

# Compacted binary trees, stretched exponential and asymptotic behavior of recurrences

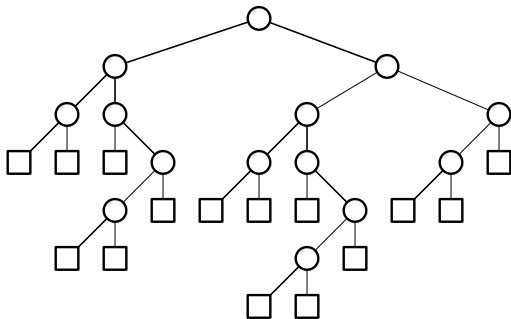
Wenjie Fang, Université Gustave Eiffel  
Joint work with Andrew Elvey Price and Michael Wallner

14 March 2024, Journées ALEA, CIRM

# What is this talk about?

- A “new method” to get asymptotic behavior of certain recurrences
- ... without generating function (*gasp!*)
- ... illustrated with compacted trees as example
- ... and some progress for generalization.

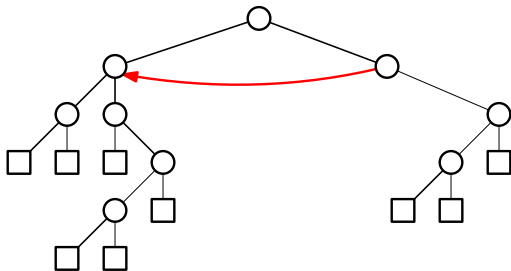
# Compacting binary trees



We try to compress a binary tree ...

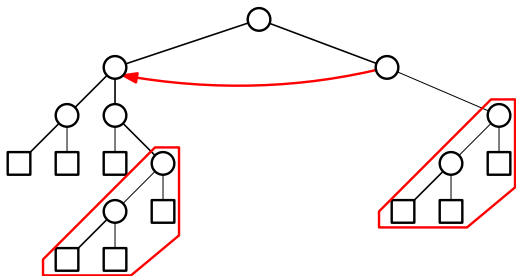


# Compacting binary trees



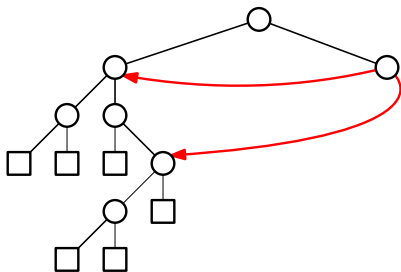
... and storing them only once ...

# Compacting binary trees



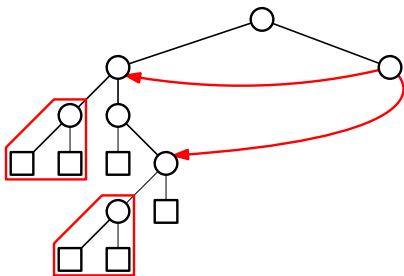
... by finding identical sub-trees ...

# Compacting binary trees



... and storing them only once ...

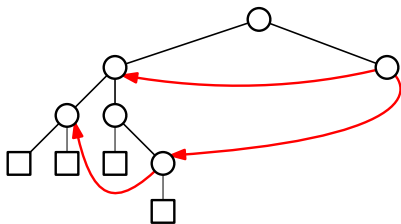
# Compacting binary trees



... by finding identical sub-trees ...

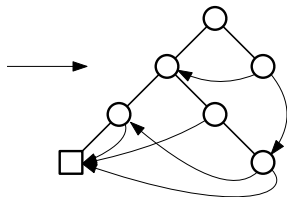
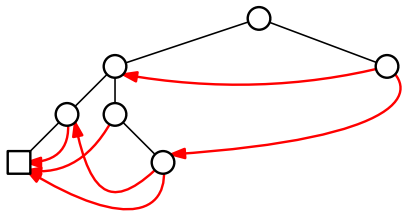


# Compacting binary trees



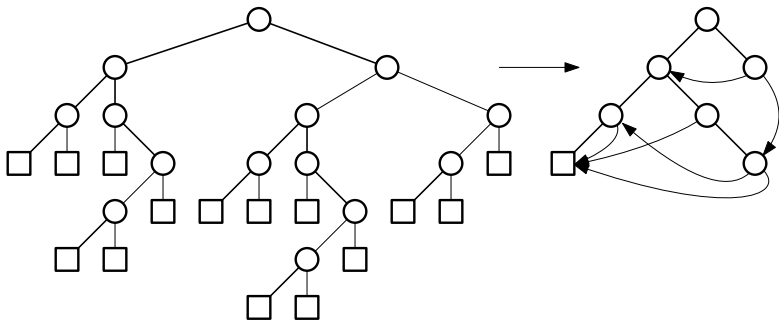
... and storing them only once ...

