# Short synchronizing words for random automata

Guillaume Chapuy

CNRS – IRIF – Université Paris Cité – ERC CombiTop

based on joint work with

Guillem Perarnau

Universitat Politècnica de Catalunya

$\longrightarrow$ on arxiv last July: arXiv:2207.14108

# Short synchronizing words for random automata

Guillaume Chapuy

CNRS – IRIF – Université Paris Cité – ERC CombiTop

based on joint work with

Guillem Perarnau

Universitat Politècnica de Catalunya

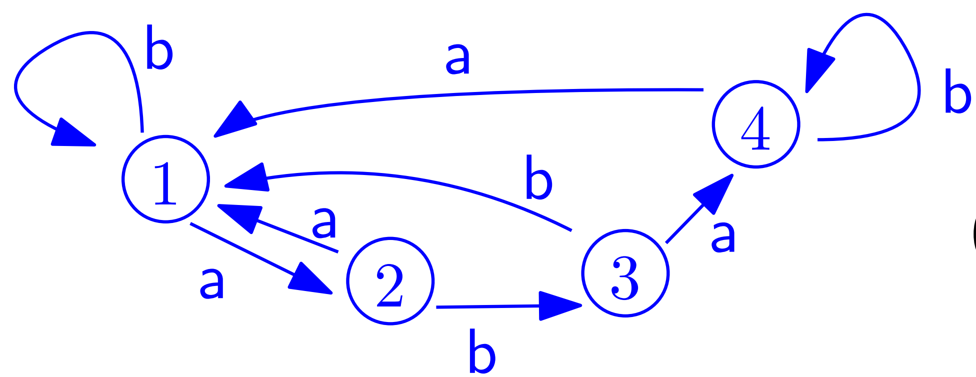
$\longrightarrow$ on arxiv last July: arXiv:2207.14108

# Automata, synchronizing words

# Automata

- An automaton with $n$ states on $\{a, b\}$ is the data of two functions:

$$a : [n] \longrightarrow [n]$$
$$b : [n] \longrightarrow [n]$$

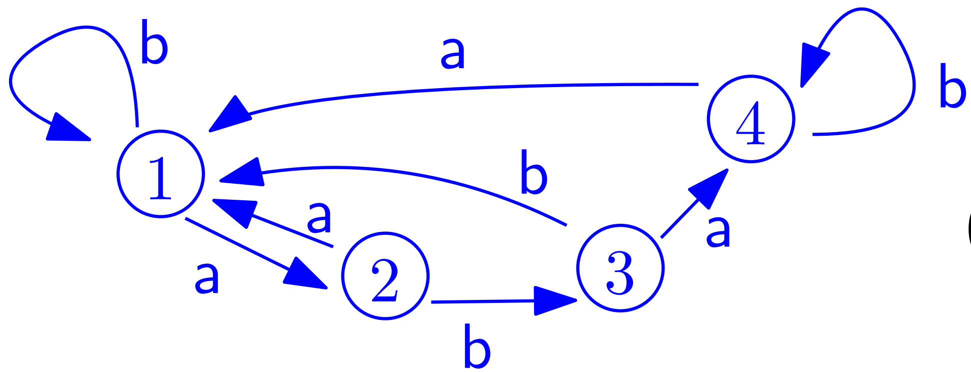



(there are $n^n \times n^n = n^{2n}$ such things)

# Automata

- An automaton with $n$ states on $\{a, b\}$ is the data of two functions:

$$a : [n] \longrightarrow [n]$$
$$b : [n] \longrightarrow [n]$$




(there are $n^n \times n^n = n^{2n}$ such things)

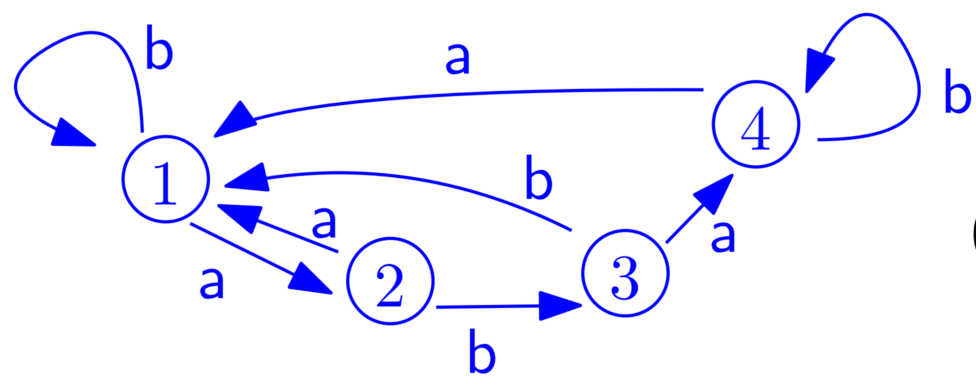
- Notion of $w$-transitions: if $v \in [n]$ and $w \in \{a, b\}^*$, we can read $w$ starting from $v$

for example: $w = ababb$, $1 \xrightarrow{w} 4$

# Automata

- An automaton with $n$ states on $\{a, b\}$ is the data of two functions:

$$a : [n] \longrightarrow [n]$$
$$b : [n] \longrightarrow [n]$$




(there are $n^n \times n^n = n^{2n}$ such things)

- Notion of $w$-transitions: if $v \in [n]$ and $w \in \{a, b\}^*$, we can read $w$ starting from $v$

for example: $w = ababb$, $1 \xrightarrow{w} 4$

- Fix a subset $S \subset [n]$. Language recognized by an automaton (not used in this talk)

$= $ set of all words $w$ s.t. $1 \xrightarrow{w} s$ with $s \in S$

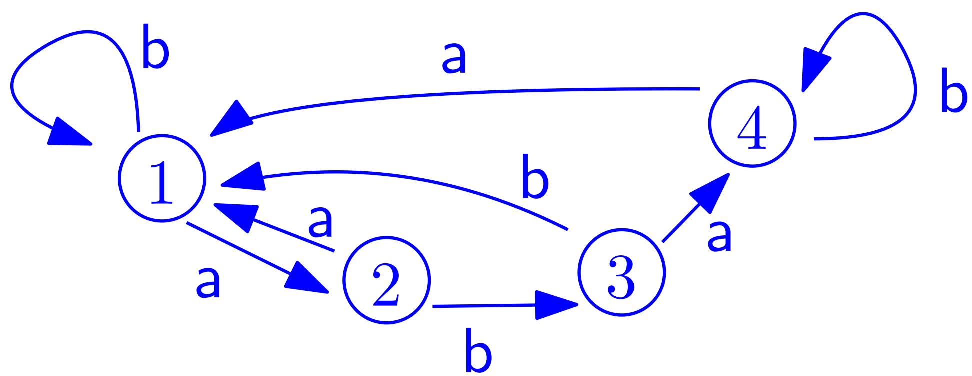
Recognized by automaton iff. recognized by regular expression
All the super nice theory of regular/rational languages (Chomtsky-Schutzenberger)
(still full of incredible open problems!!!)

# Synchronizing words

- A word $w$ is synchronizing if there exists $v_0 \in [n]$ such that

$$v \xrightarrow{w} v_0 \text{ for all } v \in [n]$$

(think of a reset word. Basic motivation: the german-speaking microwave oven at IRIF)




Here $w = b^2ab^2$ works.

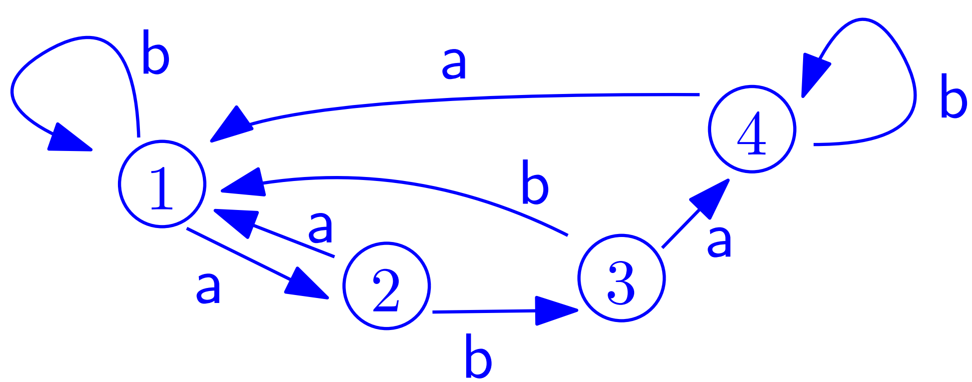
($b^2$ syncs $1, 2, 3 \to 1$ and $4 \to 4$
then $a$ sends $1, 4 \to 1, 2$
so $b^2$ again syncs everyone)

# Synchronizing words

- A word $w$ is synchronizing if there exists $v_0 \in [n]$ such that

$$v \xrightarrow{w} v_0 \text{ for all } v \in [n]$$

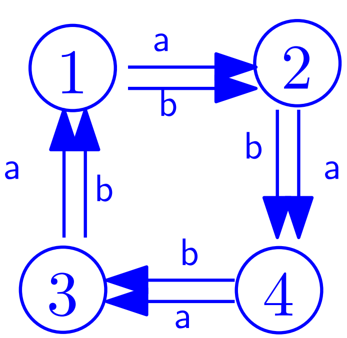
(think of a reset word. Basic motivation: the german-speaking microwave oven at IRIF)




Here $w = b^2ab^2$ works.

