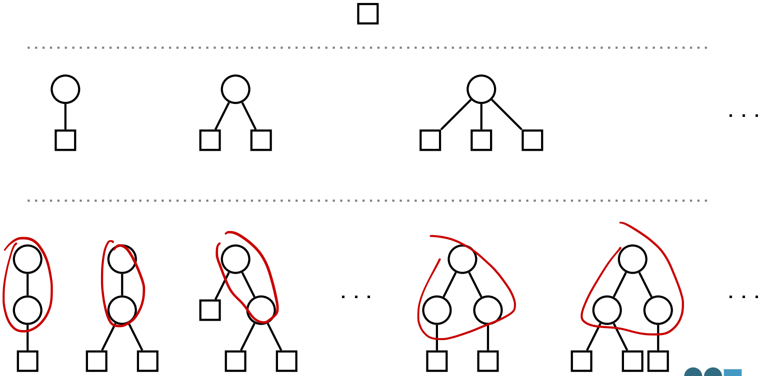
# Reduction → Expansion

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



# Reduction → Expansion

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



# Expansion operators

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

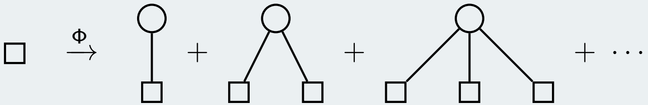
UNIVERSITÄT KLAGENFURT

# Expansion operators

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

## Leaf expansion $\Phi$

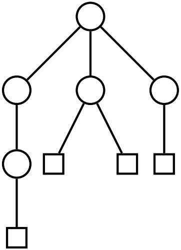
- inverse operation to leaf reduction
  - attach leaves to all current leaves (required)
  - attach leaves to inner nodes (optional)



$$\square \triangleq t, \ \bigcirc \triangleq z \quad \Rightarrow \quad \Phi(t) = zt + zt^2 + zt^3 + \cdots$$

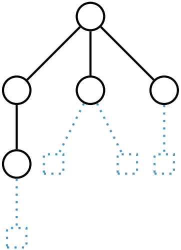
# Reductions on Plane Trees

Leaves

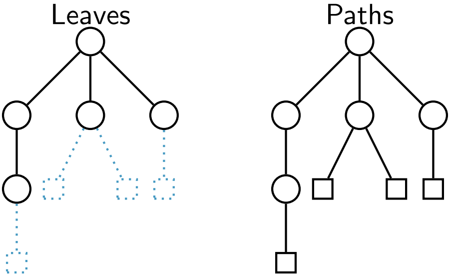


# Reductions on Plane Trees

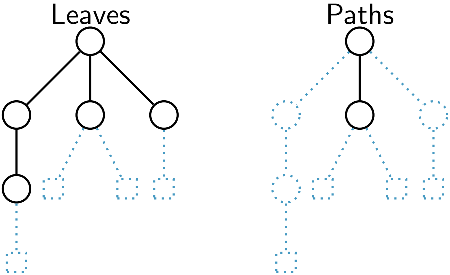
Leaves



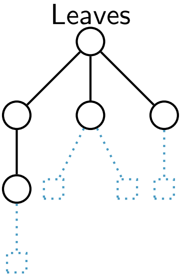
# Reductions on Plane Trees

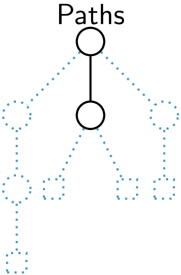


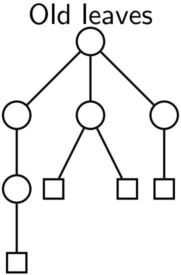
# Reductions on Plane Trees



# Reductions on Plane Trees

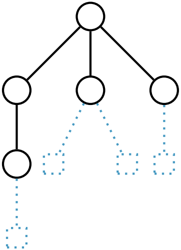


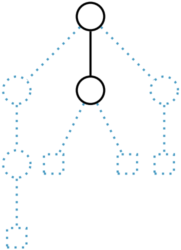



# Reductions on Plane Trees

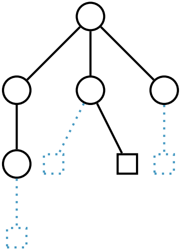
Leaves



Paths

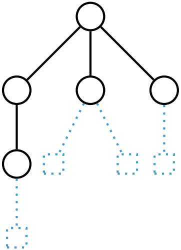


Old leaves

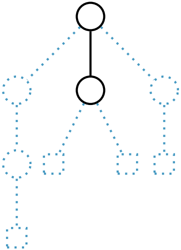


# Reductions on Plane Trees

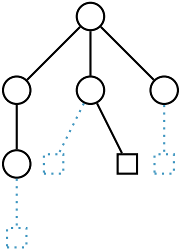
Leaves



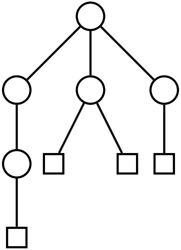
Paths



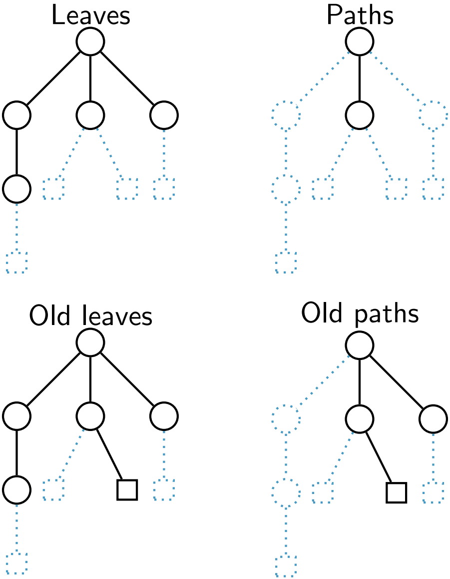
Old leaves



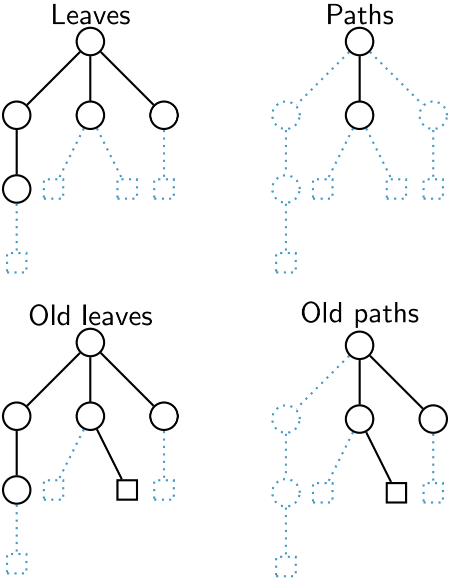
Old paths



# Reductions on Plane Trees



# Reductions on Plane Trees




**Parameters of Interest:**

- tree size after $r$ reductions
- cumulative reduction size

# Bivariate Generating Function

## Proposition

- *$\mathcal{T}$ . . . rooted plane trees*

# Bivariate Generating Function

## Proposition

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

# Bivariate Generating Function

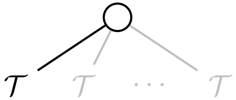
## Proposition

- $\mathcal{T}$ . . . *rooted 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

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



translates into

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

which can be solved explicitly.