# Compacting binary trees



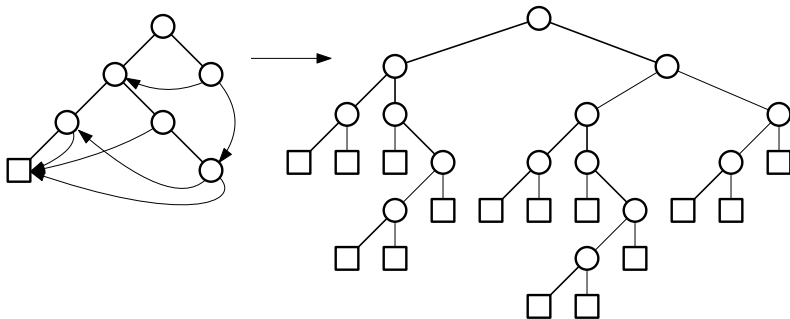
... and storing them only once ...

# Compacting binary trees



The **compacted trees** are trees with pointers obtained in this way.

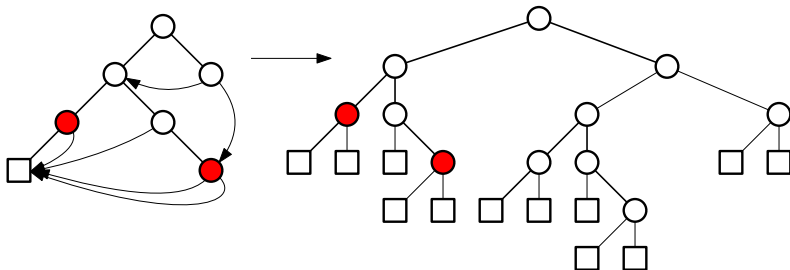
# Compacted trees



A **compacted tree** is a binary tree such that

- every leaf (except the first one) is a pointer ...
- ... towards a node preceding it in postfix order,
- and each node has a distinct “decompressed” sub-tree.

# Relaxed trees



A **relaxed tree** is a binary tree such that

- every leaf (except the first one) is a pointer ...
- ... towards a node preceding it in postfix order,
- and each node has a distinct “decompressed” sub-tree.

# What we know, and what we want to know

- (Flajolet, Sipala, Steyaert 1990)
  - Linear algorithm to “compactify” a binary tree of size  $n$
  - Average size of the compacted tree :  $O(n/\log n)$
- (Genitrini, Gittenberger, Kauers, Wallner 2019)
  - $n$  nodes, with right height  $\leq k$
  - Relaxed trees :

$$\gamma_k n! \left( 4 \cos \left( \frac{\pi}{k+3} \right) \right)^n n^{-k/2}$$

- compacted trees :

$$\gamma_k n! \left( 4 \cos \left( \frac{\pi}{k+3} \right) \right)^n n^{-\frac{k}{2} - \frac{1}{k+3} - \left( \frac{1}{4} - \frac{1}{k+3} \right) \frac{1}{\cos^2 \left( \frac{\pi}{k+3} \right)}}$$

And without any restrictions?

# Our result

- $c_n$  : the number of compacted trees with  $n$  nodes
- $r_n$  : the number of relaxed trees with  $n$  nodes

## Theorem (Elvey Price, F., Wallner 2021)

When  $n \rightarrow \infty$ , we have

$$c_n = \Theta\left(n!4^n e^{3a_1 n^{1/3}} n^{3/4}\right), \quad r_n = \Theta\left(n!4^n e^{3a_1 n^{1/3}} n\right).$$

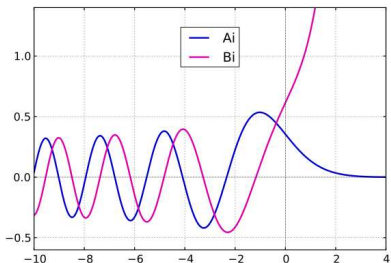
Here,  $a_1$  is the largest root of the Airy function  $\text{Ai}(x)$ , solution of  $\text{Ai}''(x) = x\text{Ai}(x)$  with  $\text{Ai}(x) \rightarrow 0$  when  $x \rightarrow +\infty$ .