($b^2$ syncs $1, 2, 3 \to 1$ and $4 \to 4$
then $a$ sends $1, 4 \to 1, 2$
so $b^2$ again syncs everyone)

- Not all automata are synchonizable !!!




( Note: checking synchronizability = easy; finding shortest word = NP-hard )

# Shortest synchronizing words?

- Remark (Czerny 1960's)

  If $A$ is synchronizable, there is sync word of length $\leq n^3$

  (synchronize $1, 2$ with a word $w$ of length $\leq n^2$ by pigeonhole on pairs of visited vertices
  then repeat $n - 1$ times)

# Shortest synchronizing words?

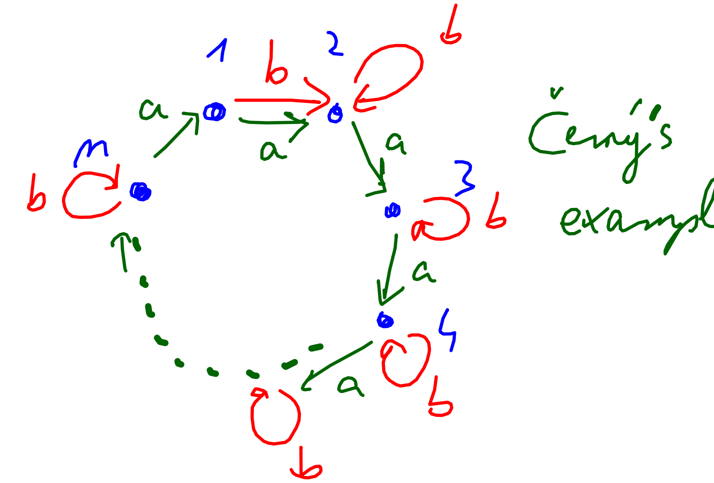
- Remark (Czerny 1960's)

  If $A$ is synchronizable, there is sync word of length $\leq n^3$

  (synchronize $1, 2$ with a word $w$ of length $\leq n^2$ by pigeonhole on pairs of visited vertices then repeat $n-1$ times)

- **Černý's conjecture (1960's)** If $A$ is synchronizable, then there is a sync word of length $\leq (n-1)^2$

  (one of the biggest open problems in automata theory!!!)



# Shortest synchronizing words?

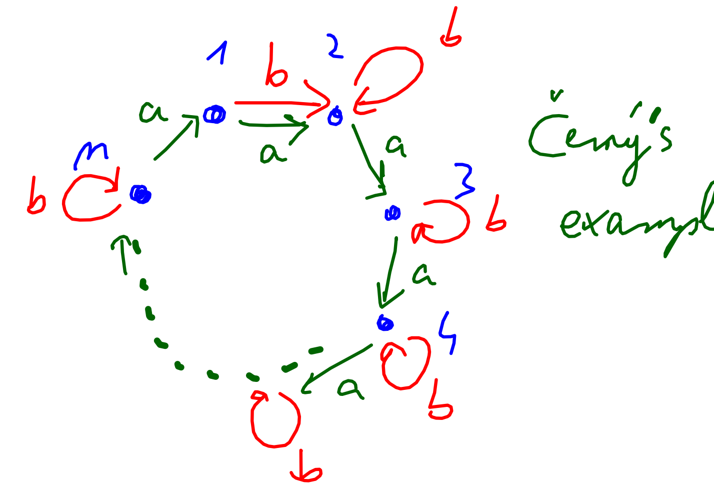
- Remark (Czerny 1960's)

  If $A$ is synchronizable, there is sync word of length $\leq n^3$

  (synchronize $1, 2$ with a word $w$ of length $\leq n^2$ by pigeonhole on pairs of visited vertices then repeat $n-1$ times)

- **Černý's conjecture (1960's)** If $A$ is synchronizable, then there is a sync word of length $\leq (n-1)^2$

  (one of the biggest open problems in automata theory!!!)




Best results are $cn^3$: [Pin-Frankl 1983] $c = \frac{1}{6}$; [Szykuła 2018] $c = 0.1666$ [Shitov 2019] $c = 0.1654$

# Shortest synchronizing words?

- Remark (Czerny 1960's)

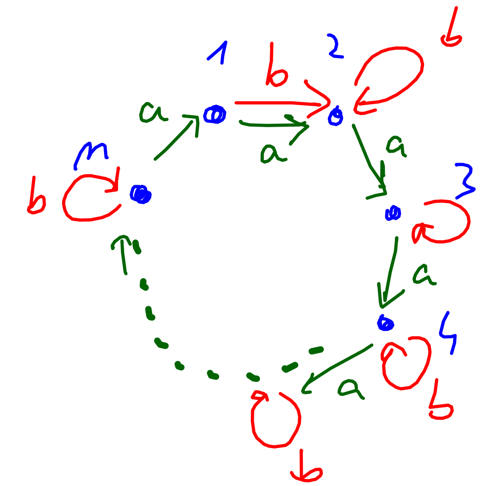
  If $A$ is synchronizable, there is sync word of length $\leq n^3$

  (synchronize $1, 2$ with a word $w$ of length $\leq n^2$ by pigeonhole on pairs of visited vertices then repeat $n-1$ times)

- **Černý's conjecture (1960's)** If $A$ is synchronizable,

  then there is a sync word of length $\leq (n-1)^2$

  (one of the biggest open problems in automata theory!!!)




Černý's example

Best results are $cn^3$: [Pin-Frankl 1983] $c = \frac{1}{6}$; [Szykuła 2018] $c = 0.1666$ [Shitov 2019] $c = 0.1654$

- What about random automata ???

- **Conjecture** [Cameron 2013] A random automaton is synchronizable w.h.p.

  Proved! [Berlinkov 2016] " abstract" proof

  [Nicaud 2016] quantitative bound $O(n \log(n)^3)$ for shortest word!

# Shortest sync words in random automata (main result!)

- Experiments and...

**Conjecture** [Kisielewicz, Kowalski, and Szykuła 2013]

The length of the shortest sync word in a uniform random automaton is $\approx \sqrt{n}$ w.h.p !!!

??!! probabilist's view: we should understand where the $\sqrt{n}$ comes from!!! (and prove it!) )

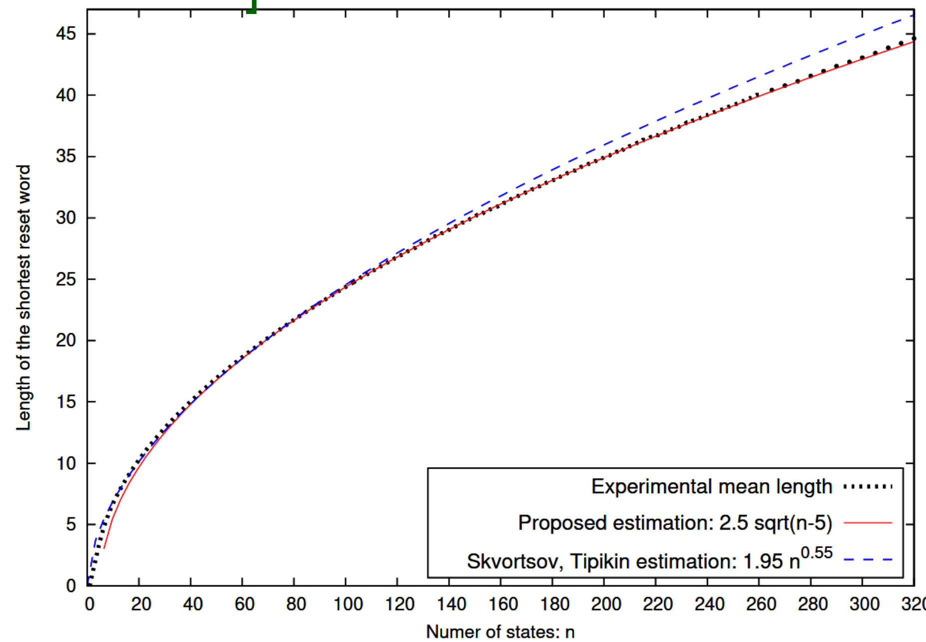



**Fig. 1.** Experimental mean length of the shortest reset words compared with estimations.

# Shortest sync words in random automata (main result!)

- Experiments and...

**Conjecture** [Kisielewicz, Kowalski, and Szykuła 2013]

The length of the shortest sync word in a uniform random automaton is $\approx \sqrt{n}$ w.h.p !!!

