Compacted binary trees admit stretched exponentials

CLA 2020 – Online

Andrew Elvey Price, Wenjie Fang, and Michael Wallner

Institute of Discrete Mathematics and Geometry, TU Wien, Austria

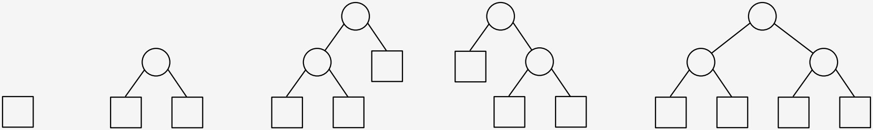
October, 2020

Based on the papers:
Compacted binary trees admit a stretched exponential,
JCTA, Vol. 177(105306), Jan. 2021; ArXiv:1908.11181

Asymptotics of minimal deterministic finite automata recognizing a finite binary language,
AofA 2020.

What is a compacted binary tree?

Let's start simple: binary trees



- *Internal node*: Node of out-degree 2 (circle)
- *Leave*: Node of out-degree 0 (square)
- *Root*: Distinguished node (top node)
- *Left-Right Order* of children

A recursive construction

- A binary tree is either a leaf,
- or it consists of a root and a left and right binary tree.

Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.

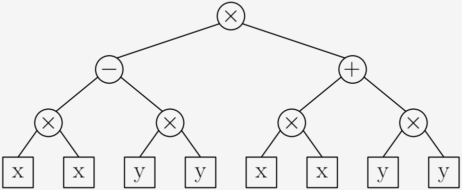
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



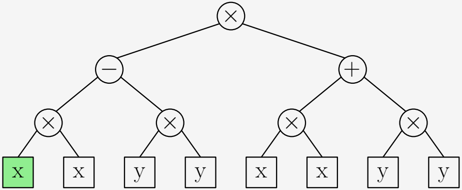
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



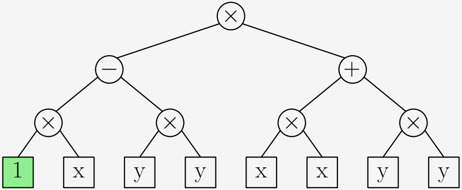
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0))$

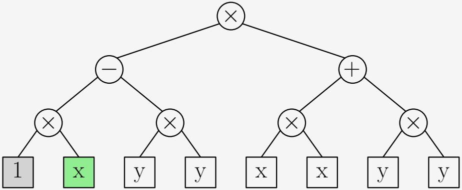
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0))$

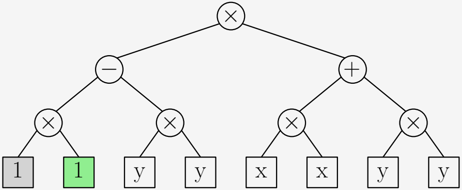
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0))$

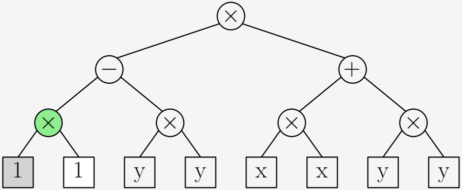
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0))$

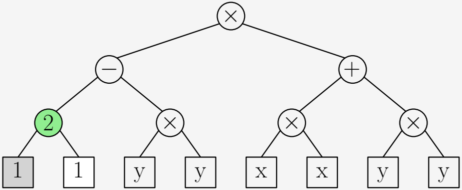
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0)), \quad (2, (\times, 1, 1))$

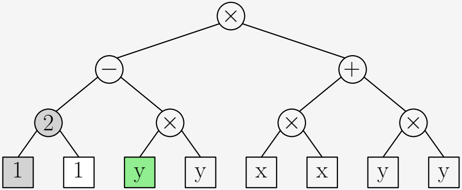
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0)),\quad (2, (\times, 1, 1))$

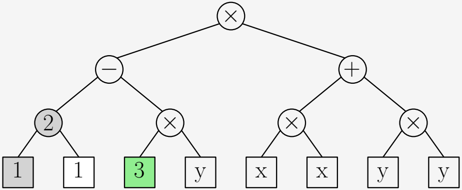
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0))$, $(2, (\times, 1, 1))$, $(3, (y, 0, 0))$

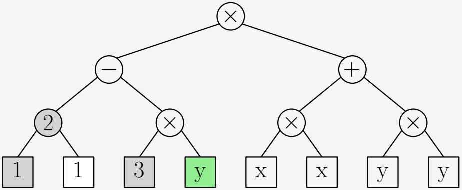
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0)), \quad (2, (\times, 1, 1)), \quad (3, (y, 0, 0))$

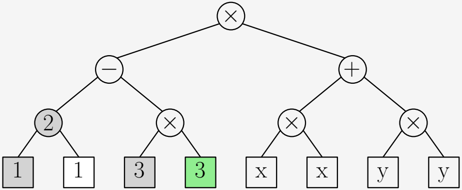
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0))$, $(2, (\times, 1, 1))$, $(3, (y, 0, 0))$

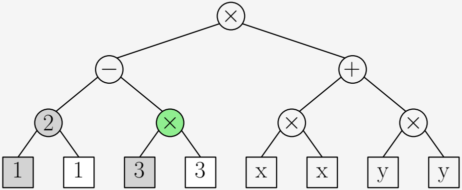
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0)), \quad (2, (\times, 1, 1)), \quad (3, (y, 0, 0))$

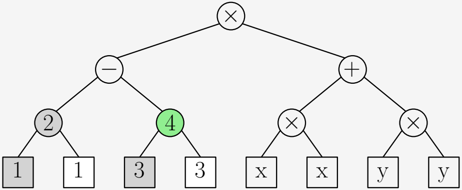
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0)), \quad (2, (\times, 1, 1)), \quad (3, (y, 0, 0)), \quad (4, (\times, 3, 3))$

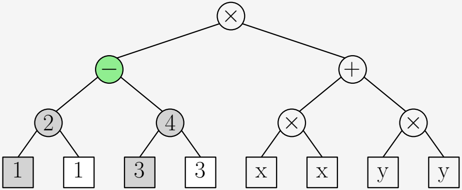
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0))$, $(2, (\times, 1, 1))$, $(3, (y, 0, 0))$, $(4, (\times, 3, 3))$

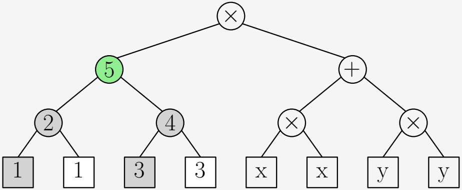
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1,(x,0,0)),\quad (2,(\times,1,1)),\quad (3,(y,0,0)),\quad (4,(\times,3,3)),\quad (5,(-,2,4))$

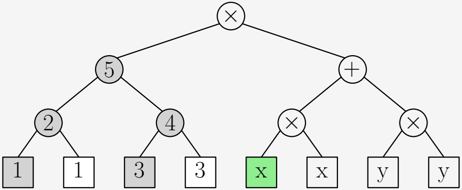
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1,(x,0,0)), \quad (2,(\times,1,1)), \quad (3,(y,0,0)), \quad (4,(\times,3,3)), \quad (5,(-,2,4))$

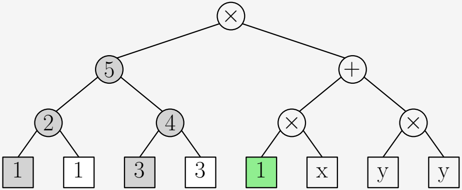
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0))$, $(2, (\times, 1, 1))$, $(3, (y, 0, 0))$, $(4, (\times, 3, 3))$, $(5, (-, 2, 4))$

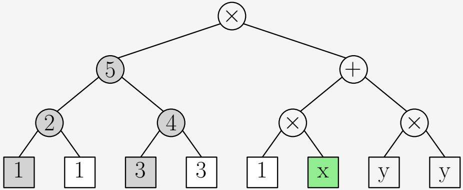
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0))$, $(2, (\times, 1, 1))$, $(3, (y, 0, 0))$, $(4, (\times, 3, 3))$, $(5, (-, 2, 4))$

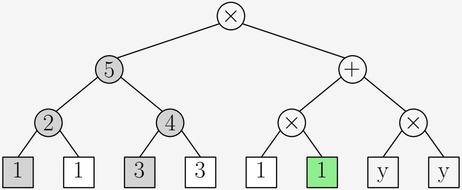
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0)), \quad (2, (\times, 1, 1)), \quad (3, (y, 0, 0)), \quad (4, (\times, 3, 3)), \quad (5, (-, 2, 4))$

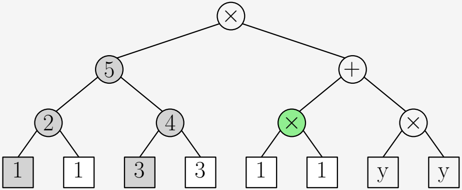
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0))$, $(2, (\times, 1, 1))$, $(3, (y, 0, 0))$, $(4, (\times, 3, 3))$, $(5, (-, 2, 4))$

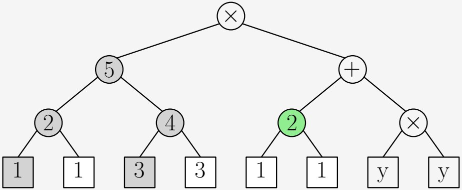
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0))$, $(2, (\times, 1, 1))$, $(3, (y, 0, 0))$, $(4, (\times, 3, 3))$, $(5, (-, 2, 4))$

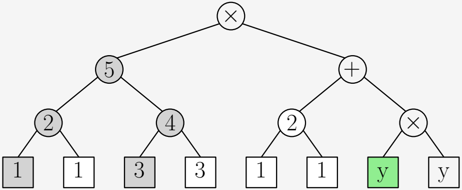
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0)), \quad (2, (\times, 1, 1)), \quad (3, (y, 0, 0)), \quad (4, (\times, 3, 3)), \quad (5, (-, 2, 4))$

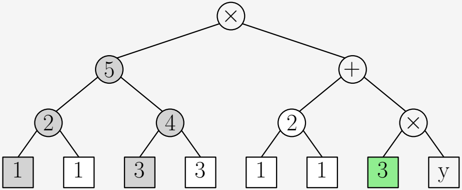
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0))$, $(2, (\times, 1, 1))$, $(3, (y, 0, 0))$, $(4, (\times, 3, 3))$, $(5, (-, 2, 4))$