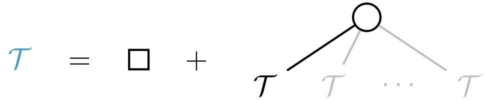
# Bivariate Generating Function

## Proposition

- $\mathcal{T}$ . . . *rooted 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.

# Bivariate Generating Function

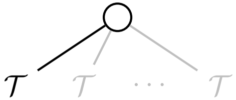
## Proposition

- $\mathcal{T}$ ... *rooted 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

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




translates into

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

which can be solved explicitly.

# Bivariate Generating Function

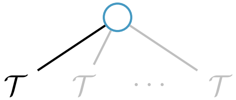
## Proposition

- $\mathcal{T}$ ... *rooted 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

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




translates into

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

which can be solved explicitly.

# Bivariate Generating Function

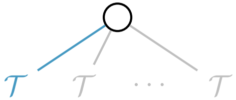
## Proposition

- *$\mathcal{T}$ … rooted 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

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




translates into

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

which can be solved explicitly.

# Bivariate Generating Function

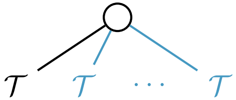
## Proposition

- $\mathcal{T}$ ... *rooted 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.

# Bivariate Generating Function

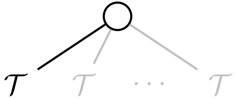
## Proposition

- $\mathcal{T}$ ... *rooted 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

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




translates into

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

which can be solved explicitly.

# Leaf expansion operator $\Phi$

**Proposition**

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

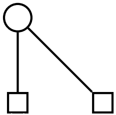
UNIVERSITÄT KLAGENFURT

# Leaf expansion operator $\Phi$

**Proposition**

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

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



- In total:

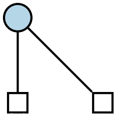
$\Phi(z^n t^k) =$

# Leaf expansion operator $\Phi$

## Proposition

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

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



- In total:

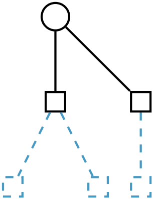
$$\Phi(z^n t^k) = z^n \cdot$$

# Leaf expansion operator Φ

## Proposition

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

- 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:

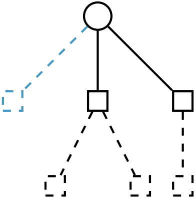
$$\Phi(z^n t^k) = z^n \cdot \left(\frac{zt}{1-t}\right)^k.$$

# Leaf expansion operator Φ

## Proposition

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

- 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:

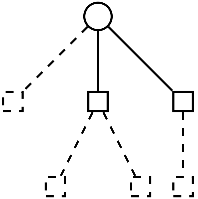
$$\Phi(z^n t^k) = z^n \cdot \left(\frac{zt}{1-t}\right)^k \cdot \frac{1}{(1-t)^{2n+k-1}}$$

# Leaf expansion operator Φ

## Proposition

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

- 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:

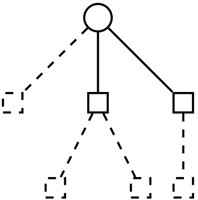
$$\Phi(z^n t^k) = z^n \cdot \left(\frac{zt}{1-t}\right)^k \cdot \frac{1}{(1-t)^{2n+k-1}} = (1-t)\left(\frac{z}{(1-t)^2}\right)^n \left(\frac{zt}{(1-t)^2}\right)^k$$

# Leaf expansion operator $\Phi$

**Proposition**

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

- 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:

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

- As $\Phi$ is linear, this proves the proposition.

# Properties of $\Phi$

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

# Properties of $\Phi$

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

$$\Phi^r(T(z,t))|_{t=z} = \frac{1-u^{r+2}}{(1-u^{r+1})(1+u)} T\Big(\frac{u(1-u^{r+1})^2}{(1-u^{r+2})^2}, \frac{u^{r+1}(1-u)^2}{(1-u^{r+2})^2}\Big)$$

# Properties of $\Phi$

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

$$\Phi^r(T(z,t))|_{t=z} = \frac{1-u^{r+2}}{(1-u^{r+1})(1+u)} T\Big(\frac{u(1-u^{r+1})^2}{(1-u^{r+2})^2}, \frac{u^{r+1}(1-u)^2}{(1-u^{r+2})^2}\Big)$$

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

# Properties of $\Phi$

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

$$\Phi^r(T(z,t))|_{t=z} = \frac{1-u^{r+2}}{(1-u^{r+1})(1+u)} T\Big(\frac{u(1-u^{r+1})^2}{(1-u^{r+2})^2}, \frac{u^{r+1}(1-u)^2}{(1-u^{r+2})^2}\Big)$$

- 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)

*After $r$ reductions of a random tree of size $n$, the remaining size $X_{n,r}$ has* **mean** *and* **variance**

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

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

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

# Cutting leaves

## Theorem (H.–Heuberger–Kropf–Prodinger)

*After $r$ reductions of a random tree of size $n$, the remaining size $X_{n,r}$ has* **mean** *and* **variance**

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

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

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

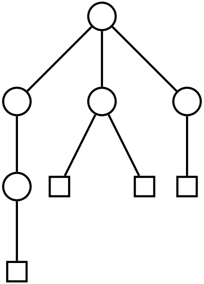
**Proof insights:**

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

UNIVERSITÄT KLAGENFURT

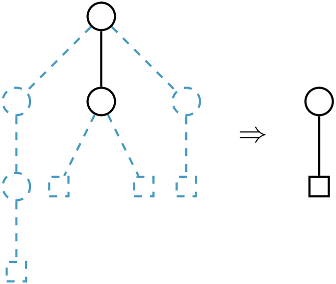
# Pruning

- Remove all paths that end in a leaf!



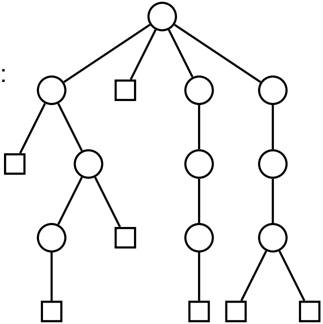
# Pruning

- Remove all paths that end in a leaf!