??!! probabilist's view: we should understand where the $\sqrt{n}$ comes from!!! (and prove it!) )

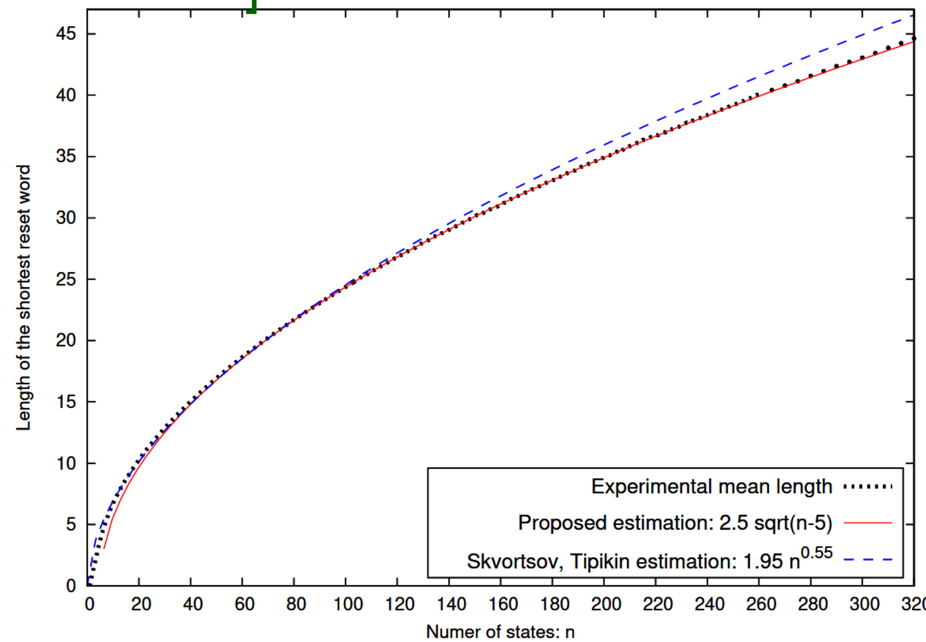



**Fig. 1.** Experimental mean length of the shortest reset words compared with estimations.

**Theorem** [GC+ Guillem Perarnau, July 2022]

The conjecture of Kisielewicz, Kowalski, and Szykuła is true! up to a log factor. With high probability, a uniform random automaton has a synchronizing word of length at most $100\sqrt{n}\log(n)$

Rest of the talk: heuristic of the proof

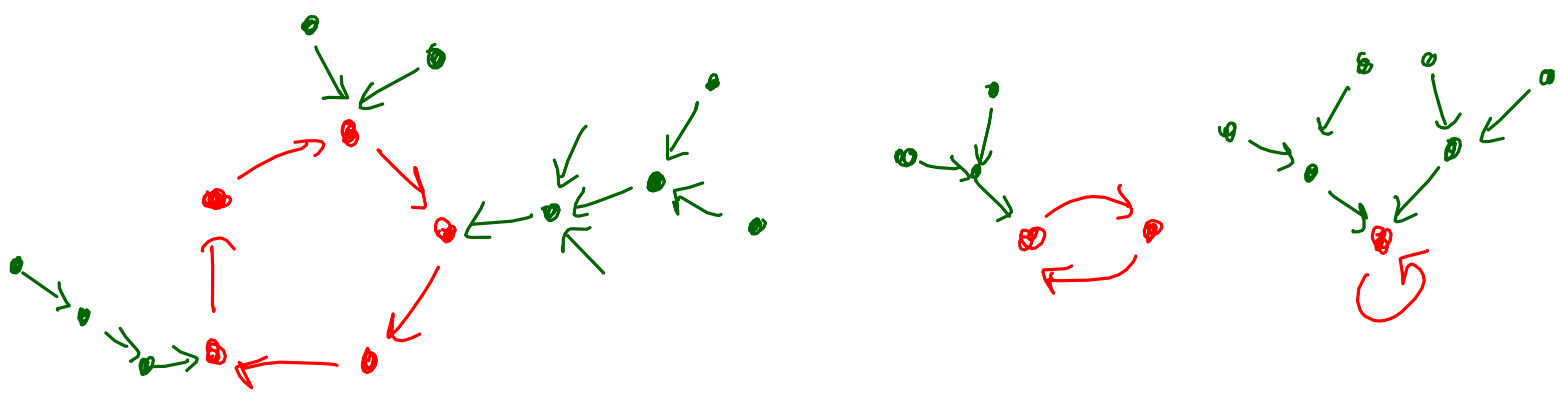
one-letter automata!

# One-letter automata (!)

# One-letter automata!!!

- A one-letter automata is just a function $a : [n] \longrightarrow [n]$
  (i.e. a one-outregular digraph on $[n]$)

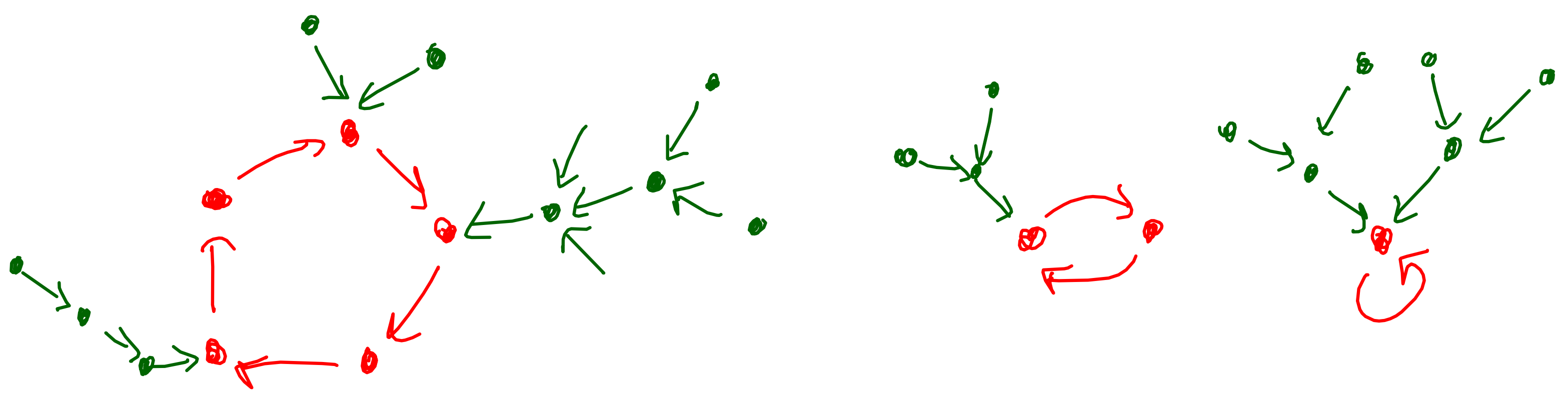
- Such an object is a collection of directed cycles with trees attached to them.



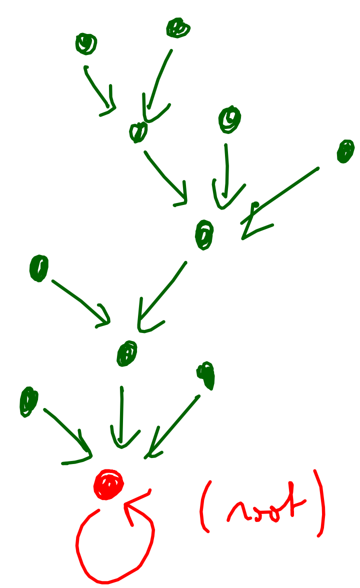
# One-letter automata!!!

- A one-letter automata is just a function $a : [n] \longrightarrow [n]$
  (i.e. a one-outregular digraph on $[n]$)

- Such an object is a collection of directed cycles with trees attached to them.



- It is synchronizable if and only if it is a (cycle-rooted) tree!!!