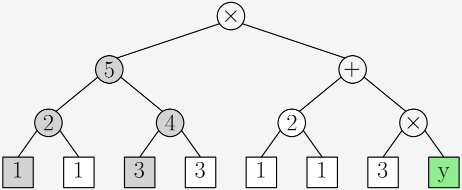
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1,(x,0,0)), \quad (2,(\times,1,1)), \quad (3,(y,0,0)), \quad (4,(\times,3,3)), \quad (5,(-,2,4))$

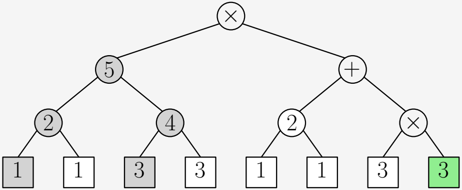
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0)), \quad (2, (\times, 1, 1)), \quad (3, (y, 0, 0)), \quad (4, (\times, 3, 3)), \quad (5, (-, 2, 4))$

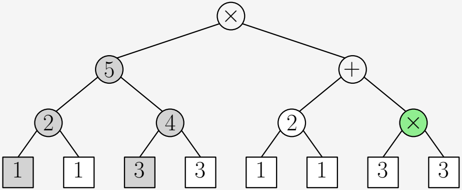
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0)), \quad (2, (\times, 1, 1)), \quad (3, (y, 0, 0)), \quad (4, (\times, 3, 3)), \quad (5, (-, 2, 4))$

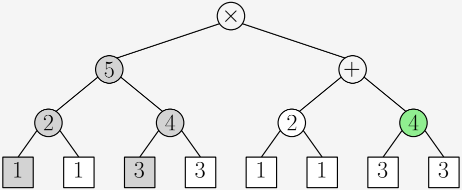
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0))$, $(2, (\times, 1, 1))$, $(3, (y, 0, 0))$, $(4, (\times, 3, 3))$, $(5, (-, 2, 4))$

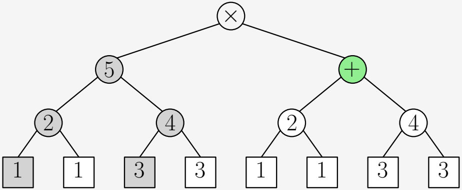
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0)),\quad (2, (\times, 1, 1)),\quad (3, (y, 0, 0)),\quad (4, (\times, 3, 3)),\quad (5, (-, 2, 4))$

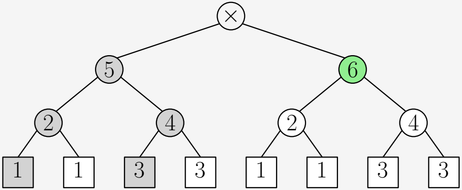
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0))$, $(2, (\times, 1, 1))$, $(3, (y, 0, 0))$, $(4, (\times, 3, 3))$, $(5, (-, 2, 4))$, $(6, (+, 2, 4))$

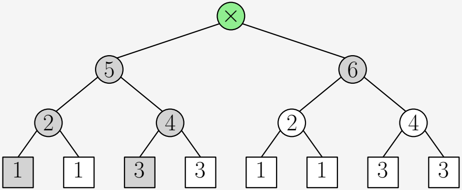
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0))$, $(2, (\times, 1, 1))$, $(3, (y, 0, 0))$, $(4, (\times, 3, 3))$, $(5, (-, 2, 4))$, $(6, (+, 2, 4))$

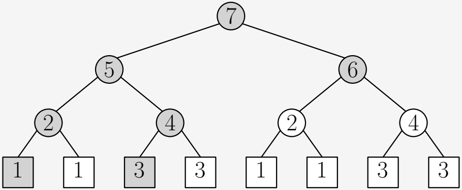
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0))$, $(2, (\times, 1, 1))$, $(3, (y, 0, 0))$, $(4, (\times, 3, 3))$, $(5, (-, 2, 4))$, $(6, (+, 2, 4))$, $(7, (\times, 5, 6))$

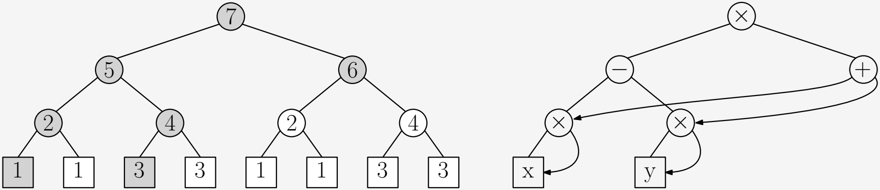
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1,(x,0,0))$, $(2,(\times,1,1))$, $(3,(y,0,0))$, $(4,(\times,3,3))$, $(5,(-,2,4))$,
$(6,(+,2,4))$, $(7,(\times,5,6))$

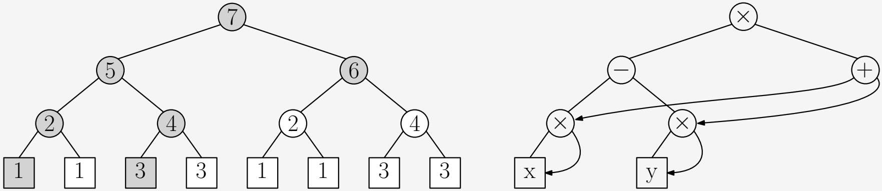
Motivation: Efficiently store redundant information

Example

Consider the labeled tree necessary to store the arithmetic expression

```
(* (- (* x x) (* y y)) (+ (* x x) (* y y)))
```

which represents $(x^2 - y^2)(x^2 + y^2)$.



$(1, (x, 0, 0))$, $(2, (\times, 1, 1))$, $(3, (y, 0, 0))$, $(4, (\times, 3, 3))$, $(5, (-, 2, 4))$,
$(6, (+, 2, 4))$, $(7, (\times, 5, 6))$

Definition

Compacted tree is the directed acyclic graph computed by this procedure.

Compacted trees

- Important property: **Subtrees are unique**
- Efficient algorithm to compute compacted tree: expected time $\mathcal{O}(n)$
- Analyzed by [Flajolet, Sipala, Steyaert 1990]: A tree of size n has a compacted form of expected size

$$C \frac{n}{\sqrt{\log n}},$$

  where C is explicit related to the type of trees and the statistical model.
- Applications:
  - **XML-Compression** [Bousquet-Mélou, Lohrey, Maneth, Noeth 2015]
  - **Compilers** [Aho, Sethi, Ullman 1986]
  - **LISP** [Goto 1974]
  - **Data storage** [Meinel, Theobald 1998], [Knuth 1968], etc.

Reverse question

How many compacted trees of (compacted) size n exist?

Main result compacted trees

A *stretched exponential* μ^{n^σ} appears!

Theorem

The number of compacted binary trees satisfy for $n \to \infty$

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

with $a_1 \approx -2.338$*: largest root of the Airy function* $\mathrm{Ai}(x) = \frac{1}{\pi}\int_0^\infty \cos\left(\frac{t^3}{3} + xt\right) dt$.

Main result compacted trees

A *stretched exponential* μ^{n^σ} appears!

Theorem

The number of compacted binary trees satisfy for $n \to \infty$

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

with $a_1 \approx -2.338$*: largest root of the Airy function* $\mathrm{Ai}(x) = \frac{1}{\pi}\int_0^\infty \cos\left(\frac{t^3}{3} + xt\right) dt$.

Conjecture

Experimentally we find

$$c_n \sim \gamma_c n! 4^n e^{3a_1 n^{1/3}} n^{3/4},$$

where

$$\gamma_c \approx 173.12670485.$$

What is a DFA?

Deterministic finite automata (DFA)

DFA on alphabet $\{a, b\}$

Graph with

- two outgoing edges from each node (state), labelled *a* and *b*
- An initial state q_0
- A set F of *final states* (coloured green).

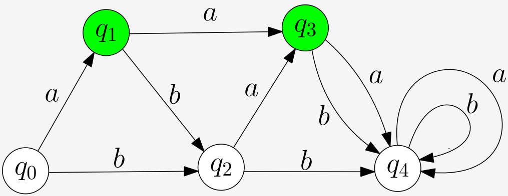


Figure: A DFA.

Deterministic finite automata (DFA)

DFA on alphabet $\{a, b\}$

Graph with

- two outgoing edges from each node (state), labelled a and b
- An initial state q_0
- A set F of *final states* (coloured green).

Properties

- **Language:** the set of accepted words
- **Minimal:** no DFA with fewer states accepts the same language
- **Acyclic:** no cycles (except loops at unique sink)

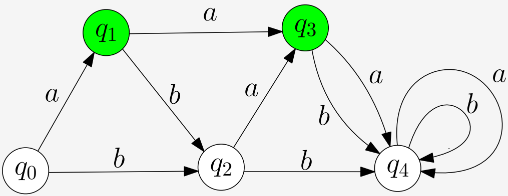
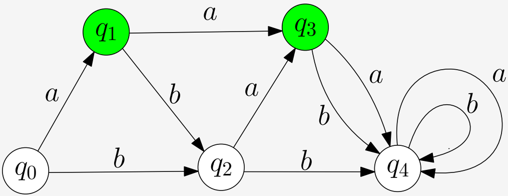


Figure: A DFA. This is the minimal DFA recognising the language $\{a, aa, ba, aba\}$.

Counting minimal acyclic DFAs

This work: Asymptotics of the numbers m_n of minimal, acyclic DFAs on a binary alphabet with $n+1$ nodes.

- Studied by Domaratzki, Kisman, Shallit, and Liskovets between 2002 and 2006
- Best bounds were out by an exponential factor
- We gave upper and lower bounds differing by a $\Theta(n^{1/4})$ factor, by relating the DFAs to compacted trees.



Main result minimal DFAs

A *stretched exponential* μ^{n^σ} appears again!

Theorem

The number m_n of minimal DFAs recognizing a finite binary for $n \to \infty$

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

with $a_1 \approx -2.338$: largest root of the Airy function $\mathrm{Ai}(x) = \frac{1}{\pi}\int_0^\infty \cos\left(\frac{t^3}{3} + xt\right) dt$.

Main result minimal DFAs

A *stretched exponential* μ^{n^σ} appears again!

Theorem

The number m_n *of minimal DFAs recognizing a finite binary for* $n \to \infty$

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

with $a_1 \approx -2.338$*: largest root of the Airy function* $\mathrm{Ai}(x) = \frac{1}{\pi}\int_0^\infty \cos\left(\frac{t^3}{3} + xt\right) dt$.