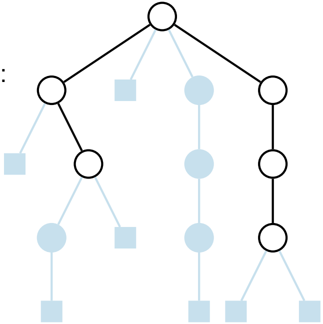
# Branches in a Tree

- Trees can be partitioned into branches:



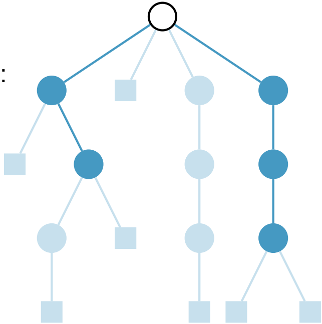
# Branches in a Tree

- Trees can be partitioned into branches:



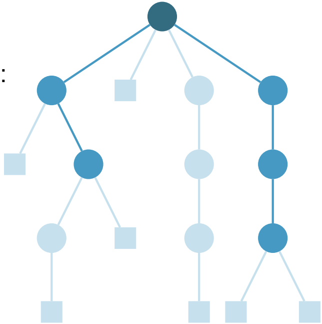
# Branches in a Tree

- Trees can be partitioned into branches:



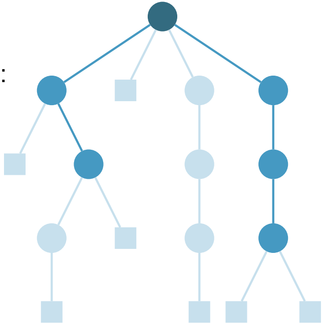
# Branches in a Tree

- Trees can be partitioned into branches:



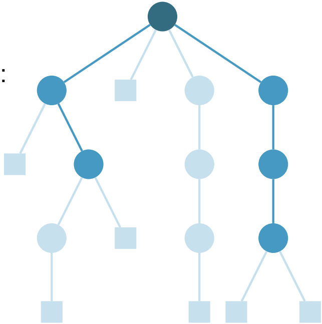
# Branches in a Tree

- Trees can be partitioned into branches:
- **Q:** How many branches are there?



# Branches in a Tree

- Trees can be partitioned into branches:
- **Q:** How many branches are there?

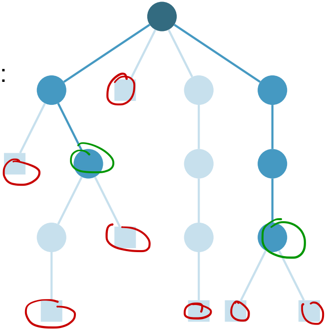


### Observation

Total # of branches $\triangleq$ # of leaves in all reduction stages

# Branches in a Tree

- Trees can be partitioned into branches:
- **Q:** How many branches are there?



## Observation

Total # of branches $\triangleq$ # of leaves in all reduction stages

**Proof:** all branches end in exactly one leaf (at some point). $\square$

# Branches in a Tree – Result

# Branches in a Tree – Result

**Theorem (H.–Heuberger–Kropf–Prodinger)**

*Average # of branches in a random plane tree of size n is*

$$\alpha n + \frac{1}{6}\log_4 n + C + \delta(\log_4 n) + O(n^{-1/4}),$$

# Branches in a Tree – Result

## Theorem (H.–Heuberger–Kropf–Prodinger)

*Average # of branches in a random plane tree of size n is*

$$\alpha n + \frac{1}{6}\log_4 n + C + \delta(\log_4 n) + O(n^{-1/4}),$$

- $\alpha = \sum_{k\geq 2} \frac{1}{2^k - 1} \approx 0.60669$,

UNIVERSITÄT KLAGENFURT

# Branches in a Tree – Result

## Theorem (H.–Heuberger–Kropf–Prodinger)

*Average # of branches in a random plane tree of size n is*

$$\alpha n + \frac{1}{6}\log_4 n + C + \delta(\log_4 n) + O(n^{-1/4}),$$

- $\alpha = \sum_{k\geq 2} \frac{1}{2^k - 1} \approx 0.60669,$
- $C = -\frac{\gamma + 4\alpha \log 2 + \log 2 + 24\zeta'(-1) + 2}{12 \log 2} \approx -0.11811,$

# Branches in a Tree – Result

## Theorem (H.–Heuberger–Kropf–Prodinger)

*Average # of branches in a random plane tree of size $n$ is*

$$\alpha n + \frac{1}{6}\log_4 n + C + \delta(\log_4 n) + O(n^{-1/4}),$$

- $\alpha = \sum_{k\geq 2} \frac{1}{2^k - 1} \approx 0.60669,$
- $C = -\frac{\gamma + 4\alpha\log 2 + \log 2 + 24\zeta'(-1) + 2}{12\log 2} \approx -0.11811,$
- $\delta$... *periodic fluctuation:*

$$\delta(x) := \frac{1}{\log 2}\sum_{k\in\mathbb{Z}\setminus\{0\}} (-1+\chi_k)\Gamma(\chi_k/2)\zeta(-1+\chi_k)e^{2k\pi i x}, \quad \chi_k = \frac{2\pi i k}{\log 2}.$$

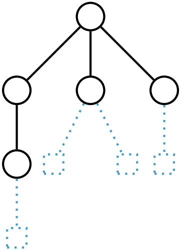
# Summary: Reductions on Plane Trees

**Leaves**

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

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

limit law: ✓



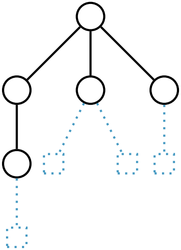
# Summary: Reductions on Plane Trees

**Leaves**

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

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

limit law: ✓

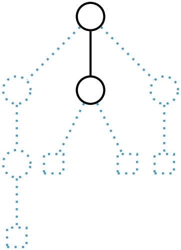


**Paths**

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

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

limit law: ✓



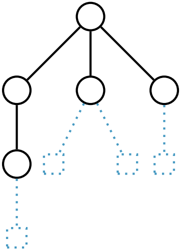
# Summary: Reductions on Plane Trees

**Leaves**

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

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

limit law: ✓

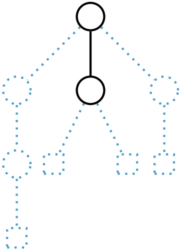


**Paths**

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

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

limit law: ✓

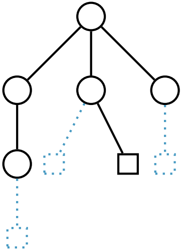


**Old leaves**

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

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

limit law: ✓



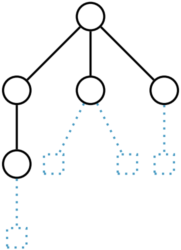
# Summary: Reductions on Plane Trees

## Leaves

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

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

limit law: ✓

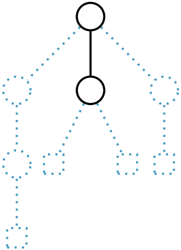


## Paths

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

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

limit law: ✓

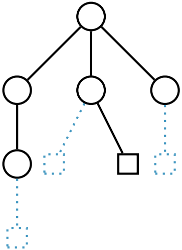


## Old leaves

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

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

limit law: ✓

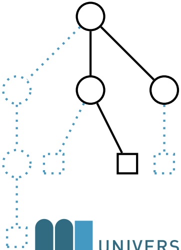


## Old paths

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

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

limit law: ✓



UNIVERSITÄT KLAGENFURT

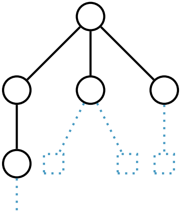
# Summary: Reductions on Plane Trees

**Leaves**

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

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

limit law: ✓

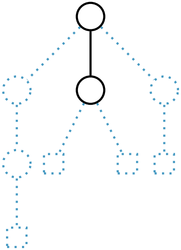


**Paths**

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

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

limit law: ✓



**Disclaimer**

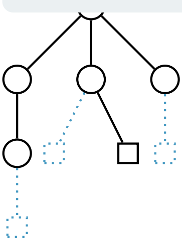
Results are **not always** that nice!