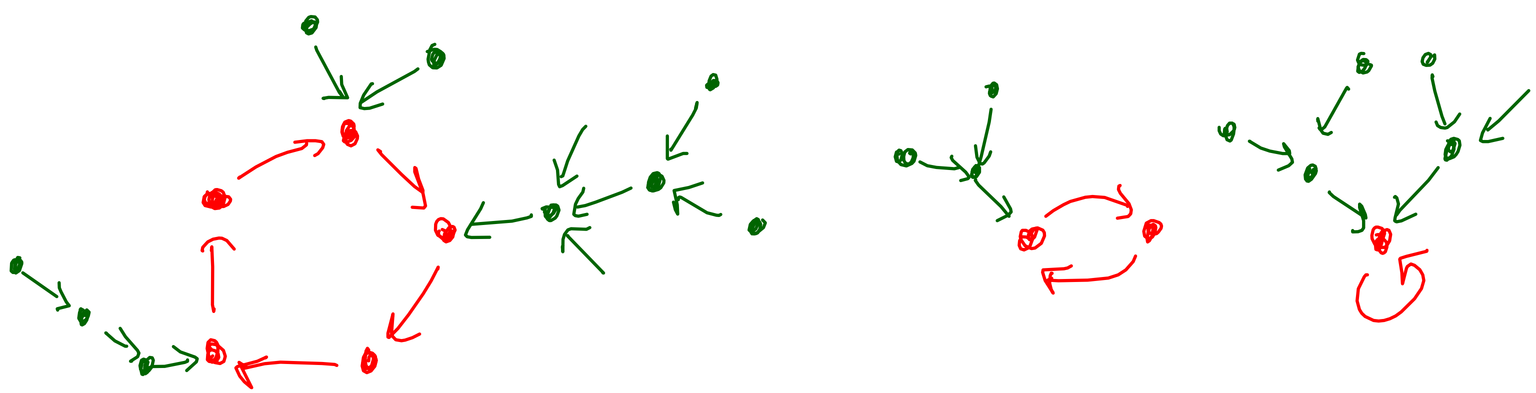

reset word $a^H$

$H =$ height of tree

# One-letter automata!!!

- A one-letter automata is just a function $a : [n] \longrightarrow [n]$
  (i.e. a one-outregular digraph on $[n]$)

- Such an object is a collection of directed cycles with trees attached to them.

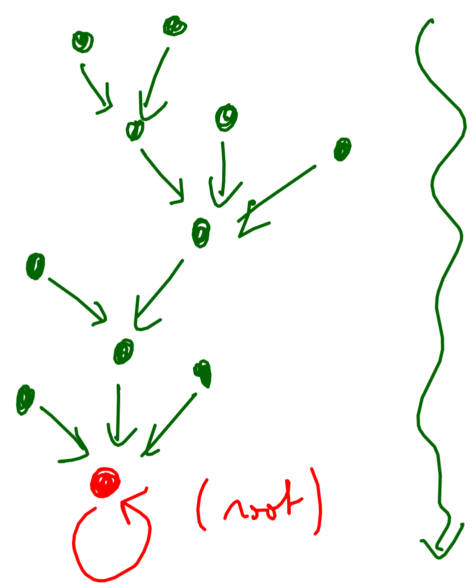


- It is synchronizable if and only if it is a (cycle-rooted) tree!!!

This is happens with probability

$$\frac{\text{nb. of trees}}{\text{nb. of automata}} = \frac{n^{n-1}}{n^n} = \frac{1}{n}$$

(this is Cayley's formula!)



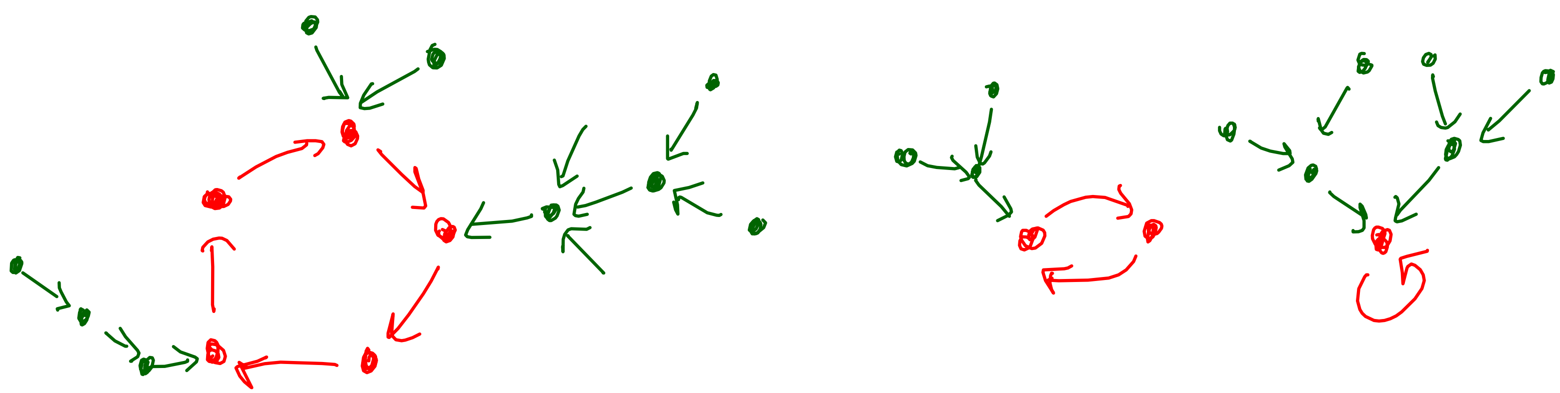

reset word $a^H$

$H$ = height of tree

# One-letter automata!!!

- A one-letter automata is just a function $a : [n] \longrightarrow [n]$
  (i.e. a one-outregular digraph on $[n]$)

- Such an object is a collection of directed cycles with trees attached to them.

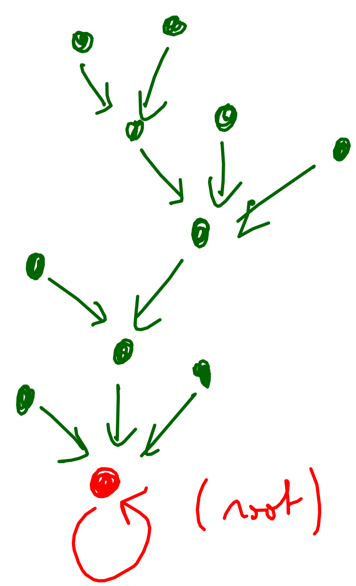


- It is synchronizable if and only if it is a (cycle-rooted) tree!!!

This is happens with probability

$$\frac{\text{nb. of trees}}{\text{nb. of automata}} = \frac{n^{n-1}}{n^n} = \frac{1}{n}$$

(this is Cayley's formula!)




reset word $a^H$

$H =$ height of tree

$H \simeq \sqrt{n}$ w.h.p. (!!?)

# A dream....

- Let $A$ be a random $2$-letter automaton.
  Let $A_w$ be the one-letter automaton induced by $w$-transitions (for some word $w$)

- Maybe....
  $A_w$ somehow behaves as a uniform random one-letter automaton...

# A dream....

- Let $A$ be a random $2$-letter automaton.

  Let $A_w$ be the one-letter automaton induced by $w$-transitions (for some word $w$)

- Maybe....

  $A_w$ somehow behaves as a uniform random one-letter automaton...

- so maybe....

  $A_w$ might be a tree with probability $\frac{1}{n}$

## A dream....

- Let $A$ be a random $2$-letter automaton.
  Let $A_w$ be the one-letter automaton induced by $w$-transitions (for some word $w$)

- Maybe....
  $A_w$ somehow behaves as a uniform random one-letter automaton...

- so maybe....
  $A_w$ might be a tree with probability $\frac{1}{n}$

- and maybe....
  combinatorics is messy enough so the $A_w$ for different $w$ are "somehow independent"
  (hum...)

# A dream....

- Let $A$ be a random $2$-letter automaton.

  Let $A_w$ be the one-letter automaton induced by $w$-transitions (for some word $w$)

- Maybe....

  $A_w$ somehow behaves as a uniform random one-letter automaton...

- so maybe....

  $A_w$ might be a tree with probability $\frac{1}{n}$

- and maybe....

  combinatorics is messy enough so the $A_w$ for different $w$ are "somehow independent"

  (hum...)

- so maybe...

  If I try all the words $w$ of length $(1+\epsilon)\log(n)$ (there are $n^{1+\epsilon} >> n$ of these)

  ... one $w$ will work.

# A dream....

- Let $A$ be a random $2$-letter automaton.

  Let $A_w$ be the one-letter automaton induced by $w$-transitions (for some word $w$)

- Maybe....

  $A_w$ somehow behaves as a uniform random one-letter automaton...

- so maybe....

  $A_w$ might be a tree with probability $\frac{1}{n}$

- and maybe....

  combinatorics is messy enough so the $A_w$ for different $w$ are "somehow independent"

  (hum...)

- so maybe...

  If I try all the words $w$ of length $(1+\epsilon)\log(n)$ (there are $n^{1+\epsilon} >> n$ of these)

  ... one $w$ will work.

- and maybe...

  The automaton $A_w$ is not too far from a uniform tree, its height will be $\approx \sqrt{n}$

  .... so the word $w^H$ of length $\approx \sqrt{n}\log(n)$ will be synchronizing in $A$ !!!

# This works!

- Say that the 2-letter automaton $A$ is a $w$-tree if (the 1-letter aut.) $A_w$ is a tree
- Let $N_k(A)$ the number of $w$ of length $k$ such that $A$ is a $w$-tree$^*$.

# This works!

- Say that the 2-letter automaton $A$ is a $w$-tree if (the 1-letter aut.) $A_w$ is a tree
- Let $N_k(A)$ the number of $w$ of length $k$ such that $A$ is a $w$-tree*.