Conjecture

Experimentally we find

$$m_n \sim \gamma n! 8^n e^{3a_1 n^{1/3}} n^{7/8},$$

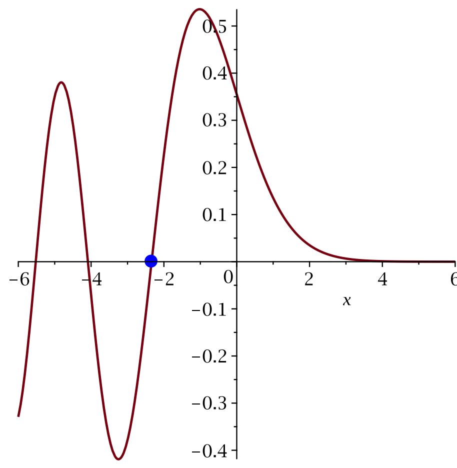
where

$$\gamma \approx 76.438160702.$$

What is the Airy function?

Properties

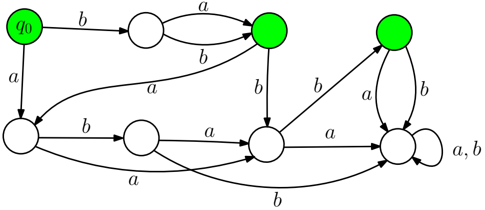
- $\mathrm{Ai}(x) = \frac{1}{\pi} \int_0^\infty \cos\left(\frac{t^3}{3} + xt\right) dt$
- Largest root $a_1 \approx -2.338$
- $\lim_{x\to\infty} \mathrm{Ai}(x) = 0$
- Also defined by $\mathrm{Ai}''(x) = x\mathrm{Ai}(x)$



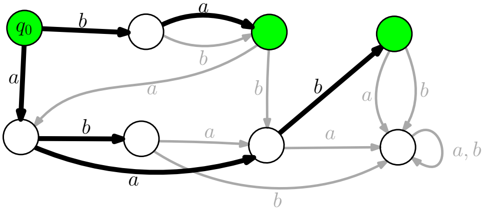
- [Banderier, Flajolet, Schaeffer, Soria 2001]: Random Maps
- [Flajolet, Louchard 2001]: Brownian excursion area

Bijection to decorated paths

Bijection to decorated paths

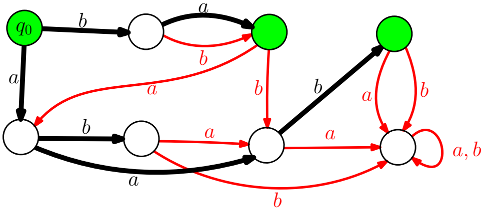


Bijection to decorated paths



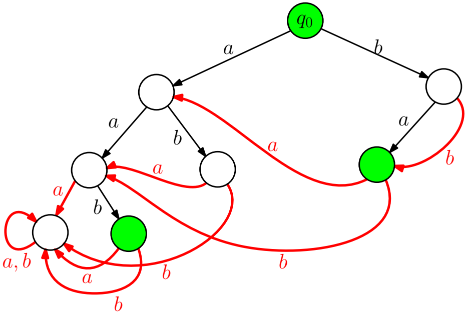
- Highlight spanning tree given by depth first search (ignoring the sink)
- I.e., black path to each vertex is first in lexicographic order
- Colour other edges red
- Draw as a binary tree with *a* edges pointing left and *b* edges pointing right

Bijection to decorated paths



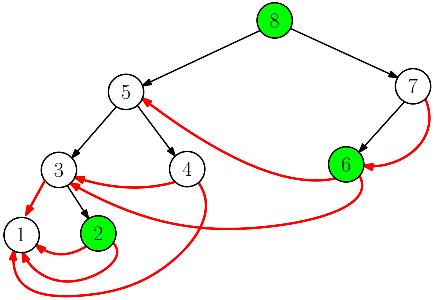
- Highlight spanning tree given by depth first search (ignoring the sink)
- I.e., black path to each vertex is first in lexicographic order
- Colour other edges red
- Draw as a binary tree with *a* edges pointing left and *b* edges pointing right

Bijection to decorated paths



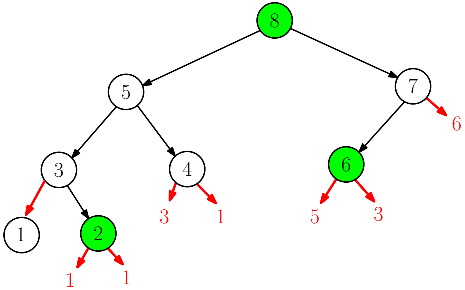
- Highlight spanning tree given by depth first search (ignoring the sink)
- I.e., black path to each vertex is first in lexicographic order
- Colour other edges red
- Draw as a binary tree with *a* edges pointing left and *b* edges pointing right

Bijection to decorated paths



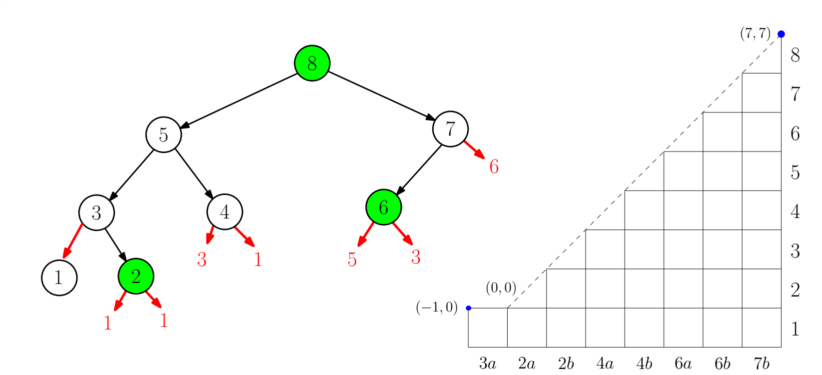
- Label nodes in post-order. By construction red edges point from a larger number to a smaller number
- $\rightarrow$ Label pointers

Bijection to decorated paths

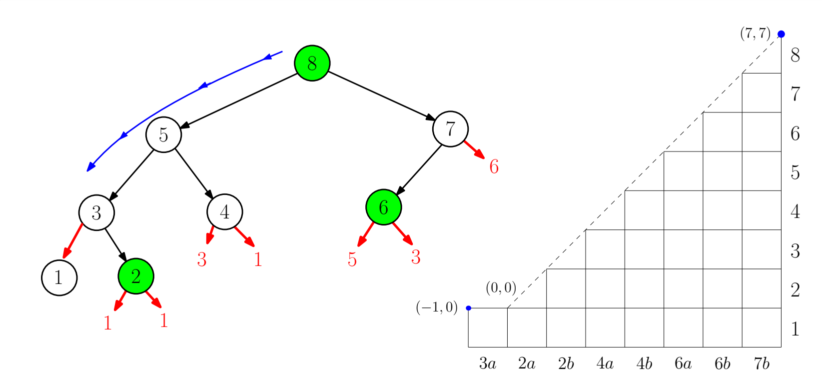


- Label nodes in post-order. By construction red edges point from a larger number to a smaller number
- $\rightarrow$ Label pointers

Bijection to decorated paths



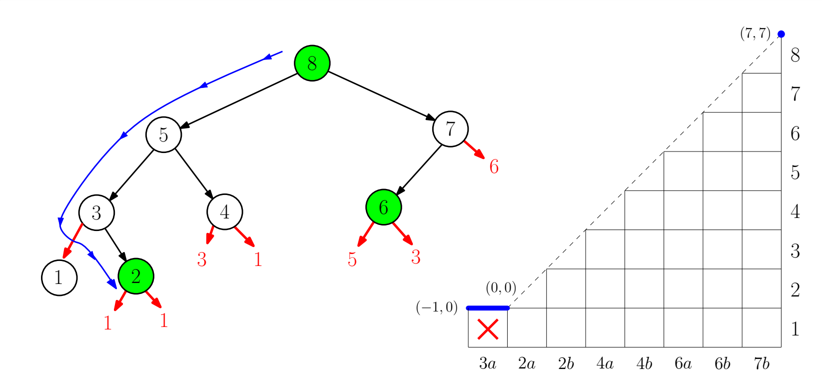
Bijection to decorated paths



When the tree traversal...

- goes up: add up step with colour matching the corresponding node.
- passes a pointer:
  - add horizontal step
  - mark box corresponding to pointer label

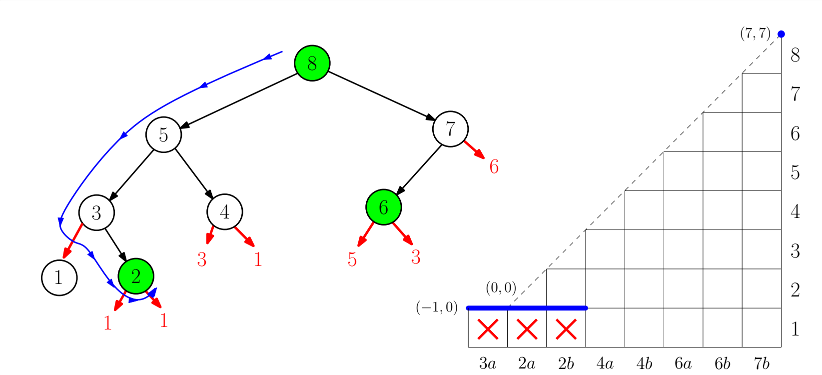
Bijection to decorated paths



When the tree traversal...

- goes up: add up step with colour matching the corresponding node.
- passes a pointer:
  - add horizontal step
  - mark box corresponding to pointer label

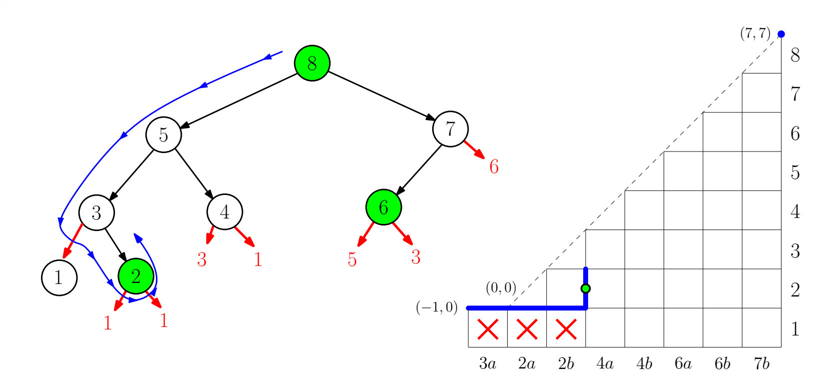
Bijection to decorated paths



When the tree traversal...