**Old leaves**

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

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

limit law: ✓

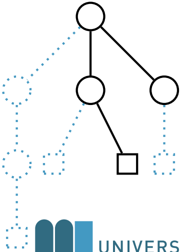


**Old paths**

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

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

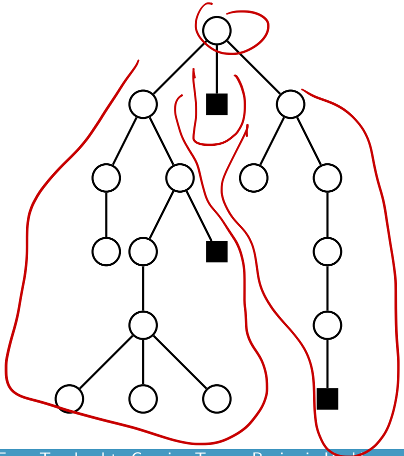
limit law: ✓



UNIVERSITÄT KLAGENFURT

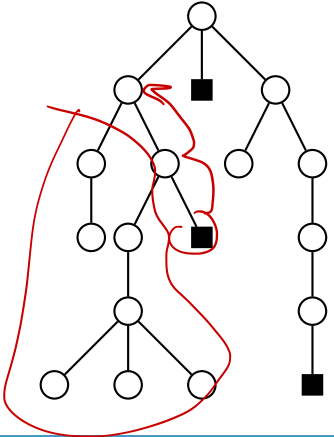
# Counterexample: Catalan–Stanley trees

- Motivation: Stanley's Catalan interpretation #26
- Rightmost leaves in all branches of root have odd distance



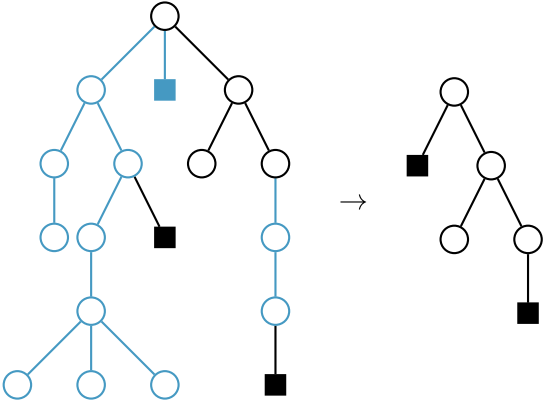
# Counterexample: Catalan–Stanley trees

- Motivation: Stanley's Catalan interpretation #26
- Rightmost leaves in all branches of root have odd distance
- **Reduction:** remove parent/grandparent (except root) of ■



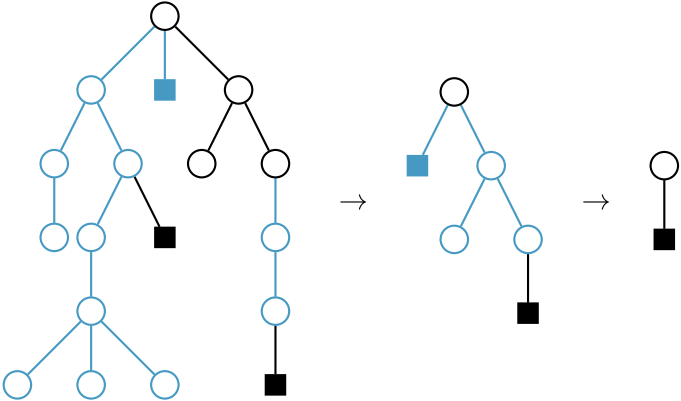
# Counterexample: Catalan–Stanley trees

- Motivation: Stanley's Catalan interpretation #26
- Rightmost leaves in all branches of root have odd distance
- **Reduction:** remove parent/grandparent (except root) of ■



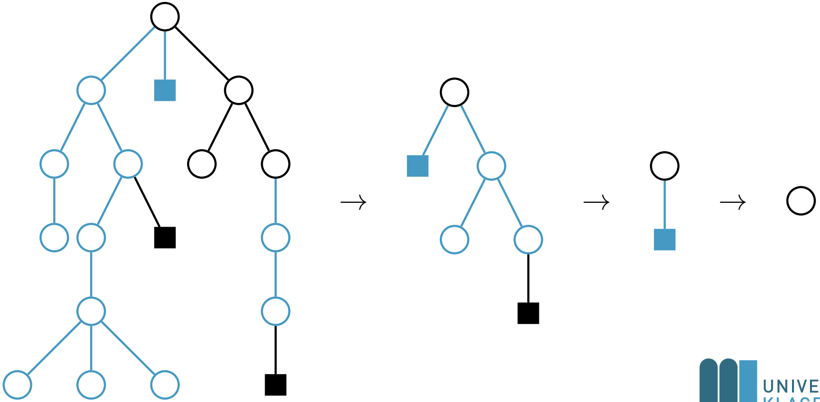
# Counterexample: Catalan–Stanley trees

- Motivation: Stanley's Catalan interpretation #26
- Rightmost leaves in all branches of root have odd distance
- **Reduction:** remove parent/grandparent (except root) of ■



# Counterexample: Catalan–Stanley trees

- Motivation: Stanley's Catalan interpretation #26
- Rightmost leaves in all branches of root have odd distance
- **Reduction:** remove parent/grandparent (except root) of ■



# Counterexample: Results

- Reduction with different parameter behavior ✓

## Age

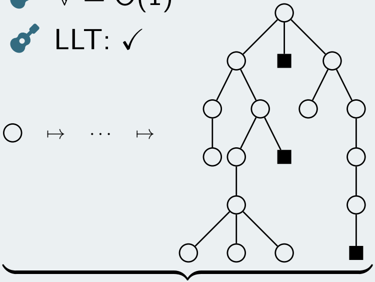
## Size of $r$th Reduction

# Counterexample: Results

- Reduction with different parameter behavior ✓

## Age

- $\mathbb{E} = \Theta(1)$
- $\mathbb{V} = \Theta(1)$
- LLT: ✓



# Generations = Age

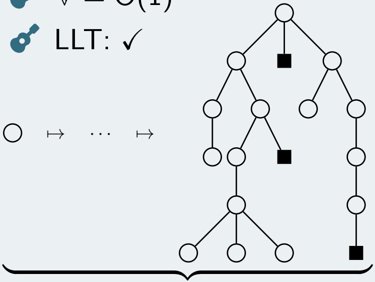
## Size of $r$th Reduction

# Counterexample: Results

- Reduction with different parameter behavior ✓

## Age

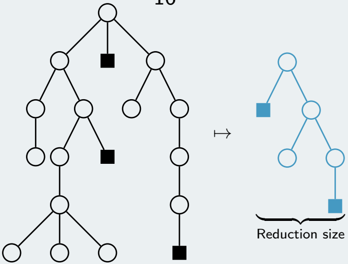
- $\mathbb{E} = \Theta(1)$
- $\mathbb{V} = \Theta(1)$
- LLT: ✓