**Theorem** [GC+ Guillem Perarnau 2022]

For a random 2-letter automaton $A$ on $n$.

$$\mathbf{P}\Big(N_k(A) > 0\Big) \longrightarrow \begin{cases} 0 & , \quad k \leq (1-\epsilon)\log n \\ 1 & , \quad k \geq (1+\epsilon)\log n \end{cases}$$

# This works!

- Say that the 2-letter automaton $A$ is a $w$-tree if (the 1-letter aut.) $A_w$ is a tree
- Let $N_k(A)$ the number of $w$ of length $k$ such that $A$ is a $w$-tree*.

**Theorem** [GC+ Guillem Perarnau 2022]
For a random 2-letter automaton $A$ on $n$.

$$\mathbf{P}\Big(N_k(A) > 0\Big) \longrightarrow \begin{cases} 0 & , \; k \leq (1-\epsilon)\log n \\ 1 & , \; k \geq (1+\epsilon)\log n \end{cases}$$

so whp there exists $w$ of length $(1+\epsilon)\log(n)$ such that $A$ is a $w$-tree.

In fact we have $\mathbf{E}N_k(A) \sim \frac{n^{1+\epsilon}}{n} = n^{\epsilon}$ and second moment concentration (this is how the pf works)

# This works!

- Say that the 2-letter automaton $A$ is a $w$-tree if (the 1-letter aut.) $A_w$ is a tree
- Let $N_k(A)$ the number of $w$ of length $k$ such that $A$ is a $w$-tree*.

**Theorem** [GC+ Guillem Perarnau 2022]
For a random 2-letter automaton $A$ on $n$.

$$\mathbf{P}\Big(N_k(A) > 0\Big) \longrightarrow \begin{cases} 0 & , \ k \leq (1-\epsilon)\log n \\ 1 & , \ k \geq (1+\epsilon)\log n \end{cases}$$

so whp there exists $w$ of length $(1+\epsilon)\log(n)$ such that $A$ is a $w$-tree.

In fact we have $\mathbf{E}N_k(A) \sim \frac{n^{1+\epsilon}}{n} = n^{\epsilon}$ and second moment concentration (this is how the pf works)

- It is easy to see that any branch $v \longrightarrow^*$ in $A_w$ has length $\leq 100\sqrt{n}$ with probability at least $1 - o(n^{-3})$ so we can take union bound on all $w$ and on all $v$ to deduce that the height of $A_w$ is smaller than $100\sqrt{n}$.

- we get a synchronizing word $w^H$ of length $H \cdot |w| = 100(1+\epsilon)\log(n)\sqrt{n}$.

# Two proofs from the book of Cayley's formula

# New (?) proof of $n^{n-1}$ by exploration – telescopic argument

(related to [Foata-Fuchs 1970])

- Let $a : [n] \longrightarrow [n]$ be a uniform random function.

  We reveal $a$ iteratively:

  - pick vertex $1$ and reveal its future until a cycle is made (at some random time $T_1$)

  - pick smallest unexplored and reveal its future until it merges with the previous graph or a cycle is made (at some random time $T_2$)

  ...repeat

  - until last vertex future is revealed (at some time $T_k = n$)

# New (?) proof of $n^{n-1}$ by exploration – telescopic argument

(related to [Foata-Fuchs 1970])

- Let $a : [n] \longrightarrow [n]$ be a uniform random function.

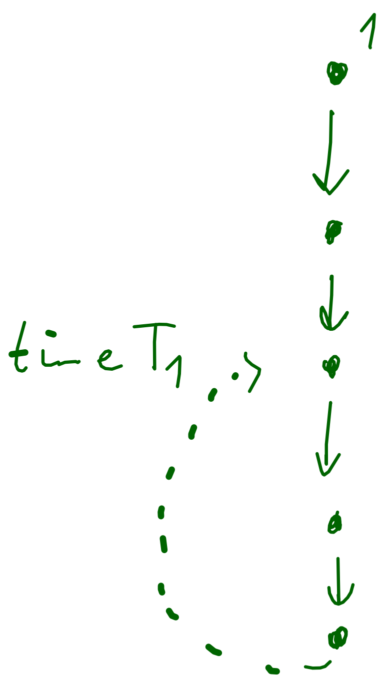
  We reveal $a$ iteratively:

  - pick vertex $1$ and reveal its future until a cycle is made (at some random time $T_1$)

  - pick smallest unexplored and reveal its future until it merges with the previous graph or a cycle is made (at some random time $T_2$)

  ...repeat

  - until last vertex future is revealed (at some time $T_k = n$)



# New (?) proof of $n^{n-1}$ by exploration – telescopic argument

(related to [Foata-Fuchs 1970])

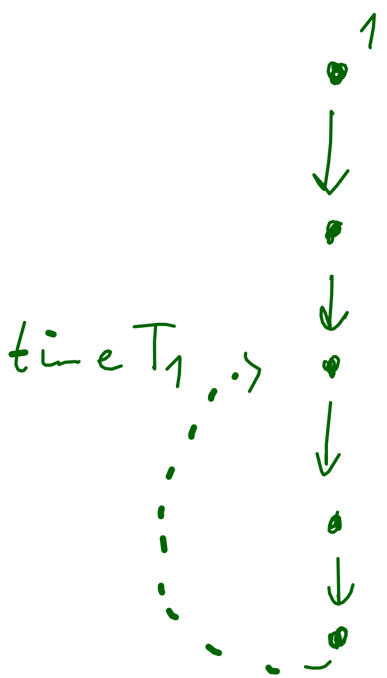
- Let $a : [n] \longrightarrow [n]$ be a uniform random function.

  We reveal $a$ iteratively:

  - pick vertex $1$ and reveal its future until a cycle is made (at some random time $T_1$)
  - pick smallest unexplored and reveal its future until it merges with the previous graph or a cycle is made (at some random time $T_2$)

  ...repeat

  - until last vertex future is revealed (at some time $T_k = n$)



# New (?) proof of $n^{n-1}$ by exploration – telescopic argument

(related to [Foata-Fuchs 1970])

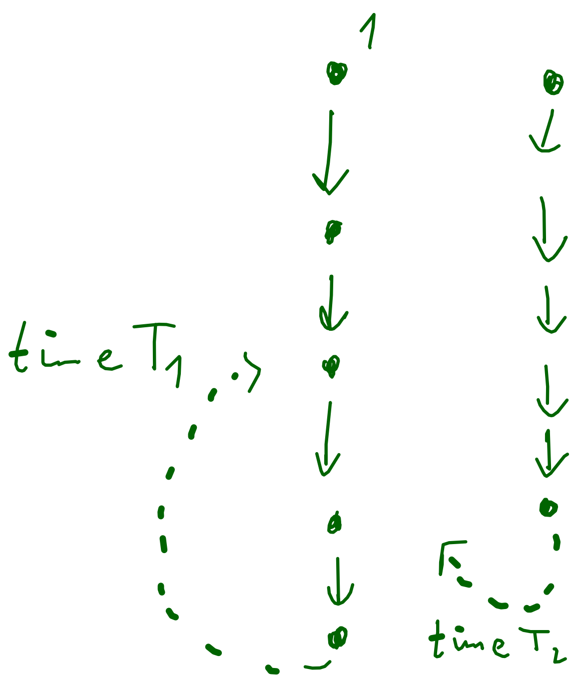
- Let $a : [n] \longrightarrow [n]$ be a uniform random function.

  We reveal $a$ iteratively:

  - pick vertex $1$ and reveal its future until a cycle is made (at some random time $T_1$)
  - pick smallest unexplored and reveal its future until it merges with the previous graph or a cycle is made (at some random time $T_2$)

  ...repeat

  - until last vertex future is revealed (at some time $T_k = n$)



# New (?) proof of $n^{n-1}$ by exploration – telescopic argument

(related to [Foata-Fuchs 1970])

- Let $a : [n] \longrightarrow [n]$ be a uniform random function.

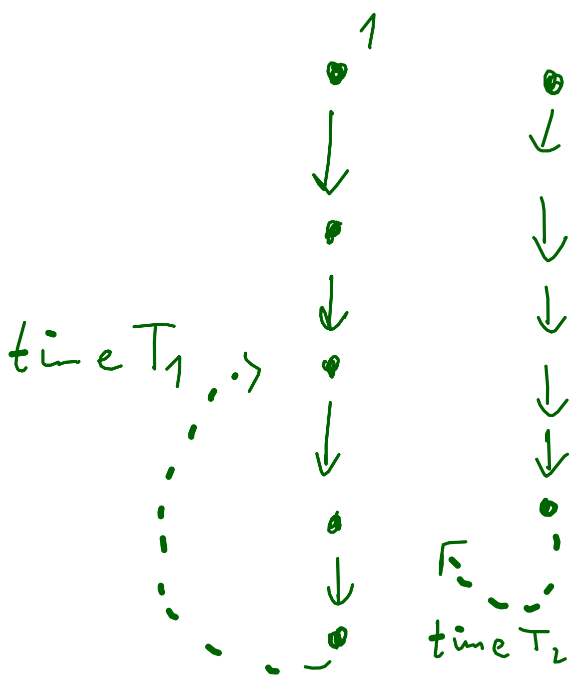
  We reveal $a$ iteratively:

  - pick vertex $1$ and reveal its future until a cycle is made (at some random time $T_1$)

  - pick smallest unexplored and reveal its future until it merges with the previous graph or a cycle is made (at some random time $T_2$)

  ...repeat

  - until last vertex future is revealed (at some time $T_k = n$)



# New (?) proof of $n^{n-1}$ by exploration – telescopic argument

(related to [Foata-Fuchs 1970])

- Let $a : [n] \longrightarrow [n]$ be a uniform random function.