- goes up: add up step with colour matching the corresponding node.
- passes a pointer:
  - add horizontal step
  - mark box corresponding to pointer label

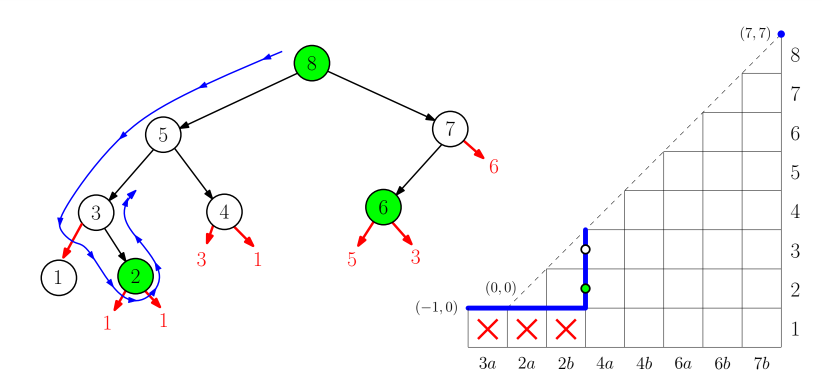
Bijection to decorated paths



When the tree traversal...

- goes up: add up step with colour matching the corresponding node.
- passes a pointer:
  - add horizontal step
  - mark box corresponding to pointer label

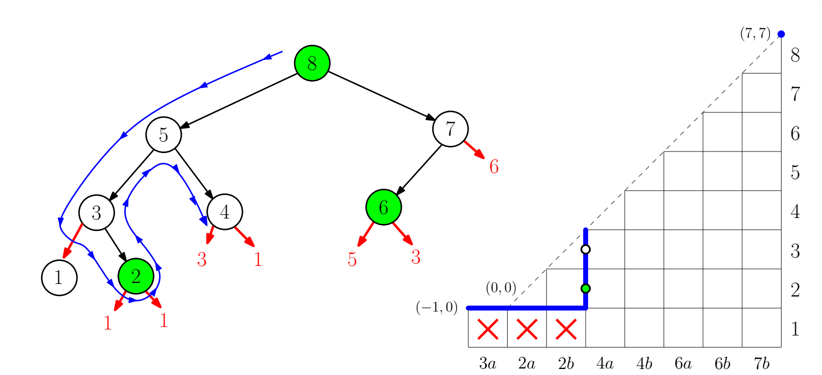
Bijection to decorated paths



When the tree traversal...

- goes up: add up step with colour matching the corresponding node.
- passes a pointer:
  - add horizontal step
  - mark box corresponding to pointer label

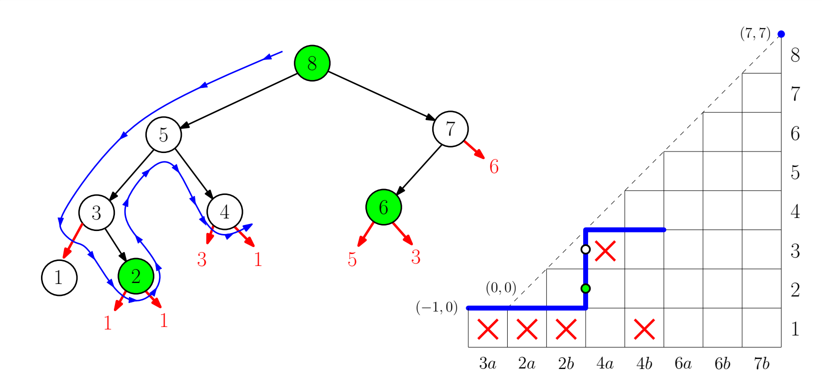
Bijection to decorated paths



When the tree traversal...

- goes up: add up step with colour matching the corresponding node.
- passes a pointer:
  - add horizontal step
  - mark box corresponding to pointer label

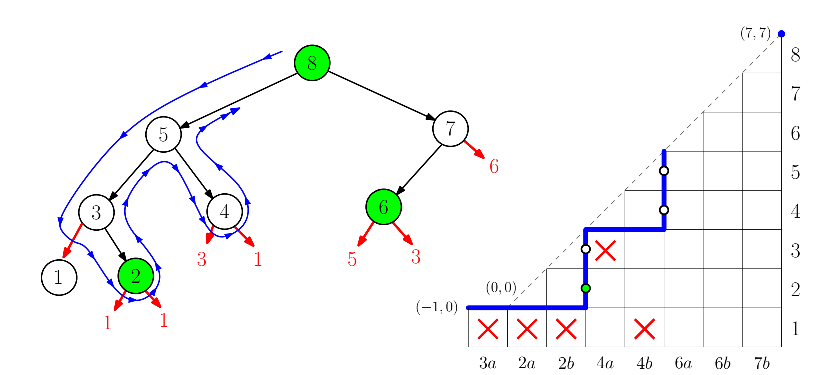
Bijection to decorated paths



When the tree traversal...

- goes up: add up step with colour matching the corresponding node.
- passes a pointer:
  - add horizontal step
  - mark box corresponding to pointer label

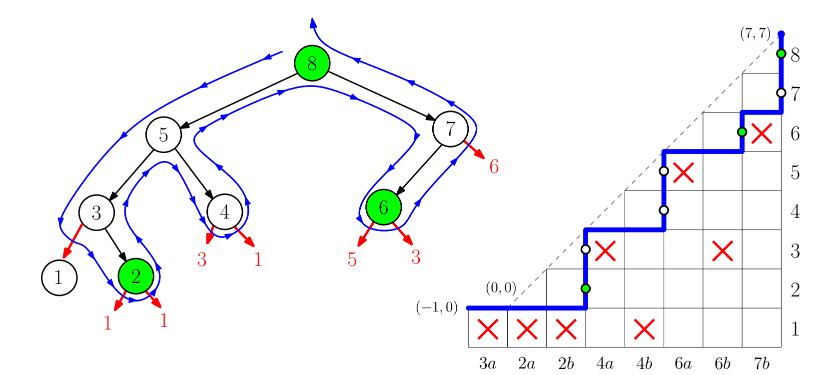
Bijection to decorated paths



When the tree traversal...

- goes up: add up step with colour matching the corresponding node.
- passes a pointer:
  - add horizontal step
  - mark box corresponding to pointer label

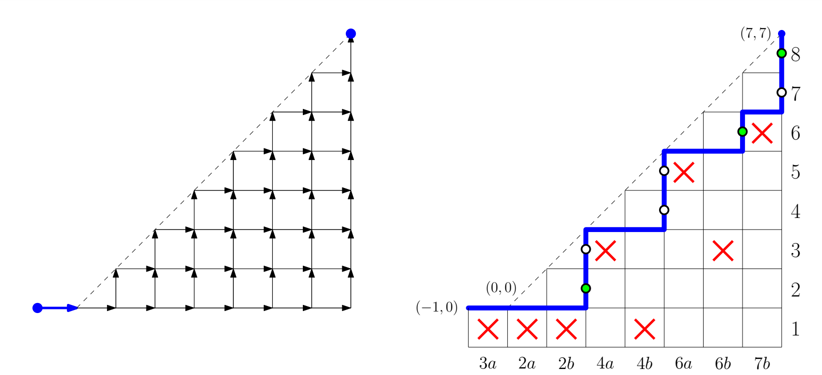
Bijection to decorated paths



When the tree traversal...

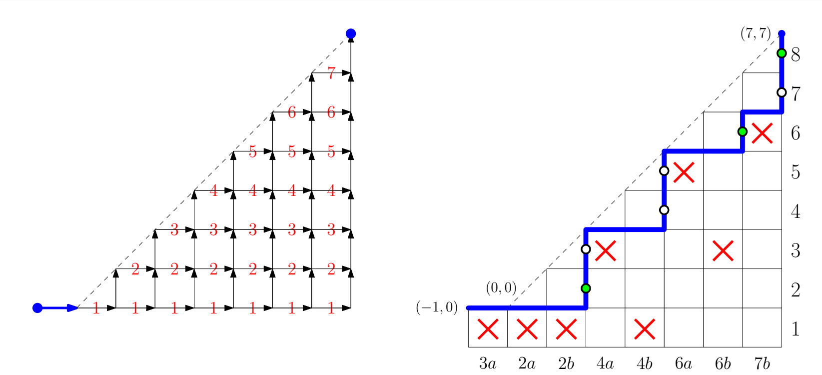
- goes up: add up step with colour matching the corresponding node.
- passes a pointer:
  - add horizontal step
  - mark box corresponding to pointer label

Decorated paths



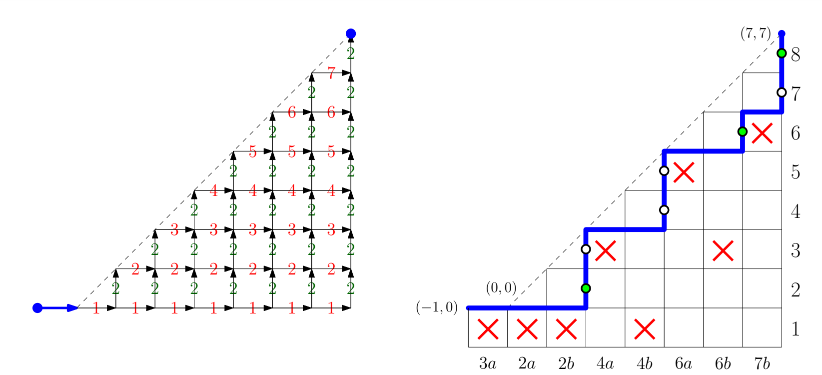
- Path starts at $(-1, 0)$ and ends at (n, n)
- Path stays below diagonal (after first step)
- One box is marked below each horizontal step
- Each vertical step is colored white or green

Decorated paths



- Path starts at $(-1, 0)$ and ends at (n, n)
- Path stays below diagonal (after first step)
- One box is marked below each horizontal step
- Each vertical step is colored white or green

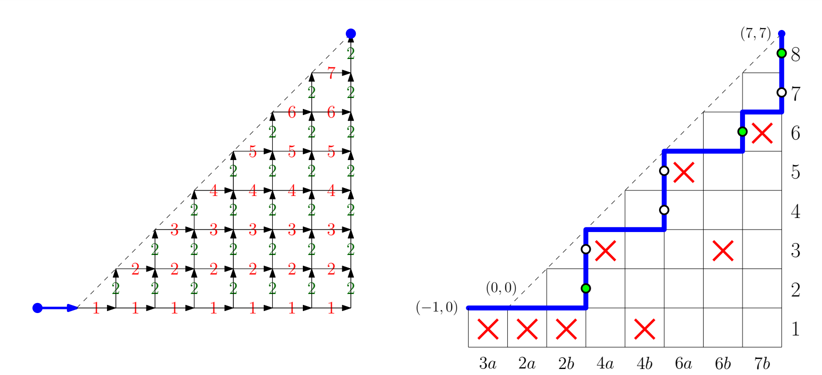
Decorated paths



- Path starts at $(-1, 0)$ and ends at (n, n)
- Path stays below diagonal (after first step)
- One box is marked below each horizontal step
- Each vertical step is colored white or green

By the bijection: The number of these paths is the number d_n of acyclic DFAs with $n + 1$ nodes.