# Generations = Age

## Size of $r$th Reduction

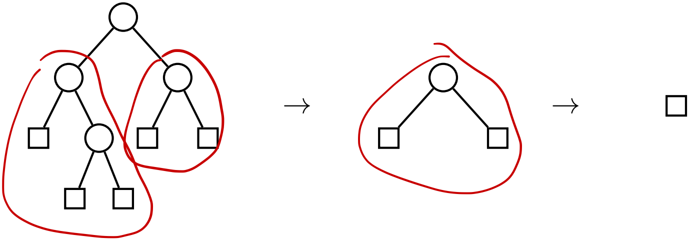
- $\mathbb{E} \sim \frac{1}{4^r} n$
- $\mathbb{V} \sim \frac{(2^r+1)(2^r-1)}{16^r} n^2$



# Trimming Binary Trees

Cutting strategy:

- Remove Leaves
- Merge single children with their corresponding parent



# Bonus: Touchard's Identity

## Proposition

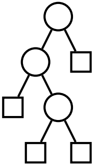
$$B(z) = 1 + \frac{z}{1-2z} B\Big(\frac{z^2}{(1-2z)^2}\Big)$$

# Bonus: Touchard's Identity

**Proposition**

$$B(z) = 1 + \frac{z}{1-2z} B\left(\frac{z^2}{(1-2z)^2}\right)$$

$$\frac{z}{1-2z} \rightsquigarrow$$

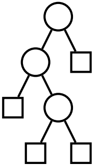


- Binary trees consist of...
  - ...just a leaf,
  - ...a "smaller" tree with all leafs replaced by "chains"

# Bonus: Touchard's Identity

## Proposition

$$B(z) = 1 + \frac{z}{1-2z} B\Big(\frac{z^2}{(1-2z)^2}\Big)$$

$\frac{z}{1-2z} \rightsquigarrow$ 

- Binary trees consist of…
  - …just a leaf,
  - …a "smaller" tree with all leafs replaced by "chains"

## Corollary (Touchard's Identity)

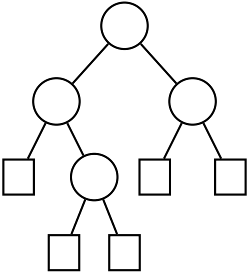
The Catalan numbers $C_n = \frac{1}{n+1}\binom{2n}{n}$ satisfy

$$C_{n+1} = \sum_{0\le k\le n/2} C_k 2^{n-2k}\binom{n}{2k}.$$

# How "old" do the nodes get?

We label the nodes according to the following rules:

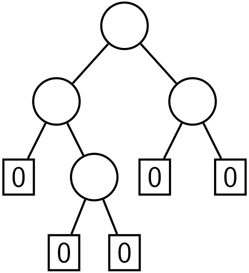
- Leaves $\to 0$
- $\text{age(left child)} = \text{age(right child)} \to \text{increase by } 1$
- Otherwise: maximum of children



# How "old" do the nodes get?

We label the nodes according to the following rules:

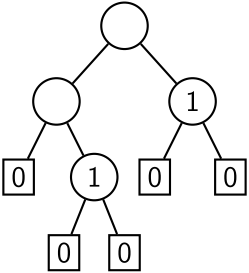
- Leaves $\to 0$
- age(left child) $=$ age(right child) $\to$ increase by 1
- Otherwise: maximum of children



# How "old" do the nodes get?

We label the nodes according to the following rules:

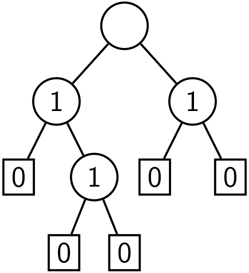
- Leaves $\rightarrow 0$
- age(left child) $=$ age(right child) $\rightarrow$ increase by 1
- Otherwise: maximum of children



# How "old" do the nodes get?

We label the nodes according to the following rules:

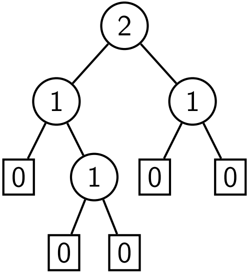
- Leaves $\to$ 0
- age(left child) $=$ age(right child) $\to$ increase by 1
- Otherwise: maximum of children



# How “old” do the nodes get?

We label the nodes according to the following rules:

- Leaves $\to 0$
- age(left child) $=$ age(right child) $\to$ increase by 1
- Otherwise: maximum of children



# The Register Function

Age $\rightsquigarrow$ *Register function* (*Horton-Strahler*-Index)

# The Register Function

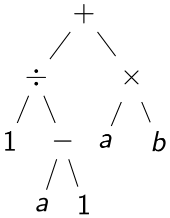
Age $\rightsquigarrow$ *Register function* (*Horton-Strahler*-Index)

- Applications:

# The Register Function

Age $\rightsquigarrow$ *Register function* (*Horton-Strahler*-Index)

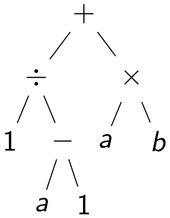
- Applications:
  - Required stack size for evaluating arithmetic expressions



# The Register Function

Age $\rightsquigarrow$ *Register function* (*Horton-Strahler*-Index)

- Applications:
  - Required stack size for evaluating arithmetic expressions
  - Branching complexity of river networks (e.g. Danube: 9)

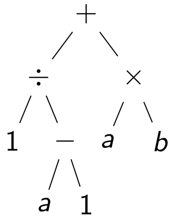



UNIVERSITÄT KLAGENFURT

# The Register Function

Age $\rightsquigarrow$ *Register function* (*Horton-Strahler*-Index)

- Applications:
  - Required stack size for evaluating arithmetic expressions
  - Branching complexity of river networks (e.g. Danube: 9)

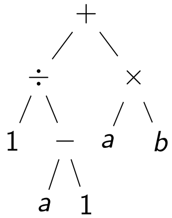


- Asymptotic analysis:

# The Register Function

Age $\rightsquigarrow$ *Register function* (*Horton-Strahler*-Index)

- Applications:
  - Required stack size for evaluating arithmetic expressions
  - Branching complexity of river networks (e.g. Danube: 9)

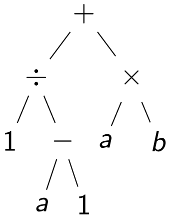



- Asymptotic analysis:
  - Flajolet, Raoult, Vuillemin (1979)

# The Register Function

Age $\rightsquigarrow$ *Register function* (*Horton-Strahler*-Index)

- Applications:
  - Required stack size for evaluating arithmetic expressions
  - Branching complexity of river networks (e.g. Danube: 9)



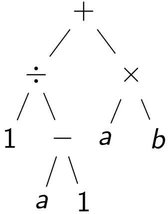

- Asymptotic analysis:
  - Flajolet, Raoult, Vuillemin (1979)
  - Flajolet, Prodinger (1986)