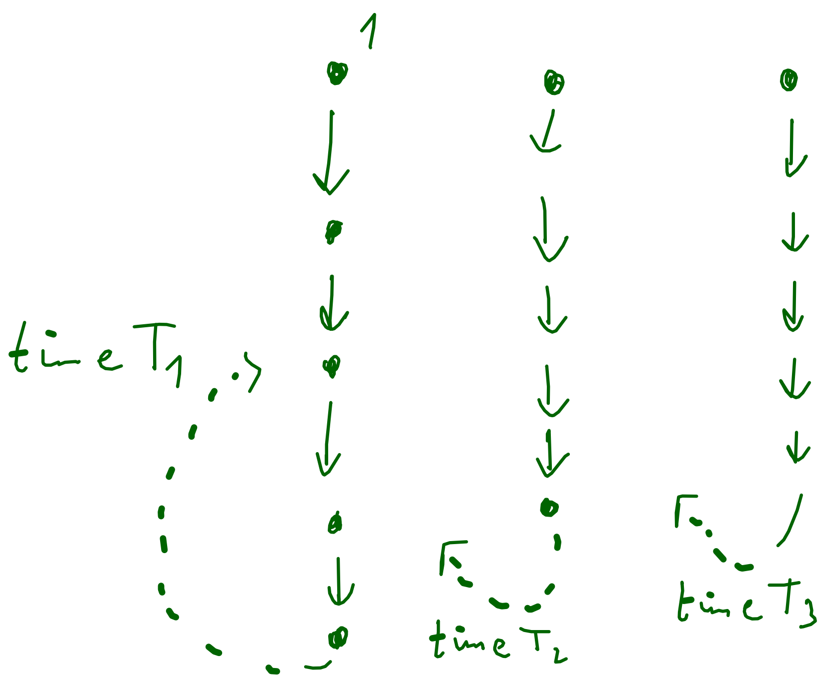
  We reveal $a$ iteratively:

  - pick vertex $1$ and reveal its future until a cycle is made (at some random time $T_1$)

  - pick smallest unexplored and reveal its future until it merges with the previous graph or a cycle is made (at some random time $T_2$)

  ...repeat

  - until last vertex future is revealed (at some time $T_k = n$)



# New (?) proof of $n^{n-1}$ by exploration – telescopic argument

(related to [Foata-Fuchs 1970])

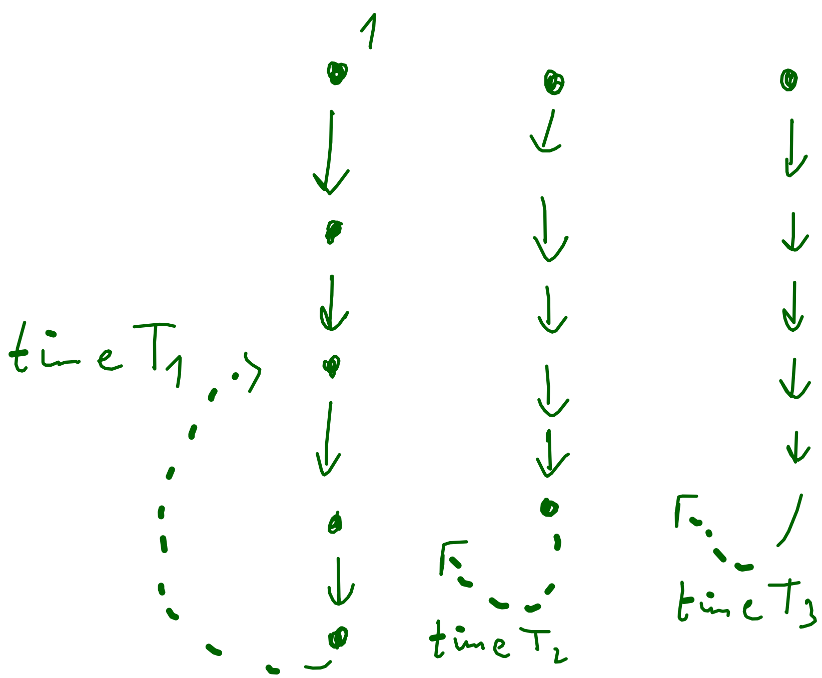
- Let $a : [n] \longrightarrow [n]$ be a uniform random function.

  We reveal $a$ iteratively:

  - pick vertex $1$ and reveal its future until a cycle is made (at some random time $T_1$)
  - pick smallest unexplored and reveal its future until it merges with the previous graph or a cycle is made (at some random time $T_2$)

  ...repeat

  - until last vertex future is revealed (at some time $T_k = n$)




$$\mathbb{P}(\text{get a tree} \mid T_1, \dots, T_K, K) = \frac{1}{T_1} \times \frac{T_1}{T_2} \times \frac{T_2}{T_3} \cdots \times \frac{T_{K-1}}{T_K} = \frac{1}{n} \qquad \text{qed (!)}$$

# New (?) proof of $n^{n-1}$ by exploration – telescopic argument

(related to [Foata-Fuchs 1970])

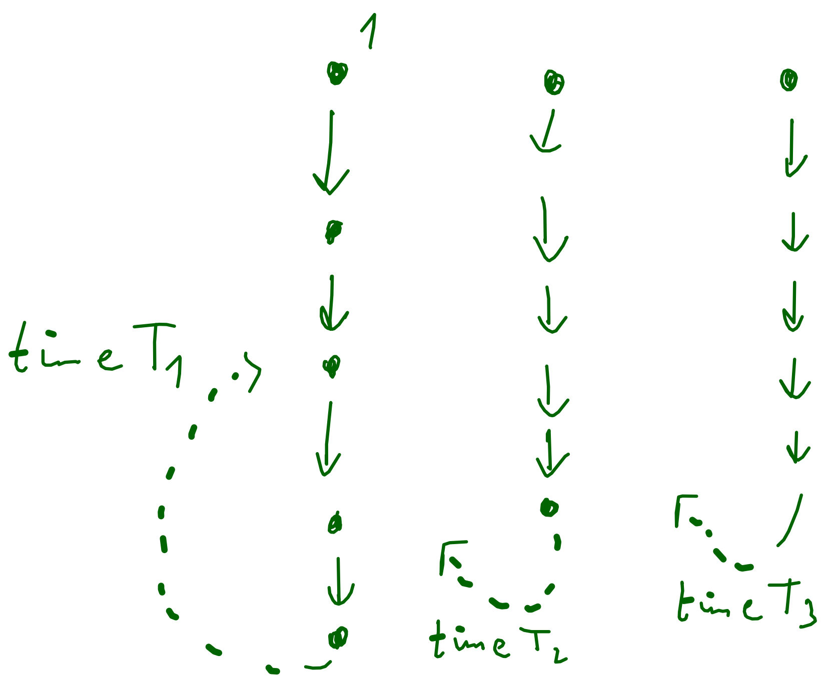
- Let $a : [n] \longrightarrow [n]$ be a uniform random function.