We don't have the multiplicative constant !

Stretched exponential:  $e^{3a_1 n^{1/3}}$

Probability for a relaxed tree of size  $n$  to be compacted :  $\Theta(n^{-1/4})$ .

# How do we do that ?



From Geek3 at Wikimedia Commons, CC-BY 3.0

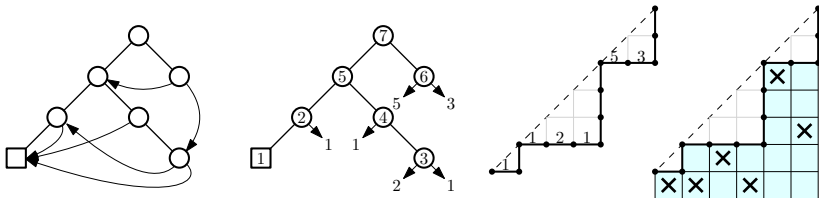
- **Bijection** with decorated Dyck paths
- **Recurrence** with two parameters
- **Heuristics** for typical behaviors
- **Truncation** of the heuristics  $\Rightarrow$  proof of the bounds

Solely based on the recurrence, the method is relatively simple.



# Encoding by decorated Dyck paths (relaxed version)

First we deal with **relaxed trees**:

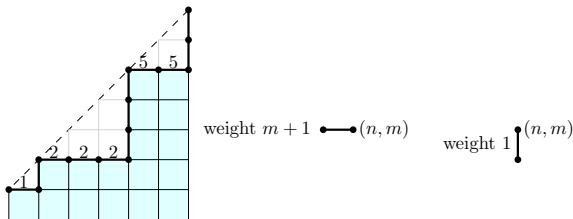


From relaxed tree to decorated Dyck paths:

- Label the nodes in **postfix** order, detach the pointers
- Draw the Dyck path :  $\rightarrow$  for pointer,  $\uparrow$  for finishing a node
- Put pointer labels on horizontal steps

# A recurrence for relaxed trees

Weight  $m + 1$  for step  $\rightarrow$  on height  $m$ .



## Proposition

Let  $r_{n,m}$  be the weighted sum of paths ending at  $(n, m)$ . Then

$$r_{n,m} = (m + 1)r_{n-1,m} + r_{n,m-1}, \quad \text{for } n \geq m \geq 1,$$

$$r_{n,m} = 0, \quad \text{for } n < m,$$

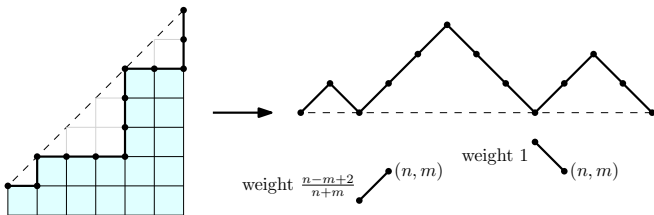
$$r_{n,0} = 1, \quad \text{for } n \geq 0.$$

The number of *relaxed trees* with  $n$  nodes is  $r_{n,n}$ .

# A transformation

Change of coordinates:  $(n, m) \rightarrow (n + m, n - m)$

We take  $d_{n+m, n-m} = r_{n, m}/n!$ , as **labeled** structure.



Recurrence :

$$d_{n, m} = \frac{n - m + 2}{n + m} d_{n-1, m-1} + d_{n-1, m+1}$$

The number of size  $n$  relaxed trees:  $r_n = n! d_{2n, 0}$ .

# Some observations

$$d_{n,m} = \frac{n-m+2}{n+m} d_{n-1,m-1} + d_{n-1,m+1}$$

Recurrence  $\Rightarrow$  diff. eq. in two variables, hard to solve.

Numerical observations :

$$d_{2n,0} = \Theta\left(4^n \rho^{n^{1/3}} n\right)$$

- $4^n$  from Dyck paths.
- Why a stretched exponential?

A higher up step has a lower weight!

# A first heuristics

Consider Dyck paths of length  $2n$  and maximal height  $\leq n^\alpha$ ,  $\alpha < 1/2$ .