UNIVERSITÄT KLAGENFURT

# The Register Function

Age $\rightsquigarrow$ *Register function* (*Horton-Strahler*-Index)

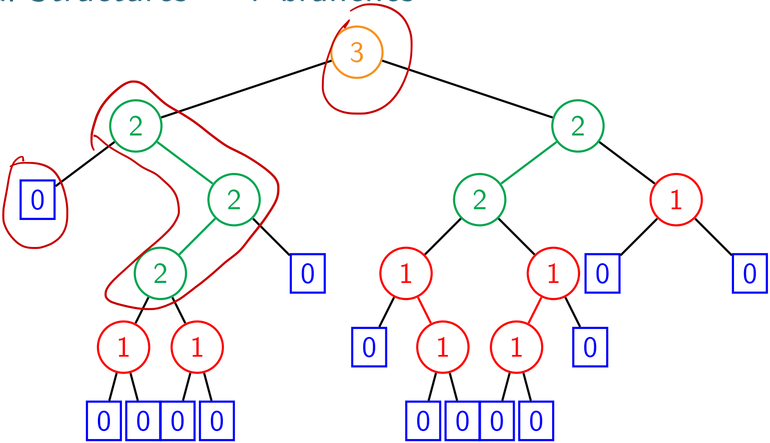
- Applications:
  - Required stack size for evaluating arithmetic expressions
  - Branching complexity of river networks (e.g. Danube: 9)



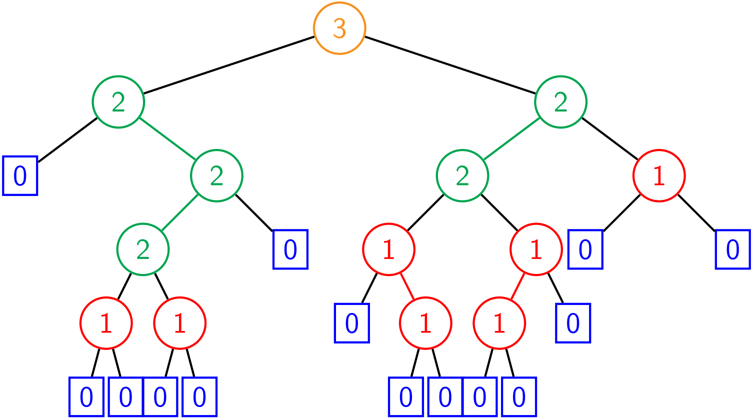

- Asymptotic analysis:
  - Flajolet, Raoult, Vuillemin (1979)
  - Flajolet, Prodinger (1986)
  - $r$-branches, Numerics: Yamamoto, Yamazaki (2009)

UNIVERSITÄT KLAGENFURT

# Local Structures – “$r$-branches”

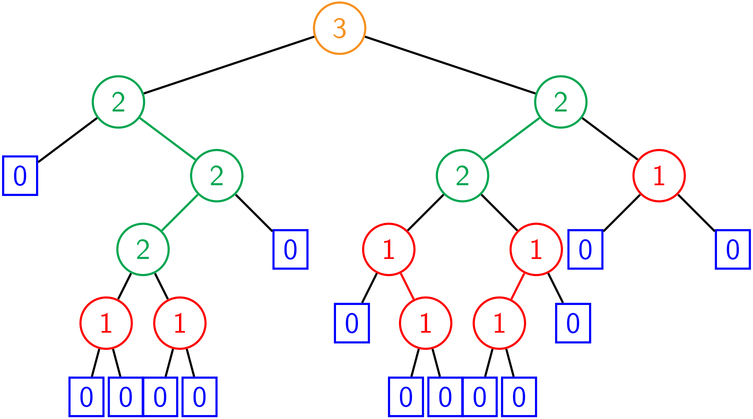


# Local Structures – “$r$-branches”




Number / Distribution of ($r$-)branches?

# Local Structures – "$r$-branches"




- Number / Distribution of ($r$-)branches?
- Example:

| $r$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| # $r$-branches | 14 | 5 | 2 | 1 |

UNIVERSITÄT KLAGENFURT

# "$r$-branches" – Results

**Theorem (H.–Heuberger–Prodinger)**

*In a random binary tree of size $n$...*

# "*r*-branches" – Results

## Theorem (H.–Heuberger–Prodinger)

*In a random binary tree of size n...*

- *# of r-branches is* **asymptotically normally distributed**

# "$r$-branches" – Results

## Theorem (H.–Heuberger–Prodinger)

*In a random binary tree of size $n$...*

- *# of $r$-branches is* **asymptotically normally distributed**
- *with* **mean** *and* **variance**

$$\mathbb{E} = \frac{n}{4^r} + \frac{1}{6}\Big(1 + \frac{5}{4^r}\Big) + O(n^{-1}), \qquad \mathbb{V} = \frac{4^r - 1}{3 \cdot 16^r} n + O(1)$$

UNIVERSITÄT KLAGENFURT

# "$r$-branches" – Results

## Theorem (H.–Heuberger–Prodinger)

*In a random binary tree of size n...*

- *# of r-branches is* **asymptotically normally distributed**
- *with* **mean** *and* **variance**

$$\mathbb{E} = \frac{n}{4^r} + \frac{1}{6}\Big(1 + \frac{5}{4^r}\Big) + O(n^{-1}), \qquad \mathbb{V} = \frac{4^r - 1}{3 \cdot 16^r} n + O(1)$$

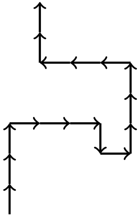
- **expected total #** *of branches is*

$$\frac{4}{3} n + \frac{1}{6} \log_4 n + C + \delta(\log_4 n) + O(n^{-1} \log n),$$

  - $C \approx 1.36190$, $\delta \ldots$ *periodic fluctuation*

# Reduction of lattice paths

Reduction of a simple, two-dimensional lattice path (i.e. a sequence of $\{\uparrow, \rightarrow, \downarrow, \leftarrow\}$):

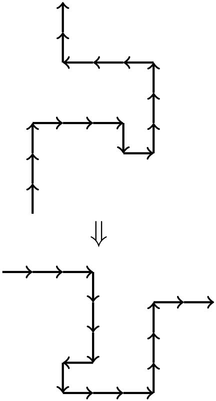


- If the path starts with $\uparrow$ or $\downarrow$: rotate it
- If the path ends with $\rightarrow$ or $\leftarrow$: rotate the last step
- Consider the pairs of horizontal-vertical segments:
  - Replace $\rightarrow \ldots \uparrow \ldots$ by $\nearrow$,
  - $\rightarrow \ldots \downarrow \ldots$ by $\searrow$,
  - $\leftarrow \ldots \downarrow \ldots$ by $\swarrow$,
  - $\leftarrow \ldots \uparrow \ldots$ by $\nwarrow$.
- Rotate the entire path again

# Reduction of lattice paths

Reduction of a simple, two-dimensional lattice path (i.e. a sequence of $\{\uparrow, \rightarrow, \downarrow, \leftarrow\}$):