Decorated paths



Recurrence: Denote by $a_{n,m}$ the number of paths ending at (n, m).

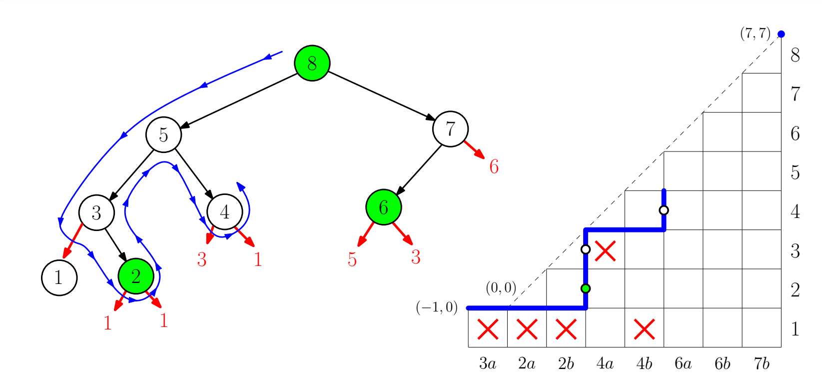
$$a_{n,m} = 2a_{n,m-1} + (m+1)a_{n-1,m}, \qquad \text{for } n \geq m$$

$$a_{-1,0} = 1.$$

By the bijection: $d_n = a_{n,n}$ is the number of acyclic DFAs with $n+1$ nodes.

What about minimality?

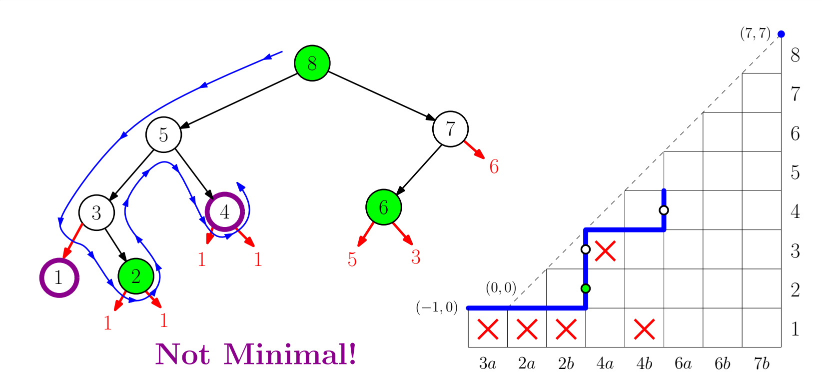
Minimal acyclic DFAs



For the DFA to be minimal, no state can be equivalent to a previous state:

- only possible if the new node is a leaf.
- If leaf is labeled $m + 1$, then m choices of pointer labels and state color must be avoided.
- Leaf corresponds to $\rightarrow \rightarrow \uparrow$ in path.

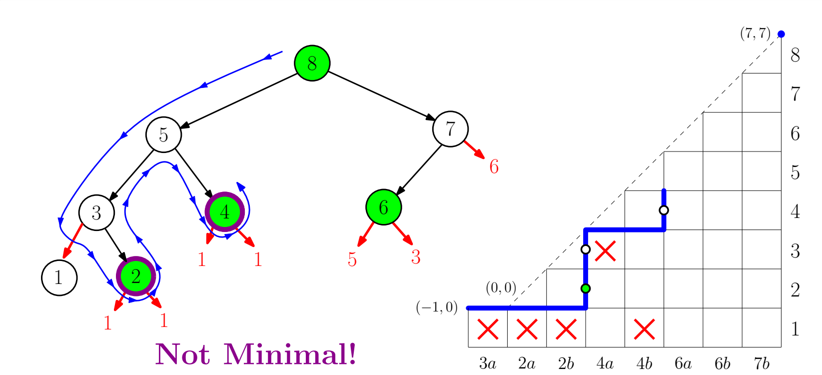
Minimal acyclic DFAs



For the DFA to be minimal, no state can be equivalent to a previous state:

- only possible if the new node is a leaf.
- If leaf is labeled $m + 1$, then m choices of pointer labels and state color must be avoided.
- Leaf corresponds to $\rightarrow \rightarrow \uparrow$ in path.

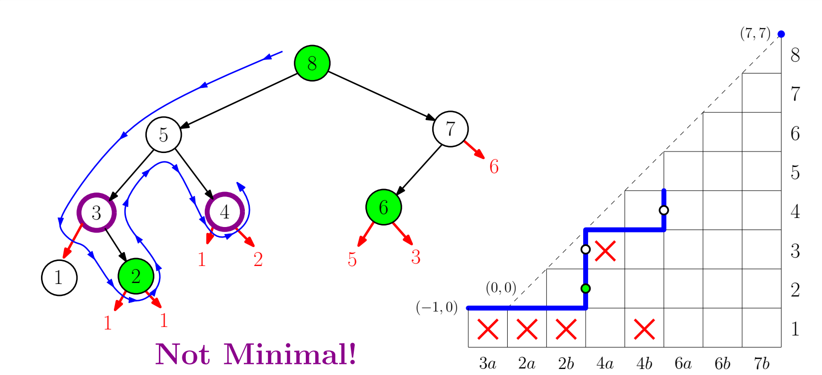
Minimal acyclic DFAs



For the DFA to be minimal, no state can be equivalent to a previous state:

- only possible if the new node is a leaf.
- If leaf is labeled $m + 1$, then m choices of pointer labels and state color must be avoided.
- Leaf corresponds to $\rightarrow \rightarrow \uparrow$ in path.

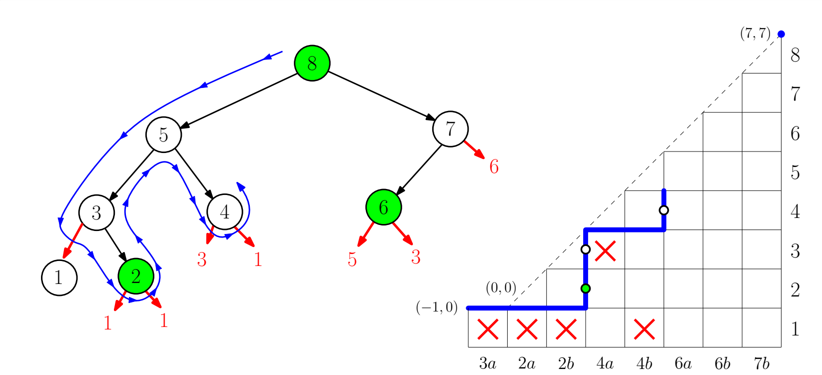
Minimal acyclic DFAs



For the DFA to be minimal, no state can be equivalent to a previous state:

- only possible if the new node is a leaf.
- If leaf is labeled $m + 1$, then m choices of pointer labels and state color must be avoided.
- Leaf corresponds to $\rightarrow \rightarrow \uparrow$ in path.

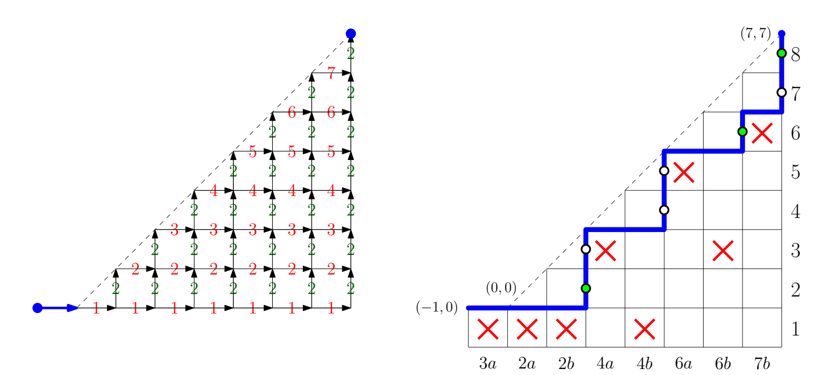
Minimal acyclic DFAs



For the DFA to be minimal, no state can be equivalent to a previous state:

- only possible if the new node is a leaf.
- If leaf is labeled $m + 1$, then m choices of pointer labels and state color must be avoided.
- Leaf corresponds to $\rightarrow \rightarrow \uparrow$ in path.

Recurrence for minimal DFAs



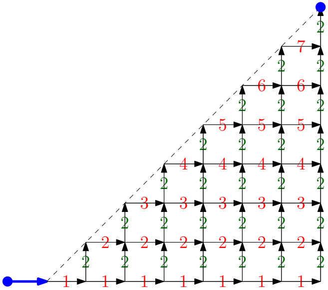
Recurrence: Denote by $b_{n,m}$ the number of paths ending at (n, m).

$$b_{n,m} = 2b_{n,m-1} + (m+1)b_{n-1,m} - mb_{n-2,m-1}, \qquad \text{for } n \geq m,$$

$$b_{-1,0} = 1.$$

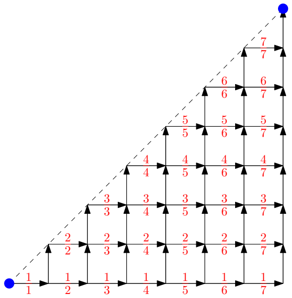
Now: $m_n = b_{n,n}$ is the number of minimal acyclic DFAs with $n+1$ nodes.