## Proposition (Kousha 2012)

*A uniformly random path has height  $n^\alpha$  ( $\alpha < 1/2$ ) with probability*

$$\log(\mathbb{P}[\text{height} \leq n^\alpha]) \sim -\pi^2 n^{1-2\alpha}$$

Weight of a typical up step:

$$\frac{\Theta(n) - \Theta(n^\alpha)}{\Theta(n) + \Theta(n^\alpha)} = 1 - \Theta(n^{\alpha-1}).$$

Typically  $\Theta(n)$  such steps, thus a total weight

$$(1 - \Theta(n^{\alpha-1}))^{\Theta(n)} = \exp(-\Theta(n^\alpha)).$$

Total contribution

$$\exp(-\Theta(n^\alpha) - \Theta(n^{1-2\alpha})),$$

maximized at  $\alpha = 1/3$ , giving a **stretched exponential**  $\exp(-\Theta(n^{1/3}))$ .

# The correct scaling

Too heuristic... But this shows that the correct height is  $n^{1/3}$ !

*Ansatz*:

$$d_{n,m} \sim h(n)f(n^{-1/3}(m+1)),$$

$$s(n) = \frac{h(n)}{h(n-1)} = 2 + cn^{-2/3} + O(n^{-1}).$$

- $h(n)$  : **general growth** in  $n$ , around  $2^n \rho^{n^{1/3}}$  for some  $\rho$
- $f(x)$  : **scaling** with typical height  $n^{1/3}$

Suppose that  $m = \kappa n^{1/3} - 1$ .

*Ansatz* + recurrence :

$$f(\kappa)s(n) = \frac{n - \kappa n^{1/3} + 1}{n + \kappa n^{1/3} - 1} f\left(\frac{\kappa n^{1/3} - 2}{(n-1)^{1/3}}\right) + f\left(\frac{\kappa n^{1/3}}{(n-1)^{1/3}}\right).$$

Approximately,

$$0 = (c + 2\kappa)f(\kappa) - f''(\kappa) + O(n^{-1/3}).$$

# The first estimation

$$0 = (c + 2\kappa)f(\kappa) - f''(\kappa) + O(n^{-1/3}).$$

Roughly the equation of the [Airy function](#) !

As  $f(\kappa) \rightarrow 0$  for  $\kappa \rightarrow \infty$ , we have

$$f(\kappa) \approx b\text{Ai}\left(\frac{c + 2\kappa}{2^{2/3}}\right).$$

$$f(\kappa) \rightarrow 0 \text{ for } \kappa \rightarrow 0 \Rightarrow c = 2^{2/3}a_1.$$

Asymptotic behavior of  $\text{Ai}(x)$  near  $x \rightarrow a_1$  implies

$$r_n = n!d_{2n,0} = n!4^n \exp\left(3a_1n^{1/3} + \dots\right).$$

# Refined heuristics

*Ansatz* of order 2 :

$$d_{n,m} \sim h(n) \left( f(n^{-1/3}(m+1)) + n^{-1/3}g(n^{-1/3}(m+1)) \right),$$

$$s(n) = 2 + cn^{-2/3} + dn^{-1} + O(n^{-4/3}).$$

We get the polynomial term:

$$r_n = n!d_{2n,0} \approx n!4^n \exp\left(3a_1n^{1/3}\right)n.$$

*Ansatz* in general :

$$d_{n,m} \approx h(n) \sum_{j=0}^k f_j(n^{-1/3}(m+1))n^{-j/3},$$

$$s(n) = 2 + \gamma_2n^{-2/3} + \gamma_3n^{-1} + \dots + \gamma_kn^{-k/3} + o(n^{-k/3}).$$

A truncation suffices, but still heuristics.



# Sandwiching the asymptotics

If there are positive  $(s_n)_{n \geq 1}$  and  $(X_{n,m})_{n \geq m \geq 0}$  such that

$$X_{n,m} s_n \leq \frac{n - m + 2}{n + m} X_{n-1,m-1} + X_{n-1,m+1},$$

for all  $m$  for large enough  $n$ .

Let  $h_n = \prod_{i=1}^n s_i$ , then  $X_{n,m} h_n \leq b_0 d_{n,m}$  for some constant  $b_0$ .

Lower bound!

Reversing the inequality give an upper bound!