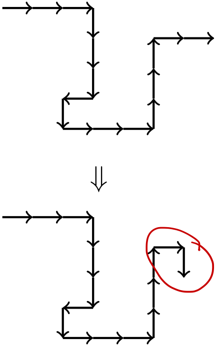
- If the path starts with $\uparrow$ or $\downarrow$: rotate it
- If the path ends with $\rightarrow$ or $\leftarrow$: rotate the last step
- Consider the pairs of horizontal-vertical segments:
  - Replace $\rightarrow \ldots \uparrow \ldots$ by $\nearrow$,
  - $\rightarrow \ldots \downarrow \ldots$ by $\searrow$,
  - $\leftarrow \ldots \downarrow \ldots$ by $\swarrow$,
  - $\leftarrow \ldots \uparrow \ldots$ by $\nwarrow$.
- Rotate the entire path again



# Reduction of lattice paths

Reduction of a simple, two-dimensional lattice path (i.e. a sequence of $\{\uparrow, \rightarrow, \downarrow, \leftarrow\}$):

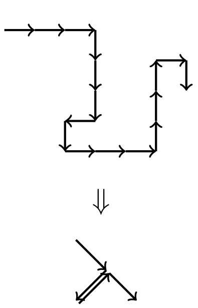
- If the path starts with $\uparrow$ or $\downarrow$: rotate it
- If the path ends with $\rightarrow$ or $\leftarrow$: rotate the last step
- Consider the pairs of horizontal-vertical segments:
  - Replace $\rightarrow \ldots \uparrow \ldots$ by $\nearrow$,
  - $\rightarrow \ldots \downarrow \ldots$ by $\searrow$,
  - $\leftarrow \ldots \downarrow \ldots$ by $\swarrow$,
  - $\leftarrow \ldots \uparrow \ldots$ by $\nwarrow$.
- Rotate the entire path again



# Reduction of lattice paths

Reduction of a simple, two-dimensional lattice path (i.e. a sequence of $\{\uparrow, \rightarrow, \downarrow, \leftarrow\}$):

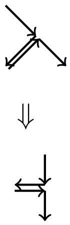
- If the path starts with $\uparrow$ or $\downarrow$: rotate it
- If the path ends with $\rightarrow$ or $\leftarrow$: rotate the last step
- Consider the pairs of horizontal-vertical segments:
  - Replace $\rightarrow \dots \uparrow \dots$ by $\nearrow$,
  - $\rightarrow \dots \downarrow \dots$ by $\searrow$,
  - $\leftarrow \dots \downarrow \dots$ by $\swarrow$,
  - $\leftarrow \dots \uparrow \dots$ by $\nwarrow$.
- Rotate the entire path again



# Reduction of lattice paths

Reduction of a simple, two-dimensional lattice path (i.e. a sequence of $\{\uparrow, \rightarrow, \downarrow, \leftarrow\}$):

- If the path starts with $\uparrow$ or $\downarrow$: rotate it
- If the path ends with $\rightarrow$ or $\leftarrow$: rotate the last step
- Consider the pairs of horizontal-vertical segments:
  - Replace $\rightarrow \dots \uparrow \dots$ by $\nearrow$,
  - $\rightarrow \dots \downarrow \dots$ by $\searrow$,
  - $\leftarrow \dots \downarrow \dots$ by $\swarrow$,
  - $\leftarrow \dots \uparrow \dots$ by $\nwarrow$.
- Rotate the entire path again

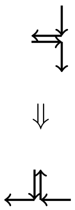


UNIVERSITÄT KLAGENFURT

## Reduction of lattice paths

Reduction of a simple, two-dimensional lattice path (i.e. a sequence of $\{\uparrow, \rightarrow, \downarrow, \leftarrow\}$):

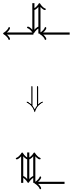
- If the path starts with $\uparrow$ or $\downarrow$: rotate it
- If the path ends with $\rightarrow$ or $\leftarrow$: rotate the last step
- Consider the pairs of horizontal-vertical segments:
  - Replace $\rightarrow \dots \uparrow \dots$ by $\nearrow$,
  - $\rightarrow \dots \downarrow \dots$ by $\searrow$,
  - $\leftarrow \dots \downarrow \dots$ by $\swarrow$,
  - $\leftarrow \dots \uparrow \dots$ by $\nwarrow$.
- Rotate the entire path again



# Reduction of lattice paths

Reduction of a simple, two-dimensional lattice path (i.e. a sequence of $\{\uparrow, \rightarrow, \downarrow, \leftarrow\}$):

- If the path starts with $\uparrow$ or $\downarrow$: rotate it
- If the path ends with $\rightarrow$ or $\leftarrow$: rotate the last step
- Consider the pairs of horizontal-vertical segments:
  - Replace $\rightarrow \ldots \uparrow \ldots$ by $\nearrow$,
  - $\rightarrow \ldots \downarrow \ldots$ by $\searrow$,
  - $\leftarrow \ldots \downarrow \ldots$ by $\swarrow$,
  - $\leftarrow \ldots \uparrow \ldots$ by $\nwarrow$.
- Rotate the entire path again



UNIVERSITÄT KLAGENFURT

# Reduction of lattice paths

Reduction of a simple, two-dimensional lattice path (i.e. a sequence of $\{\uparrow, \rightarrow, \downarrow, \leftarrow\}$):

- If the path starts with $\uparrow$ or $\downarrow$: rotate it
- If the path ends with $\rightarrow$ or $\leftarrow$: rotate the last step
- Consider the pairs of horizontal-vertical segments:
  - Replace $\rightarrow \dots \uparrow \dots$ by $\nearrow$,
  - $\rightarrow \dots \downarrow \dots$ by $\searrow$,
  - $\leftarrow \dots \downarrow \dots$ by $\swarrow$,
  - $\leftarrow \dots \uparrow \dots$ by $\nwarrow$.
- Rotate the entire path again

UNIVERSITÄT KLAGENFURT

# Reduction of lattice paths

Reduction of a simple, two-dimensional lattice path (i.e. a sequence of $\{\uparrow, \rightarrow, \downarrow, \leftarrow\}$):

- If the path starts with $\uparrow$ or $\downarrow$: rotate it
- If the path ends with $\rightarrow$ or $\leftarrow$: rotate the last step
- Consider the pairs of horizontal-vertical segments:
  - Replace $\rightarrow \ldots \uparrow \ldots$ by $\nearrow$,
  - $\rightarrow \ldots \downarrow \ldots$ by $\searrow$,
  - $\leftarrow \ldots \downarrow \ldots$ by $\swarrow$,
  - $\leftarrow \ldots \uparrow \ldots$ by $\nwarrow$.
- Rotate the entire path again