Transforming the recurrence for minimal DFAs



Transformation: $$e_{n+m,n-m} = \frac{1}{n!2^{m-1}} b_{n,m}.$$

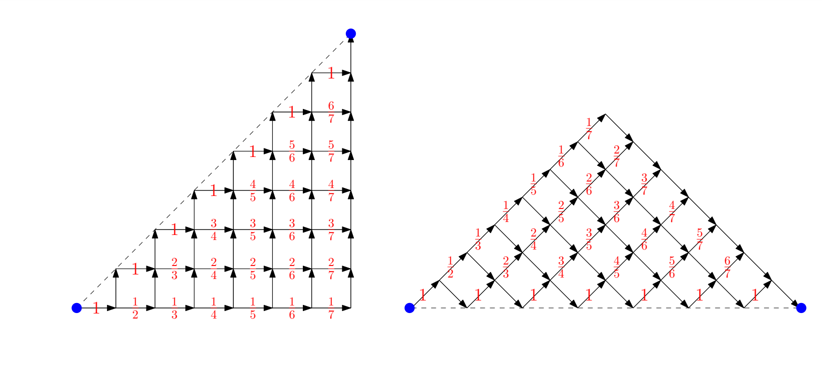
Transforming the recurrence for minimal DFAs



Transformation:

$$e_{n+m,n-m} = \frac{1}{n!2^{m-1}} b_{n,m}.$$

Transforming the recurrence for minimal DFAs



Transformation:

$$e_{n+m,n-m} = \frac{1}{n!2^{m-1}} b_{n,m}.$$

New recurrence:

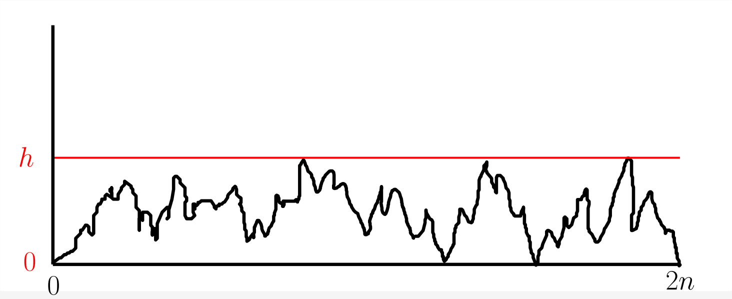
$$e_{n,m} = \frac{n-m+2}{n+m} e_{n-1,m-1} + e_{n-1,m+1} - \frac{n-m}{(n+m)(n+m-2)} e_{n-3,m-1}.$$

Now: $m_n = n!2^{n-1} e_{2n,0}$.

Heuristics

Side note: Pushed Dyck paths

Dyck paths of length $2n$ where paths of height h get weight 2^{-h}



Consider paths with max height n^α (for $0 < \alpha \leq 1/2$):

Number of paths $\approx 4^n e^{-c_1 n^{1-2\alpha}}$, **Weight** $= 2^{-n^\alpha} = e^{-\log(2) n^\alpha}$.

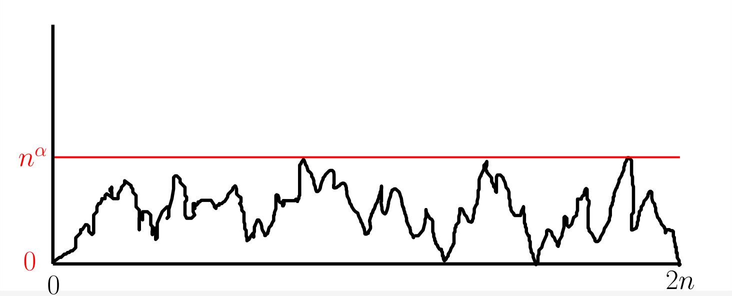
Weighted number of paths is $\approx 4^n e^{-c_1 n^{1-2\alpha} - \log(2) n^\alpha}$.

Maximum occurs when $\alpha = 1/3$ and is equal to $4^n e^{-c n^{1/3}}$.

Our case: weights decrease similarly with height so we expect similar behavior

Side note: Pushed Dyck paths

Dyck paths of length $2n$ where paths of height h get weight 2^{-h}



Consider paths with max height n^α (for $0 < \alpha \leq 1/2$):

Number of paths $\approx 4^n e^{-c_1 n^{1-2\alpha}}$, **Weight** $= 2^{-n^\alpha} = e^{-\log(2) n^\alpha}$.

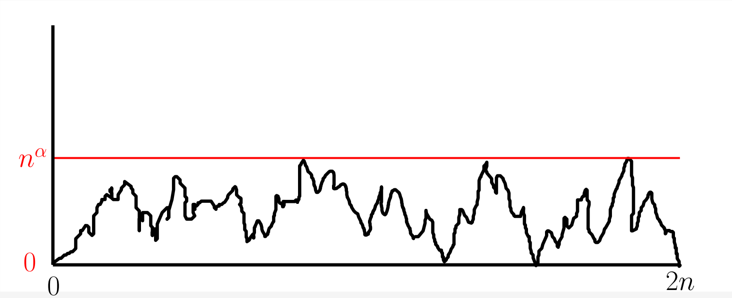
Weighted number of paths is $\approx 4^n e^{-c_1 n^{1-2\alpha} - \log(2) n^\alpha}$.

Maximum occurs when $\alpha = 1/3$ and is equal to $4^n e^{-cn^{1/3}}$.

Our case: weights decrease similarly with height so we expect similar behavior

Side note: Pushed Dyck paths

Dyck paths of length $2n$ where paths of height h get weight 2^{-h}



Consider paths with max height n^α (for $0 < \alpha \leq 1/2$):

Number of paths $\approx 4^n e^{-c_1 n^{1-2\alpha}}$, **Weight** $= 2^{-n^\alpha} = e^{-\log(2) n^\alpha}$.

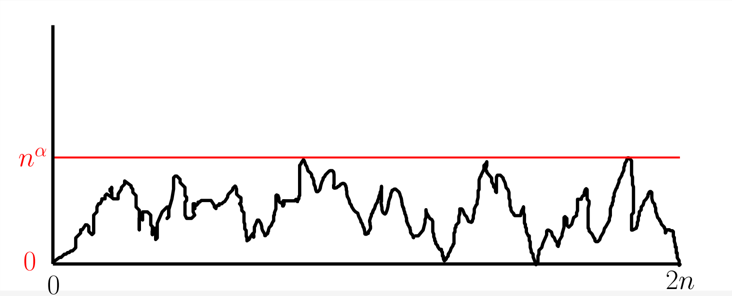
Weighted number of paths is $\approx 4^n e^{-c_1 n^{1-2\alpha} - \log(2) n^\alpha}$.

Maximum occurs when $\alpha = 1/3$ and is equal to $4^n e^{-cn^{1/3}}$.

Our case: weights decrease similarly with height so we expect similar behavior

Side note: Pushed Dyck paths

Dyck paths of length $2n$ where paths of height h get weight 2^{-h}



Consider paths with max height n^α (for $0 < \alpha \leq 1/2$):

Number of paths $\approx 4^n e^{-c_1 n^{1-2\alpha}}$, **Weight** $= 2^{-n^\alpha} = e^{-\log(2) n^\alpha}$.

Weighted number of paths is $\approx 4^n e^{-c_1 n^{1-2\alpha} - \log(2) n^\alpha}$.

Maximum occurs when $\alpha = 1/3$ and is equal to $4^n e^{-c n^{1/3}}$.

Our case: weights decrease similarly with height so we expect similar behavior

Heuristics

We want to understand $e_{n,m}$ for large n.

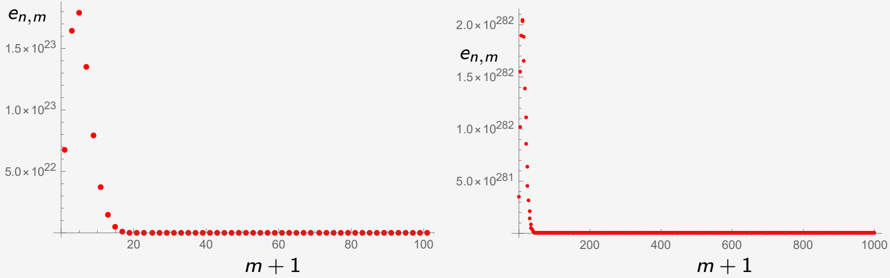


Figure: Plots of $e_{n,m}$ against $m+1$. **Left:** $n = 100$, **Right:** $n = 1000$.

- Let's zoom in to the left (small m) where interesting things are happening.
- It seems to be converging to something...

Guess: $e_{n,m} \approx h(n) f\left(\frac{m+1}{g(n)}\right)$. Moreover, we guess $g(n) = \sqrt[3]{n}$.

Heuristics

We want to understand $e_{n,m}$ for large n.

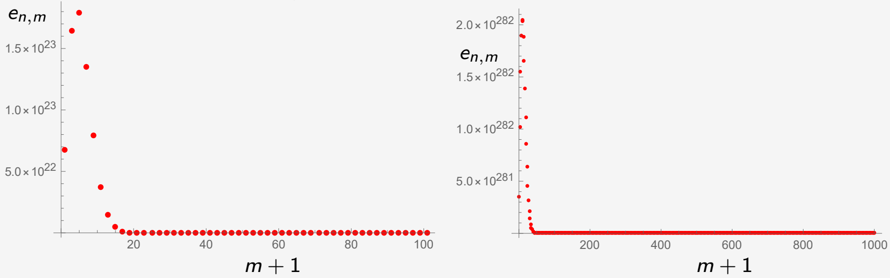


Figure: Plots of $e_{n,m}$ against $m+1$. **Left:** $n = 100$, **Right:** $n = 1000$.

- Let's zoom in to the left (small m) where interesting things are happening.
- It seems to be converging to something...

Guess: $e_{n,m} \approx h(n) f\left(\frac{m+1}{g(n)}\right)$. Moreover, we guess $g(n) = \sqrt[3]{n}$.

Heuristics

We want to understand $e_{n,m}$ for large n.

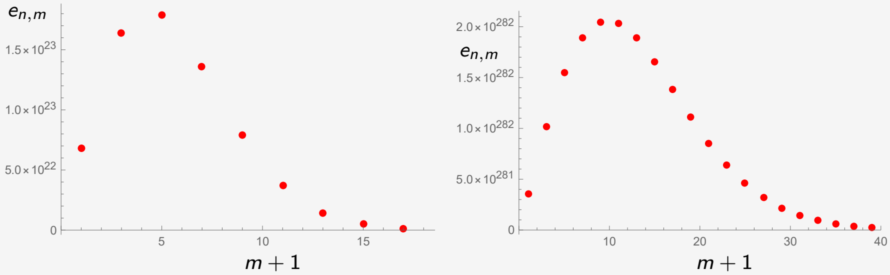


Figure: Plots of $e_{n,m}$ against $m+1$. **Left:** $n = 100$, **Right:** $n = 1000$.

- Let's zoom in to the left (small m) where interesting things are happening.
- It seems to be converging to something...

Guess: $e_{n,m} \approx h(n) f\left(\frac{m+1}{g(n)}\right)$. Moreover, we guess $g(n) = \sqrt[3]{n}$.

Heuristics

We want to understand $e_{n,m}$ for large n.