## Lower bound - *ansatz* and expansion

We take

$$X_{n,m} = \left(1 - \frac{2m^2}{3n} + \frac{m}{2n}\right) \text{Ai} \left( a_1 + \frac{2^{1/3}(m+1)}{n^{1/3}} \right),$$

$$s_n = 2 + \frac{2^{2/3}a_1}{n^{2/3}} + \frac{8}{3n} - \frac{1}{n^{7/6}}.$$

The difference is

$$P_{n,m} = -X_{n,m}s_n + \frac{n-m+2}{n+m}X_{n-1,m-1} + X_{n-1,m+1}.$$

Only need to prove  $P_{n,m} \geq 0$  for  $m < n^{2/3-\varepsilon}$ . The other zone negligible.

By substitution and asymptotic expansion near  $n$ , we have

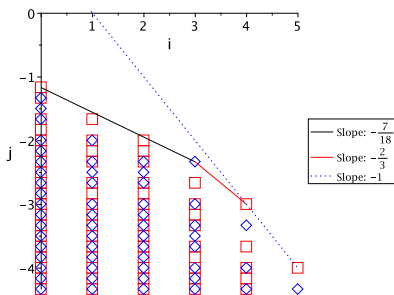
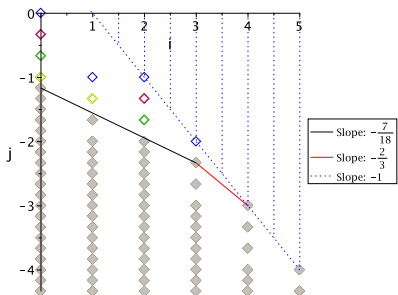
$$P_{n,m} = p_0(n, m)\text{Ai}(\alpha) + p_1(n, m)\text{Ai}'(\alpha), \text{ with } \alpha = a_1 + \frac{2^{1/3}m}{n^{1/3}}.$$

$p_0(n, m), p_1(n, m)$ : series in  $n^{-1/6}$  with polynomial coeffs in  $m$ .

# Lower bound - Newton polygon

$$P_{n,m} = \text{Ai}(\alpha) \left( \frac{1}{n^{7/6}} - \frac{2^{5/3} a_1 m}{3n^{5/3}} - \frac{41m^2}{9n^2} - \frac{2^{8/3} a_1 m^3}{3n^{8/3}} - \frac{34m^4}{9n^3} + \dots \right) +$$

$$\text{Ai}'(\alpha) \left( \frac{2^{1/3}}{n^{3/2}} - \frac{8a_1 m}{9n^2} - 19 \frac{2^{1/3} m^2}{9n^{7/3}} - \frac{2^{13/3} m^3}{9n^{7/3}} + \dots \right).$$



# Lower bound - case analysis

$$P_{n,m} = \text{Ai}(\alpha) \left( \frac{1}{n^{7/6}} - \frac{2^{5/3}a_1m}{3n^{5/3}} - \frac{41m^2}{9n^2} - \frac{2^{8/3}a_1m^3}{3n^{8/3}} - \frac{34m^4}{9n^3} + \dots \right) +$$

$$\text{Ai}'(\alpha) \left( \frac{2^{1/3}}{n^{3/2}} - \frac{8a_1m}{9n^2} - 19\frac{2^{1/3}m^2}{9n^{7/3}} - \frac{2^{13/3}m^3}{9n^{7/3}} + \dots \right).$$

- $m \leq x_0(n/2)^{1/3}$ , where  $\text{Ai}'(a_1 + x)$  changes sign,
- $x_0(n/2)^{1/3} < m \leq n^{7/18}$ ,
- $n^{7/18} < m < n^{2/3-\varepsilon}$ .

All cases are positive using properties of the Airy function.

# Upper bound

It is the same, with a different *ansatz*:

$$\hat{X}_{n,m} = \left( 1 - \frac{2m^2}{3n} + \frac{m}{2n} + \frac{3m^4}{10n^2} \right) \text{Ai} \left( a_1 + \frac{2^{1/3}(m+1)}{n^{1/3}} \right),$$
$$\hat{s}_n = 2 + \frac{2^{2/3}a_1}{n^{2/3}} + \frac{8}{3n} + \frac{1}{n^{7/6}}.$$


Yet another case analysis ...

$$r_n = \Theta \left( n! 4^n e^{3a_1 n^{1/3}} n \right).$$

# Cherry lemma

On **compacted trees**:

## Lemma



*For a relaxed tree  $T$ , if no cherry  reproduces a node that has appeared, then  $T$  is compacted.*

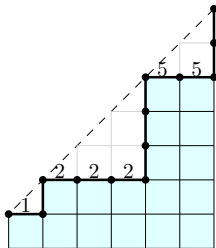
$T$  not compacted  $\Rightarrow$  two nodes with the same decompressed trees


The same holds for their children.