  We reveal $a$ iteratively:

  - pick vertex $1$ and reveal its future until a cycle is made (at some random time $T_1$)
  - pick smallest unexplored and reveal its future until it merges with the previous graph or a cycle is made (at some random time $T_2$)

  ...repeat

  - until last vertex future is revealed (at some time $T_k = n$)




The proof also shows that the height of a random vertex in a random tree is the time of first collision in birthday paradox problem!
(exact equality, in law)

$$P(height = h) = \left(1 - \frac{1}{n}\right)\left(1 - \frac{2}{n}\right)\left(1 - \frac{h-1}{n}\right)\frac{h}{n} \approx \frac{h}{n}e^{-\frac{h^2}{2n}}$$

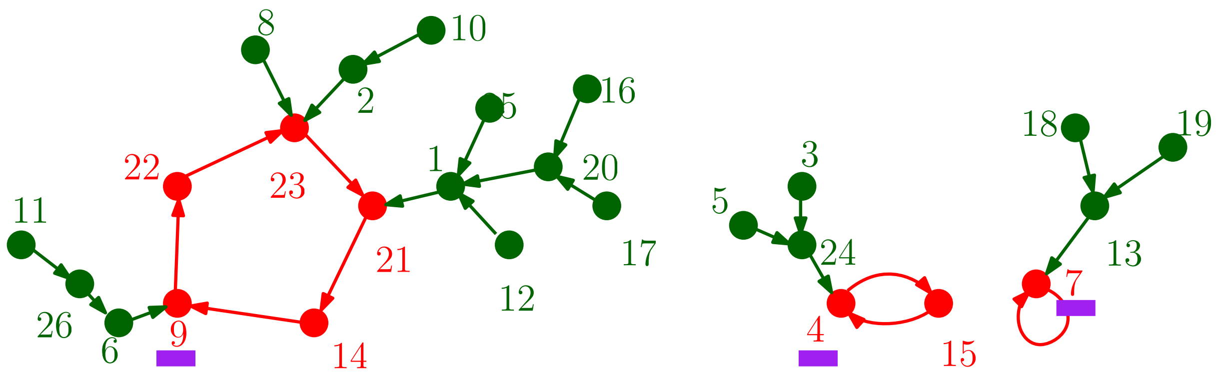
Rayleigh law in scale $\sqrt{n}$ and deviations estimates are trivial.

$$\mathbb{P}(\text{get a tree} \mid T_1, \dots, T_K, K) = \frac{1}{T_1} \times \frac{T_1}{T_2} \times \frac{T_2}{T_3} \cdots \times \frac{T_{K-1}}{T_K} = \frac{1}{n}$$

qed (!)

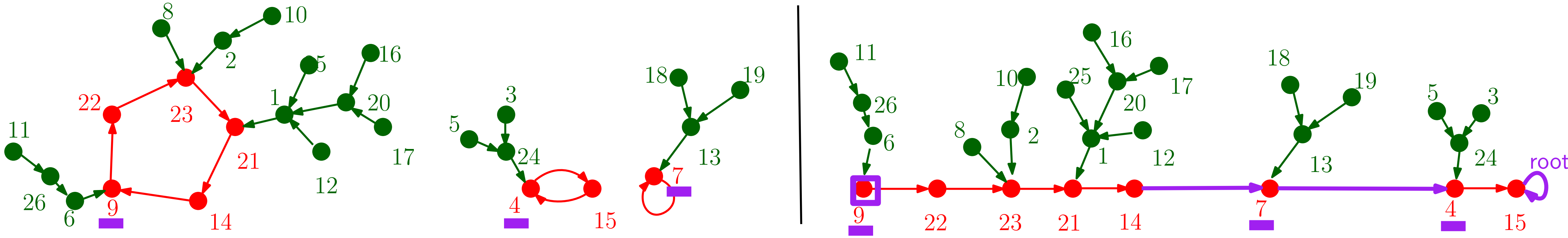
# Joyal's bijection

- Let $a : [n] \longrightarrow [n]$ be a function.
  Remove the edge after the minimum in each cycle
  and concatenate by decreasing minima.



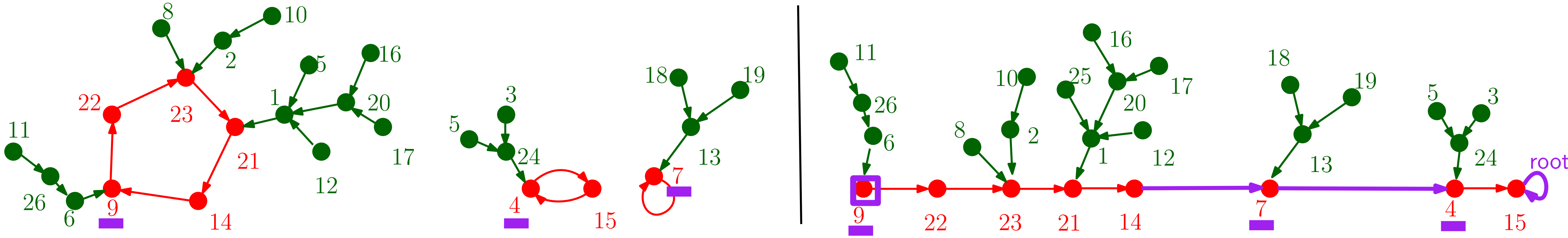
# Joyal's bijection

- Let $a : [n] \longrightarrow [n]$ be a function.
  Remove the edge after the minimum in each cycle
  and concatenate by decreasing minima.



# Joyal's bijection

- Let $a : [n] \longrightarrow [n]$ be a function.
  Remove the edge after the minimum in each cycle
  and concatenate by decreasing minima.

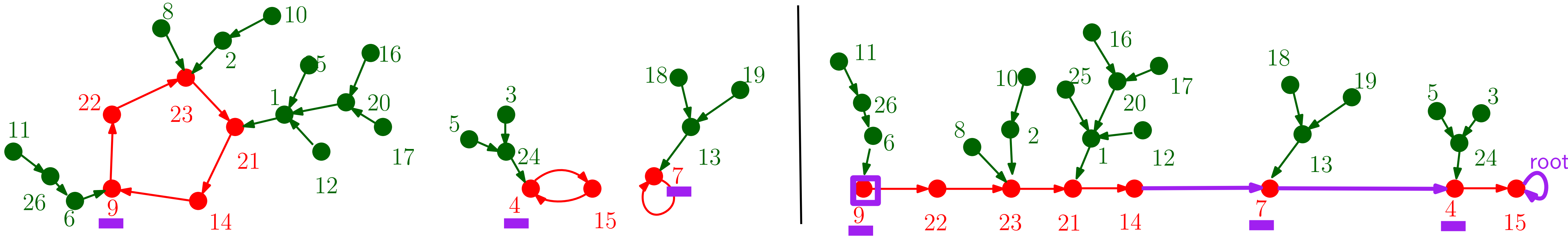



One obtains a doubly marked tree (rewired edges = lower records on the branch)

so $n \times$ rooted trees $= n^n$

# Joyal's bijection

- Let $a : [n] \longrightarrow [n]$ be a function.
  Remove the edge after the minimum in each cycle
  and concatenate by decreasing minima.



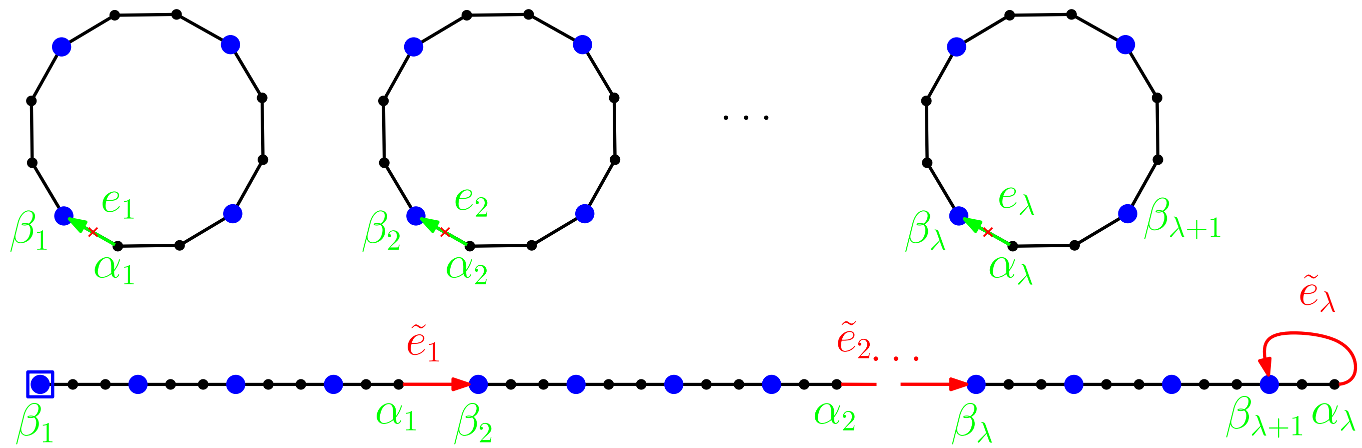

One obtains a doubly marked tree (rewired edges = lower records on the branch)

so $n \times$ rooted trees $= n^n$

- This is super powerful: a random tree and a random function
differ only on $O(\log(n))$ edges!