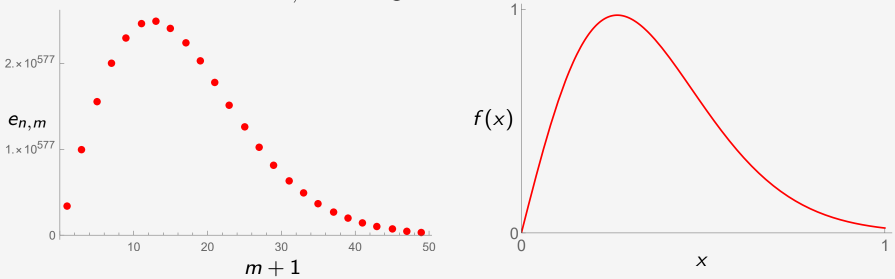


Figure: **Left:** Plot of $e_{n,m}$ against $m+1$ for $n = 2000$. **Right:** Limiting function $f(x)$.

- Let's zoom in to the left (small m) where interesting things are happening.
- It seems to be converging to something...

Guess: $e_{n,m} \approx h(n) f\left(\frac{m+1}{g(n)}\right)$. Moreover, we guess $g(n) = \sqrt[3]{n}$.

Heuristics

We want to understand $e_{n,m}$ for large n.

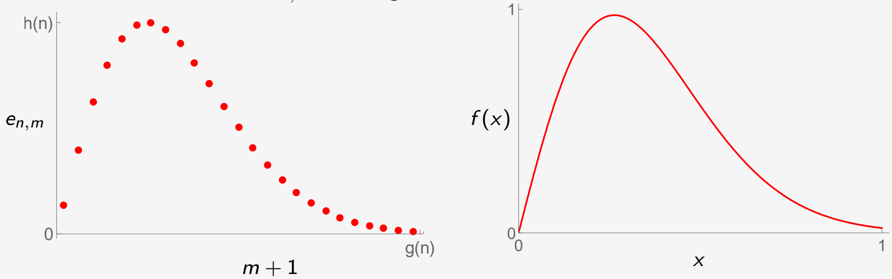


Figure: **Left:** Plot of $e_{n,m}$ against $m+1$ for $n = 2000$. **Right:** Limiting function $f(x)$.

- Let's zoom in to the left (small m) where interesting things are happening.
- It seems to be converging to something...

Guess: $e_{n,m} \approx h(n) f\left(\frac{m+1}{g(n)}\right)$. Moreover, we guess $g(n) = \sqrt[3]{n}$.

Heuristics

We want to understand $e_{n,m}$ for large n.

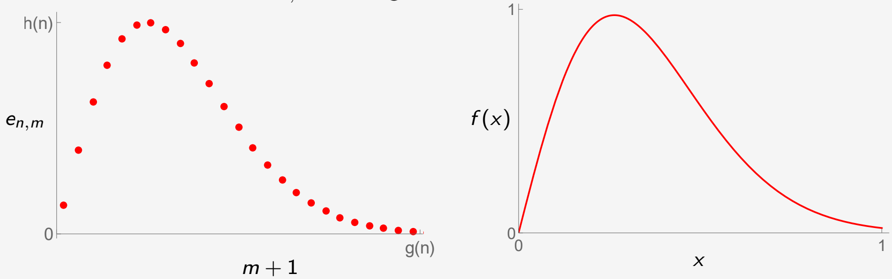


Figure: **Left:** Plot of $e_{n,m}$ against $m+1$ for $n = 2000$. **Right:** Limiting function $f(x)$.

- Let's zoom in to the left (small m) where interesting things are happening.
- It seems to be converging to something...

Guess: $e_{n,m} \approx h(n) f\left(\frac{m+1}{\sqrt[3]{n}}\right)$. Moreover, we guess $g(n) = \sqrt[3]{n}$.

Heuristic analysis of weighted paths

Recurrence:

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

Guess: $e_{n,m} \approx h(n) f\left(\frac{m+1}{\sqrt[3]{n}}\right).$

Heuristic analysis of weighted paths

Recurrence:

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

Guess: $e_{n,m} \approx h(n) f\left(\frac{m+1}{\sqrt[3]{n}}\right).$

- Substitute into recurrence and set $m = \kappa\sqrt[3]{n} - 1$:

$$s_n := \frac{h(n)}{h(n-1)} \approx 2 + \frac{f''(\kappa) - 2\kappa f(\kappa)}{f(\kappa)} n^{-2/3} + O(n^{-1})$$

Heuristic analysis of weighted paths

Recurrence:

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

Guess: $e_{n,m} \approx h(n) f\left(\frac{m+1}{\sqrt[3]{n}}\right).$

- Substitute into recurrence and set $m = \kappa\sqrt[3]{n} - 1$:

$$s_n := \frac{h(n)}{h(n-1)} \approx 2 + \frac{f''(\kappa) - 2\kappa f(\kappa)}{f(\kappa)} n^{-2/3} + O(n^{-1})$$

- Solution (assuming equality above):

$$s_n = 2 + c n^{-2/3} + O(n^{-1}) \qquad h(n) \approx 2^n e^{\frac{3c}{2} n^{1/3}}$$

$$f''(\kappa) = (2\kappa + c) f(\kappa)$$

Where c is constant

Heuristic analysis of weighted paths

Recurrence:

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

Guess: $e_{n,m} \approx h(n) f\left(\frac{m+1}{\sqrt[3]{n}}\right).$

- Substitute into recurrence and set $m = \kappa\sqrt[3]{n} - 1$:

$$s_n := \frac{h(n)}{h(n-1)} \approx 2 + \frac{f''(\kappa) - 2\kappa f(\kappa)}{f(\kappa)} n^{-2/3} + O(n^{-1})$$

- Solution (assuming equality above):

$$s_n = 2 + c n^{-2/3} + O(n^{-1}) \quad \Rightarrow \quad h(n) \approx 2^n e^{\frac{3c}{2} n^{1/3}}$$

$$f''(\kappa) = (2\kappa + c) f(\kappa) \quad \Rightarrow \quad f(\kappa) = \mathrm{Ai}(2^{-2/3}(2\kappa + c))$$

Where c is constant and Ai is the Airy function.

Heuristic analysis of weighted paths

Recurrence:

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

Guess: $e_{n,m} \approx h(n) f\left(\frac{m+1}{\sqrt[3]{n}}\right).$

- Substitute into recurrence and set $m = \kappa\sqrt[3]{n} - 1$:

$$s_n := \frac{h(n)}{h(n-1)} \approx 2 + \frac{f''(\kappa) - 2\kappa f(\kappa)}{f(\kappa)} n^{-2/3} + O(n^{-1})$$

- Solution (assuming equality above):

$$s_n = 2 + c n^{-2/3} + O(n^{-1}) \qquad \Rightarrow \qquad h(n) \approx 2^n e^{\frac{3c}{2} n^{1/3}}$$

$$f''(\kappa) = (2\kappa + c) f(\kappa) \qquad \Rightarrow \qquad f(\kappa) = \mathrm{Ai}(2^{-2/3}(2\kappa + c))$$

  Where c is constant and Ai is the Airy function.

- Boundary condition $e_{n,-1} = 0$. Then $f(0) = 0$ implies $c = 2^{2/3} a_1$, where $a_1 \approx -2.338$ satisfies $\mathrm{Ai}(a_1) = 0$.

Inductive proof

Proof method

Recall:

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

Number of minimal acyclic DFAs is $m_n = 2^{n-1} n! e_{2n,0}$.

Method:

Find sequences $A_{n,k}$ and $B_{n,k}$ with the same asymptotic form, such that

$$A_{n,k} \leq e_{n,k} \leq B_{n,k},$$

for all k and all n large enough.

Proof method

Recall:

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

Number of minimal acyclic DFAs is $m_n = 2^{n-1} n! e_{2n,0}$.

Method:
Find sequences $A_{n,k}$ and $B_{n,k}$ with the same asymptotic form, such that

$$A_{n,k} \leq e_{n,k} \leq B_{n,k},$$

for all k and all n large enough.

How to find them?

1. Use heuristics
2. Fiddle until they satisfy the recurrence of $e_{n,k}$ with the equalities replaced by inequalities:
   $$= \quad \longrightarrow \quad \leq \text{ and } \geq$$
3. Prove $A_{n,k} \leq e_{n,k} \leq B_{n,k}$ by induction.

Summary

Enumeration of minimal acyclic DFAs

1. Bijection to decorated paths
2. Recurrence for decorated paths
3. Heuristic analysis of recurrence
4. Inductive proof using heuristics

Lower bound:

$$m_n \geq \gamma_1 \, n! 8^n e^{3a_1 n^{1/3}} n^{7/8},$$

for some constant $\gamma_1 > 0$.

Summary

Enumeration of minimal acyclic DFAs

1. Bijection to decorated paths
2. Recurrence for decorated paths
3. Heuristic analysis of recurrence
4. Inductive proof using heuristics

Lower bound:

$$m_n \geq \gamma_1 \, n! 8^n e^{3a_1 n^{1/3}} n^{7/8},$$

for some constant $\gamma_1 > 0$.

Upper bound (similar proof):

$$m_n \leq \gamma_2 \, n! 8^n e^{3a_1 n^{1/3}} n^{7/8},$$

for some constant $\gamma_2 > 0$.

The end

Theorem

The number c_n compacted binary trees and the number m_n of minimal DFAs recognizing a finite binary language satisfy for $n \to \infty$

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

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

with $a_1 \approx -2.338$: largest root of the Airy function $\mathrm{Ai}(x) = \frac{1}{\pi}\int_0^\infty \cos\left(\frac{t^3}{3} + xt\right) dt$.