**Descend until reaching a cherry**

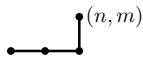
# Encoding by decorated Dyck paths (compacted version)

Cherry lemma   $\Rightarrow$  avoid certain 



weight  $m + 1$    $(n, m)$

weight 1   $(n, m)$



remove  $(m - 1)$  possibilities

## Proposition

Let  $e_{n,m}$  be the number of “strict” decorated paths to  $(n, m)$ . Then

$$e_{n,m} = (m + 1)e_{n-1,m} + e_{n,m-1} - (m - 1)e_{n-2,m-1}, \text{ for } n \geq m \geq 1.$$

The number of **compacted trees** with  $n$  nodes is  $c_n = e_{n,n}$ .

# Compacted trees

Recurrence for compacted trees:

$$e_{n,m} = \frac{n-m+2}{n+m} e_{n-1,m-1} + e_{n-1,m+1} - \frac{2(n-m-2)}{(n+m)(n+m-2)} e_{n-3,m-1}.$$

Negative terms ...

Sandwich it by two positive recurrences.

With two appropriate *Ansätze*, we have

$$c_n = \Theta \left( n! 4^n e^{3a_1 n^{1/3}} n^{3/4} \right).$$



# A change in the polynomial factor

*Ansatz* for lower bound :

$$\hat{X}_{n,m} = \left(1 - \frac{2m^2}{3n} + \frac{m}{4n}\right) \text{Ai} \left(a_1 + \frac{2^{1/3}(m+1)}{n^{1/3}}\right),$$

$$\hat{s}_n = 2 + \frac{2^{2/3}a_1}{n^{2/3}} + \frac{13}{6n} - \frac{1}{n^{7/6}}.$$

The only difference in  $\hat{s}_n \Rightarrow$  change the polynomial factor

# An application on automata

## Theorem (Elvey Price, F., Wallner 2020)

The number  $m_{2,n}$  of minimal automata for finite languages in  $A = \{a, b\}$  with  $n$  states is

$$m_{2,n} = \Theta \left( n! 8^n e^{3a_1 n^{1/3}} n^{7/8} \right).$$

- Similar “compression”: [minimal automata as compressed trie](#)
- Encoding by decorated Dyck paths, similar recurrence
- A “cherry lemma”
- [Exactly the same method](#), can do any fixed alphabet size

# Summing up

## What is good:

- Using only a (quite simple) recurrence;
- Without looking at the generating function;
- Relatively simple, so possible to generalize.
- Sometimes negative terms are not a problem.

## Still need work :

- Which type of recurrence? Which type of diff. eq.?
- We still need to start from some heuristics...
- And we miss the multiplicative constant.

## Already some other applications!

Michael Fuchs, Guan-Ru Yu, Louxin Zhang, *On the Asymptotic Growth of the Number of Tree-Child Networks*, European J. Combin., 2021.

Yu-Sheng Chang, Michael Fuchs, Hexuan Liu, Michael Wallner, Guan-Ru Yu, *Enumerative and Distributional Results for  $d$ -combining Tree-Child Networks*, arXiv:2209.03850, 2022.

# Ongoing work

- With Baptiste Louf, we are trying to apply the method to [maps](#).
- Classification of “[linearly rational up-step](#)” recurrences:
  - degenerated or trivial,
  - stretched exponential  $\rho^{n^{1/3}}$ ,
  - macroscopic limit,
  - ... maybe more?
- General theorem for stretched exponential other than the Airy type
  - [Whittaker type](#):  $\rho^{n^{1/2}}$ ,
  - ... and further types like  $\rho^{n^{\frac{p}{p+2}}}$ .

Any recurrences in two parameters for asymptotics?