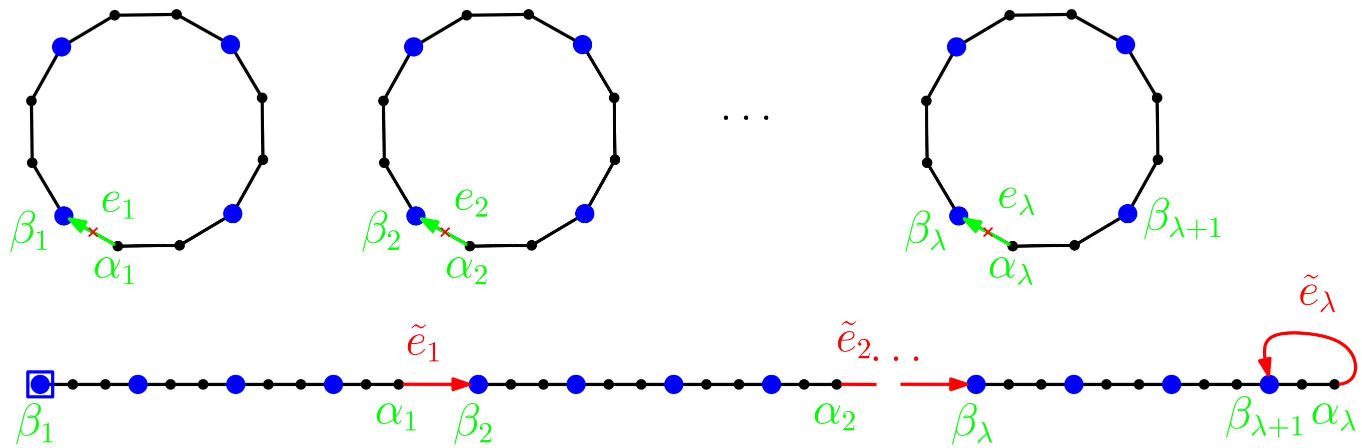
# Our proof

- First moment = count $w$-trees. Apply $w$-variant of Joyal bijection.



# Our proof

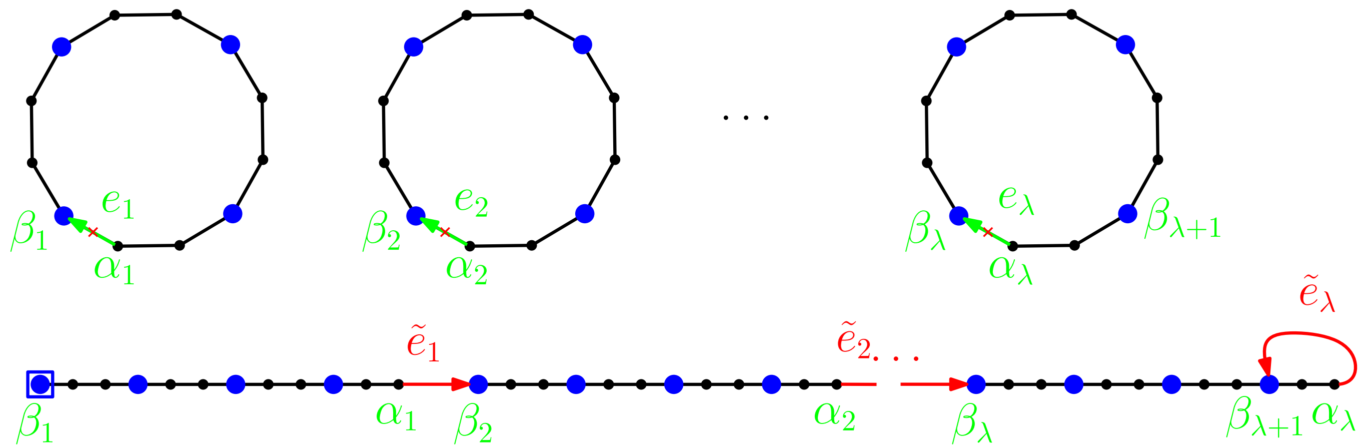
- First moment = count $w$-trees. Apply $w$-variant of Joyal bijection.




- PROBLEM: The $w$-version of the Joyal bijection is only approximate
  - rewiring one edge in fact rewires many edges!!!!
  - could create new cycles by accident!
  - no independence!

# Our proof

- First moment $=$ count $w$-trees. Apply $w$-variant of Joyal bijection.

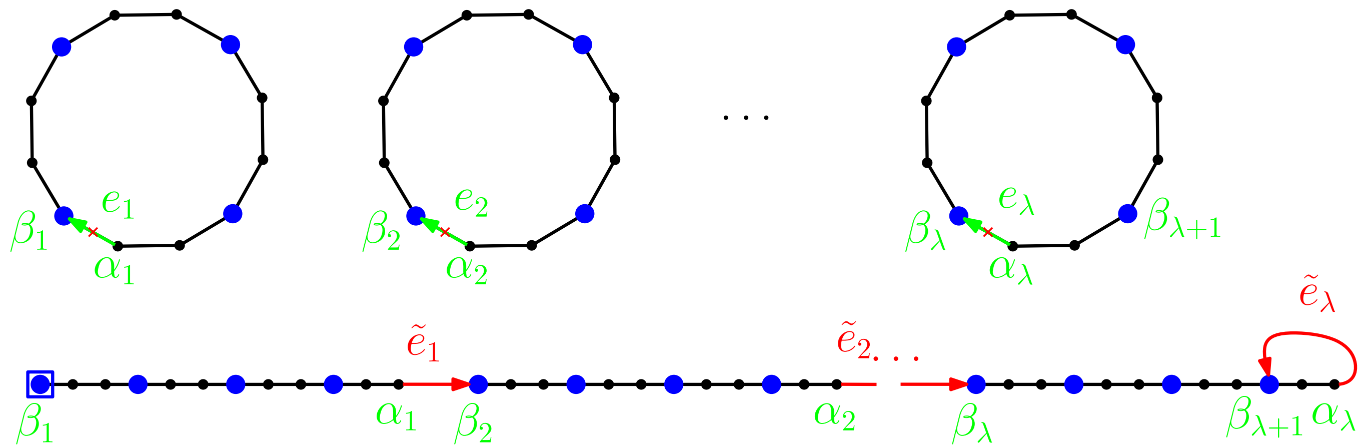



- PROBLEM: The $w$-version of the Joyal bijection is only approximate
  - rewiring one edge in fact rewires many edges!!!!
  - could create new cycles by accident!
  - no independence!

- Second moment: count things which are both $w_1$ and $w_2$ trees.
  Apply $w$-variant of Joyal bijection twice in a row!!!

# Our proof

- First moment $=$ count $w$-trees. Apply $w$-variant of Joyal bijection.




- PROBLEM: The $w$-version of the Joyal bijection is only approximate
  - rewiring one edge in fact rewires many edges!!!!
  - could create new cycles by accident!
  - no independence!

- Second moment: count things which are both $w_1$ and $w_2$ trees.
  Apply $w$-variant of Joyal bijection twice in a row!!!

- SOLUTION:
  We need to control certain bad events under which the bijection fails.
  Example: a $w_1$-lower record contains a $w_2$-lower record in its future
  Final proof is suprisingly messy (with many case disjunctions)

  using the $w$-variant of the exploration process.

# Open problems

- Exact counting of $w$-trees? (start e.g. with $w = aab$)

- Do random $w$-trees converge to the CRT ?

- Problem: improve bounds on the height of a random $w$-tree and (hopefully) improve our result to something like $\sqrt{n}\sqrt{\log n} \times O_P(1)$.

- Statistics question: I give you a sample of $A_w$, can you tell me $w$ ?

  (e.g. discriminate $aa$ from $ab$)

# Open problems

- Exact counting of $w$-trees? (start e.g. with $w = aab$)

- Do random $w$-trees converge to the CRT ?

- Problem: improve bounds on the height of a random $w$-tree and (hopefully) improve our result to something like $\sqrt{n}\sqrt{\log n} \times O_P(1)$.

- Statistics question: I give you a sample of $A_w$, can you tell me $w$ ?
(e.g. discriminate $aa$ from $ab$)

- Fun fact: we prove the conjecture of Kisielewicz, Kowalski, and Szykuła (2013) about the $n^{0.5}$ exponent. But almost the same day we put our paper on arxiv, Szykuła and Zyzik put a paper going much further in the simulations and saying that the estimate is probably wrong, suggesting $n^{0.55}$ instead...

# Open problems

- Exact counting of $w$-trees? (start e.g. with $w = aab$)

- Do random $w$-trees converge to the CRT ?

- Problem: improve bounds on the height of a random $w$-tree and (hopefully) improve our result to something like $\sqrt{n}\sqrt{\log n} \times O_P(1)$.

- Statistics question: I give you a sample of $A_w$, can you tell me $w$ ?
  (e.g. discriminate $aa$ from $ab$)

- Fun fact: we prove the conjecture of Kisielewicz, Kowalski, and Szykuła (2013) about the $n^{0.5}$ exponent. But almost the same day we put our paper on arxiv, Szykuła and Zyzik put a paper going much further in the simulations and saying that the estimate is probably wrong, suggesting $n^{0.55}$ instead...

Merci!

## Open problems

- Exact counting of $w$-trees? (start e.g. with $w = aab$)

- Do random $w$-trees converge to the CRT ?

- Problem: improve bounds on the height of a random $w$-tree and (hopefully) improve our result to something like $\sqrt{n}\sqrt{\log n} \times O_P(1)$.

- Statistics question: I give you a sample of $A_w$, can you tell me $w$ ?
(e.g. discriminate $aa$ from $ab$)

Merci!