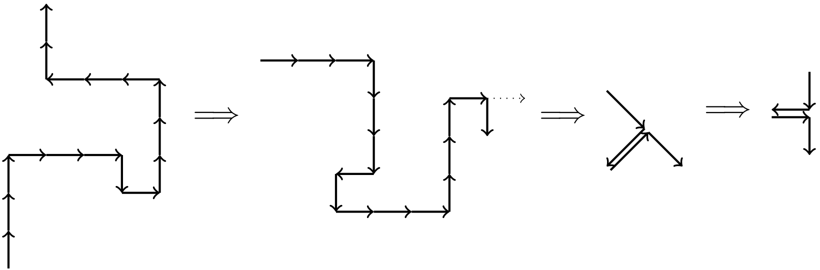
$$\nwarrow$$

$$\Downarrow$$

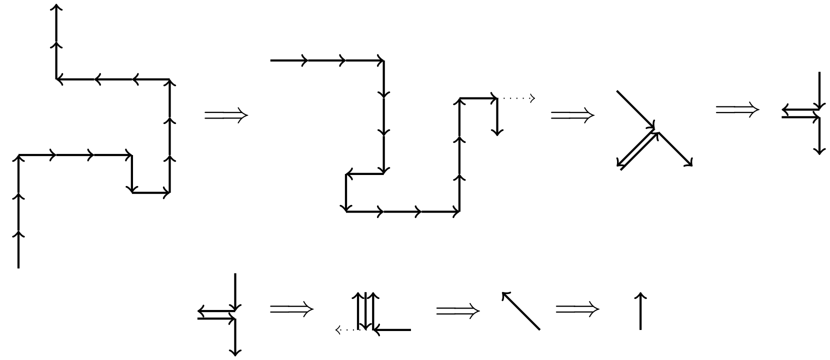
$$\uparrow$$

UNIVERSITÄT KLAGENFURT

# Reduction – Example



# Reduction – Example



# Reduction degree and functional equation

- Reduction degree: "age" of a lattice path w.r.t. reduction

# Reduction degree and functional equation

- Reduction degree: "age" of a lattice path w.r.t. reduction

## Proposition

*The generating function of simple two-dimensional lattice paths of length $\geq 1$, $L(z) = \frac{4z}{1-4z}$, satisfies the functional equation*

$$L(z) = 4z + 4L\Big(\frac{z^2}{(1-2z)^2}\Big).$$

# Reduction degree and functional equation

- Reduction degree: "age" of a lattice path w.r.t. reduction

## Proposition

*The generating function of simple two-dimensional lattice paths of length* $\geq 1$, $L(z) = \frac{4z}{1-4z}$, *satisfies the functional equation*

$$L(z) = 4z + 4L\Big(\frac{z^2}{(1-2z)^2}\Big).$$

Can be checked directly—or proven combinatorially!

# Modeling the Reduction Degree

- $L_r^{=}(z)$ ... OGF for paths with degree $r$

# Modeling the Reduction Degree

- $L_r^{=}(z)$ ... OGF for paths with degree $r$
- $X_n$ ... reduction degree of (unif. random) path, length $n$

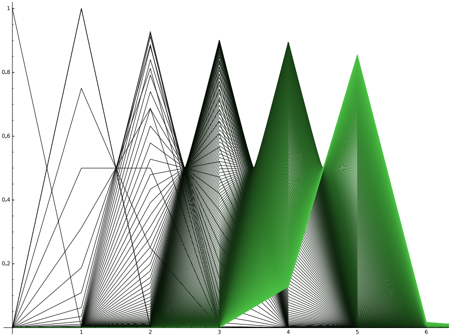
$$\Rightarrow \mathbb{P}(X_n = r) = \frac{[z^n]L_r^{=}(z)}{4^n}$$

# Modeling the Reduction Degree

- $L_r^{=}(z)$ ... OGF for paths with degree $r$
- $X_n$ ... reduction degree of (unif. random) path, length $n$

$$\Rightarrow \mathbb{P}(X_n = r) = \frac{[z^n]L_r^{=}(z)}{4^n}$$

- Probability densities of $X_1$ up to $X_{512}$:



# Reduction Degree – Expectation

- Analyze $G(z) = \sum_{r\geq 0} rL_r^{=}(z)$

# Reduction Degree – Expectation

- Analyze $G(z) = \sum_{r\geq 0} rL_r^{=}(z)$
- Substitutions $z = \frac{u}{(1+u)^2}$ and $u = e^{-t}$ yield

$$G(z) = \sum_{r,\lambda\geq 0} r4^{r+1}(-1)^{\lambda-1}\lambda e^{-t\lambda 2^r}$$

# Reduction Degree – Expectation

- Analyze $G(z) = \sum_{r\geq 0} rL_r^{=}(z)$
- Substitutions $z = \frac{u}{(1+u)^2}$ and $u = e^{-t}$ yield

$$G(z) = \sum_{r,\lambda\geq 0} r4^{r+1}(-1)^{\lambda-1}\lambda e^{-t\lambda 2^r}$$

- Mellin transform

$$G^*(s) = \Gamma(s)\zeta(s-1)\frac{2^{2-s}}{1-2^{2-s}},$$

# Reduction Degree – Expectation

- Analyze $G(z) = \sum_{r\geq 0} rL_r^{=}(z)$
- Substitutions $z = \frac{u}{(1+u)^2}$ and $u = e^{-t}$ yield

$$G(z) = \sum_{r,\lambda\geq 0} r4^{r+1}(-1)^{\lambda-1}\lambda e^{-t\lambda 2^r}$$

- Mellin transform

$$G^*(s) = \Gamma(s)\zeta(s-1)\frac{2^{2-s}}{1-2^{2-s}},$$

  - Poles: $s = 2$ (order 2), $s = 2 + \frac{2\pi i}{\log 2}k$ (order 1) for $k \in \mathbb{Z}\setminus\{0\}$

UNIVERSITÄT KLAGENFURT

# Reduction Degree – Expectation

**Theorem (H.–Heuberger–Prodinger)**

*The expected compactification degree among all simple 2D lattice paths of length n admits the asymptotic expansion*

$$\mathbb{E}X_n = \log_4 n + \frac{\gamma + 2 - 3\log 2}{2\log 2} + \delta_1(\log_4 n) + O(n^{-1})$$

*where*

$$\delta_1(x) = \frac{1}{\log 2} \sum_{k\neq 0} \frac{\Gamma(2+\chi_k)\zeta(1+\chi_k)}{\Gamma(1+\chi_k/2)} e^{2k\pi i x}$$

*is a small 1-periodic fluctuation.*

- Poles: $s = 2$ (order 2), $s = 2 + \frac{}{\log 2}k$ (order 1) for $k \in \mathbb{Z} \setminus \{0\}$

# Reduction Degree – Variance

**Theorem (H.–Heuberger–Prodinger)**

*The corresponding variance is given by*

$$\mathbb{V}X_n = \frac{\pi^2 - 24\log^2\pi - 48\zeta''(0) - 24}{24\log^2 2} - \frac{2\log\pi}{\log 2} - \frac{11}{12} + \delta_2(\log_4 n) + \frac{\gamma + 2 - 3\log 2}{\log 2}\delta_1(\log_4 n) + \delta_1^2(\log_4 n) + O\Big(\frac{1}{\log n}\Big),$$

*where $\delta_1(x)$ is defined as above and $\delta_2(x)$ is a small 1-periodic fluctuation as well.*

UNIVERSITÄT KLAGENFURT