Further problems:

- Multiplicative constant? Does it exist?
- Other statistics: number of words? length of longest word? etc.
- Another solved problem by the method: counting the number of tree-child networks in phylogenomics [Fuchs, Yu, Zhang 2020]
  Does anyone have a tricky recurrence to try? (Suggested by referees: Piece-wise testable)

The end

Theorem

The number c_n compacted binary trees and the number m_n of minimal DFAs recognizing a finite binary language satisfy for $n \to \infty$

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

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

with $a_1 \approx -2.338$: largest root of the Airy function $\mathrm{Ai}(x) = \frac{1}{\pi}\int_0^\infty \cos\left(\frac{t^3}{3} + xt\right) dt.$

Further problems:

- Multiplicative constant? Does it exist?
- Other statistics: number of words? length of longest word? etc.
- Another solved problem by the method: counting the number of tree-child networks in phylogenomics [Fuchs, Yu, Zhang 2020]
  Does anyone have a tricky recurrence to try? (Suggested by referees: Piece-wise testable)

Backup

Other appearances of stretched exponentials

Known exactly:

- Pushed Dyck paths (Beaton and Mckay):

$$\sim \left(2^{8/3}3^{-1/2}\pi^{5/6}(\log 2)^{1/3}\right) 4^n e^{-3\left(\frac{\pi \log 2}{2}\right)^{2/3} n^{1/3}} n^{-5/6}$$

- Cogrowth sequence of lamplighter group (Revelle)

$$\approx \mu^n e^{-cn^{1/3}}$$

- Cogrowth sequences of other wreath products (Pittet and Saloffe-Coste):

$$\approx \mu^n e^{-c(\log n)^{2/3} n^{1/3}}$$

Conjectured:

- Permutations avoiding 1324 (Conway, Guttmann, and Zinn-Justin):

$$\approx \mu^n e^{-cn^{1/2}}$$

- Pushed self avoiding walks (Beaton, Guttmann, Jensen, and Lawler):

$$\approx \mu^n e^{-cn^{3/7}}$$

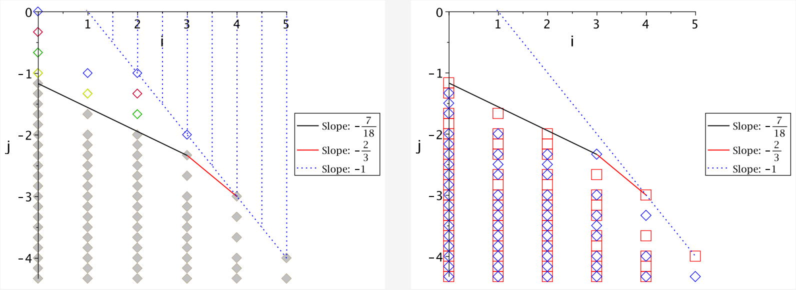
- Cogrowth sequence of Thompson's group F (Elvey Price and Guttmann):

$$\approx \mu^n e^{-cn^{0.5}(\log n)^{0.5}}$$

Technicalities

Lots of technicalities:

- Before induction, we have to remove the negative term from the recurrence, but we have to do so precisely for asymptotics to stay the same.
- We only prove bounds for small *m*; we prove that large *m* terms don't matter
- The lower bound is negative for very large *m*, so we have to be careful with induction
- We only prove the bounds for sufficiently large *n*, but this only makes a difference to the constant term. Proof involves colorful Newton polygons:



Relaxed trees: Proof idea – lower bound

Main idea

Suppose $(X_{n,m})_{n\geq m\geq 0}$ and $(s_n)_{n\geq 1}$ satisfy

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

for all sufficiently large n and all integers $m \in [0, n]$.

Define $(h_n)_{n\geq 0}$ by $h_0 = 1$ and $h_n = s_n h_{n-1}$; then prove that

$$X_{n,m}h_n \leq b_0 d_{n,m}$$

for some constant b_0 by induction:

Relaxed trees: Proof idea – lower bound

Main idea

Suppose $(X_{n,m})_{n\geq m\geq 0}$ and $(s_n)_{n\geq 1}$ satisfy

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

for all sufficiently large n and all integers $m \in [0, n]$.

Define $(h_n)_{n\geq 0}$ by $h_0 = 1$ and $h_n = s_n h_{n-1}$; then prove that

$$X_{n,m}h_n \leq b_0 d_{n,m}$$

for some constant b_0 by induction:

Relaxed trees: Proof idea – lower bound

Main idea

Suppose $(X_{n,m})_{n\geq m\geq 0}$ and $(s_n)_{n\geq 1}$ satisfy

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

for all sufficiently large n and all integers $m \in [0, n]$.

Define $(h_n)_{n\geq 0}$ by $h_0 = 1$ and $h_n = s_n h_{n-1}$; then prove that

$$X_{n,m}h_n \leq b_0 d_{n,m}$$

for some constant b_0 by induction:

$$X_{n,m}h_n \overset{(1)}{\leq} \frac{n-m+2}{n+m}X_{n-1,m-1}h_{n-1} + X_{n-1,m+1}h_{n-1}$$

$$\overset{\text{(Induction)}}{\leq} \frac{n-m+2}{n+m}b_0 d_{n-1,m-1} + b_0 d_{n-1,m+1}$$

$$\overset{\text{Rec. } d_{n,m}}{=} b_0 d_{n,m}.$$

Relaxed trees: Proof idea – lower bound

Main idea

Suppose $(X_{n,m})_{n\geq m\geq 0}$ and $(s_n)_{n\geq 1}$ satisfy

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

for all sufficiently large n and all integers $m \in [0, n]$.

Define $(h_n)_{n\geq 0}$ by $h_0 = 1$ and $h_n = s_n h_{n-1}$; then prove that

$$X_{n,m}h_n \leq b_0 d_{n,m}$$

for some constant b_0 by induction:

$$\begin{aligned} X_{n,m}h_n &\overset{(1)}{\leq} \frac{n-m+2}{n+m}X_{n-1,m-1}h_{n-1} + X_{n-1,m+1}h_{n-1} \\ &\overset{\text{(Induction)}}{\leq} \frac{n-m+2}{n+m}b_0 d_{n-1,m-1} + b_0 d_{n-1,m+1} \\ &\overset{\text{Rec. } d_{n,m}}{=} b_0 d_{n,m}. \end{aligned}$$

Relaxed trees: Proof idea – lower bound

Main idea

Suppose $(X_{n,m})_{n\geq m\geq 0}$ and $(s_n)_{n\geq 1}$ satisfy

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

for all sufficiently large n and all integers $m \in [0,n]$.

Define $(h_n)_{n\geq 0}$ by $h_0 = 1$ and $h_n = s_n h_{n-1}$; then prove that

$$X_{n,m}h_n \leq b_0 d_{n,m}$$

for some constant b_0 by induction:

$$\begin{aligned} X_{n,m}h_n &\overset{(1)}{\leq} \frac{n-m+2}{n+m}X_{n-1,m-1}h_{n-1} + X_{n-1,m+1}h_{n-1} \\ &\overset{\text{(Induction)}}{\leq} \frac{n-m+2}{n+m}b_0 d_{n-1,m-1} + b_0 d_{n-1,m+1} \\ &\overset{\text{Rec. } d_{n,m}}{=} b_0 d_{n,m}. \end{aligned}$$

Relaxed trees: Lower bound

Lemma

For all $n, m \geq 0$ *let*

$$\tilde{X}_{n,m} := \left(1 - \frac{2m^2}{3n} + \frac{m}{2n}\right) \mathrm{Ai}\left(a_1 + \frac{2^{1/3}(m+1)}{n^{1/3}}\right) \qquad \text{and}$$

$$\tilde{s}_n := 2 + \frac{2^{2/3}a_1}{n^{2/3}} + \frac{8}{3n} - \frac{1}{n^{7/6}}.$$

Then, for any $\varepsilon > 0$*, there exists an* $\tilde{n}_0$ *such that*

$$\tilde{X}_{n,m}\tilde{s}_n \leq \frac{n-m+2}{n+m}\tilde{X}_{n-1,m-1} + \tilde{X}_{n-1,m+1},$$

for all $n \geq \tilde{n}_0$ *and for all* $0 \leq m < n^{1-\varepsilon}$*.*

Define $X_{n,m} := \max\{\tilde{X}_{n,m}, 0\}$. Then,

1. $X_{n,m}\tilde{s}_n \leq \frac{n-m+2}{n+m}\tilde{X}_{n-1,m-1} + \tilde{X}_{n-1,m+1} \leq \frac{n-m+2}{n+m}X_{n-1,m-1} + X_{n-1,m+1}$;
2. $X_{n,m}\tilde{s}_n = 0 \leq \frac{n-m+2}{n+m}X_{n-1,m-1} + X_{n-1,m+1}$.

Relaxed trees: Lower bound

Lemma

For all $n, m \geq 0$ *let*

$$\tilde{X}_{n,m} := \left(1 - \frac{2m^2}{3n} + \frac{m}{2n}\right) \mathrm{Ai}\left(a_1 + \frac{2^{1/3}(m+1)}{n^{1/3}}\right) \qquad \textit{and}$$

$$\tilde{s}_n := 2 + \frac{2^{2/3} a_1}{n^{2/3}} + \frac{8}{3n} - \frac{1}{n^{7/6}}.$$

Then, for any $\varepsilon > 0$*, there exists an* $\tilde{n}_0$ *such that*

$$\tilde{X}_{n,m} \tilde{s}_n \leq \frac{n-m+2}{n+m} \tilde{X}_{n-1,m-1} + \tilde{X}_{n-1,m+1},$$

for all $n \geq \tilde{n}_0$ *and for all* $0 \leq m < n^{1-\varepsilon}$*.*

Define $X_{n,m} := \max\{\tilde{X}_{n,m}, 0\}$. Then,

1. $X_{n,m}\tilde{s}_n \leq \frac{n-m+2}{n+m}\tilde{X}_{n-1,m-1} + \tilde{X}_{n-1,m+1} \leq \frac{n-m+2}{n+m} X_{n-1,m-1} + X_{n-1,m+1}$;
2. $X_{n,m}\tilde{s}_n = 0 \qquad \leq \frac{n-m+2}{n+m} X_{n-1,m-1} + X_{n-1,m+1}.$

Relaxed trees: Lower bound

Lemma

For all $n, m \geq 0$ *let*

$$\tilde{X}_{n,m} := \left(1 - \frac{2m^2}{3n} + \frac{m}{2n}\right) \mathrm{Ai}\left(a_1 + \frac{2^{1/3}(m+1)}{n^{1/3}}\right) \qquad \textit{and}$$

$$\tilde{s}_n := 2 + \frac{2^{2/3}a_1}{n^{2/3}} + \frac{8}{3n} - \frac{1}{n^{7/6}}.$$

Then, for any $\varepsilon > 0$, *there exists an* $\tilde{n}_0$ *such that*

$$\tilde{X}_{n,m}\tilde{s}_n \leq \frac{n-m+2}{n+m}\tilde{X}_{n-1,m-1} + \tilde{X}_{n-1,m+1},$$

for all $n \geq \tilde{n}_0$ *and for all* $0 \leq m < n^{1-\varepsilon}$.

Define $X_{n,m} := \max\{\tilde{X}_{n,m}, 0\}$. Then,

1. $X_{n,m}\tilde{s}_n \leq \frac{n-m+2}{n+m}\tilde{X}_{n-1,m-1} + \tilde{X}_{n-1,m+1} \leq \frac{n-m+2}{n+m}X_{n-1,m-1} + X_{n-1,m+1}$;
2. $X_{n,m}\tilde{s}_n = 0 \leq \frac{n-m+2}{n+m}X_{n-1,m-1} + X_{n-1,m+1}$.