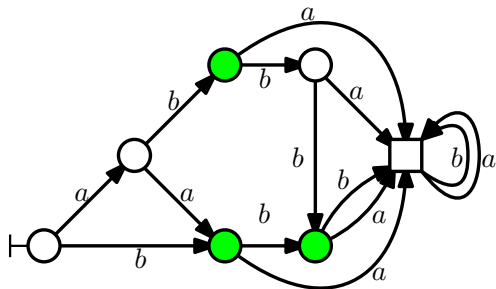
# Ongoing work

- With Baptiste Louf, we are trying to apply the method to [maps](#).
- Classification of “[linearly rational up-step](#)” recurrences:
  - degenerated or trivial,
  - stretched exponential  $\rho^{n^{1/3}}$ ,
  - macroscopic limit,
  - ... maybe more?
- General theorem for stretched exponential other than the Airy type
  - [Whittaker type](#):  $\rho^{n^{1/2}}$ ,
  - ... and further types like  $\rho^{n^{\frac{p}{p+2}}}$ .

Any recurrences in two parameters for asymptotics?

Thank you for your attention!

# Automata



Deterministic automaton  $Q$  on alphabet  $A$ :

- States and transitions,
- Initial state  $q_0$  and some final states,
- Recognizing  $w \Leftrightarrow$  the walk from  $q_0$  reading  $w$  arrives at a final state.

Example:  $aab$  recognized, but  $aaba$  not.

# Minimal automata of a finite language

A language = a set of words  $\Rightarrow$  a **unique** minimal automaton

An automaton is

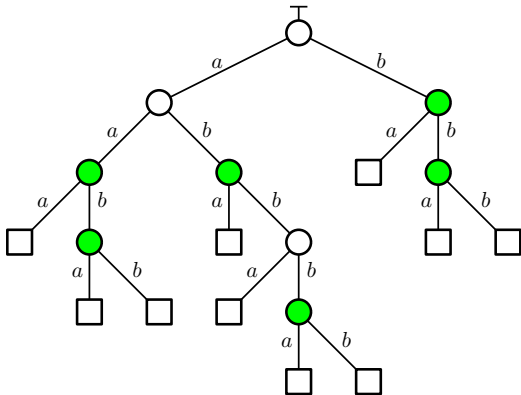
- **accessible**: all states reachable from the initial one,
- **acyclic**: no oriented cycle,
- **reduced**: no redundant state for language recognition.

These three conditions  $\Leftrightarrow$  minimal automaton of some finite language

**Question** : How many such automata with  $n$  states?

Quite “compacted trees”!

# Minimize a trie



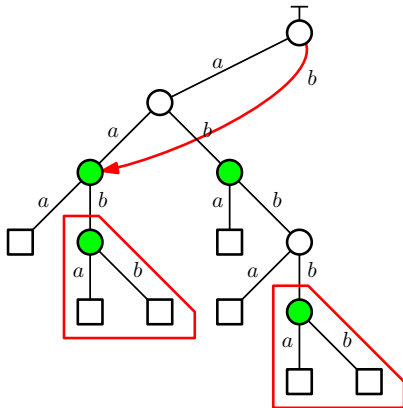
Take a trie and compactify it ...





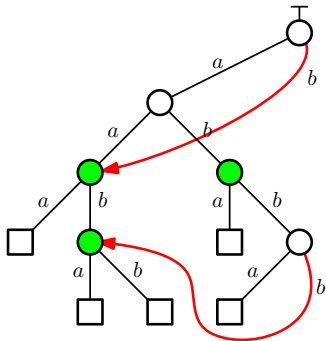


# Minimize a trie



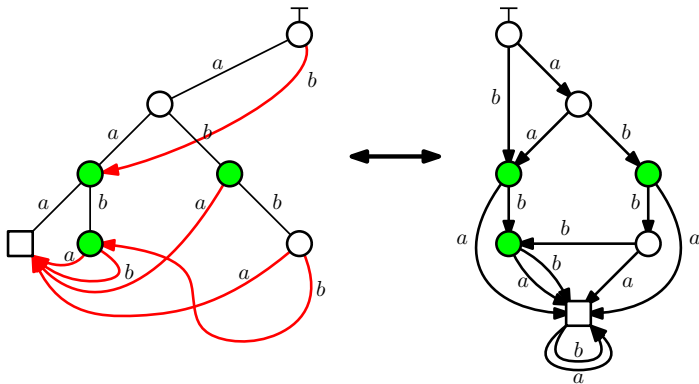
... while exhausting all possibilities ...

# Minimize a trie



... while exhausting all possibilities ...

## Minimize a trie



... and we get a minimal automaton. > Back <