LIPN nonpermanent seminar, 16/01/26

# The "phoenix" and "fork anywhere" models of git graphs

Julien Clément, Julien Courtiel,
Rémi Maréchal, Martin Pépin

GREYC, Université de Caen Normandie

# MOTIVATION

"DAGs are too broad; feature branch git graphs could be broadened."

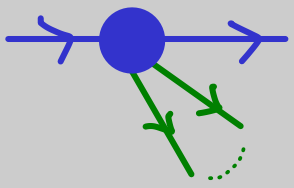
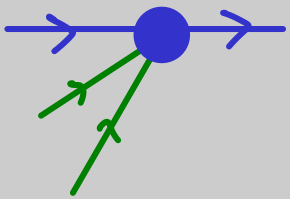
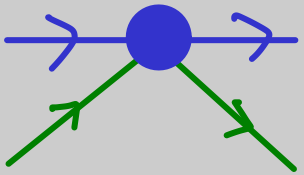
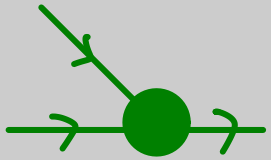
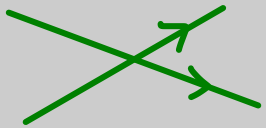
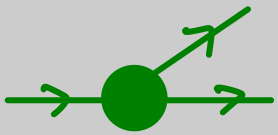
→ Come up with superclasses of feature branch git graphs, such that:

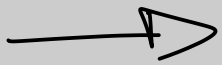
* they still describe realistic git workflows;
* their enumeration / random generation are not too difficult.

# STARTING POINT

Reminder

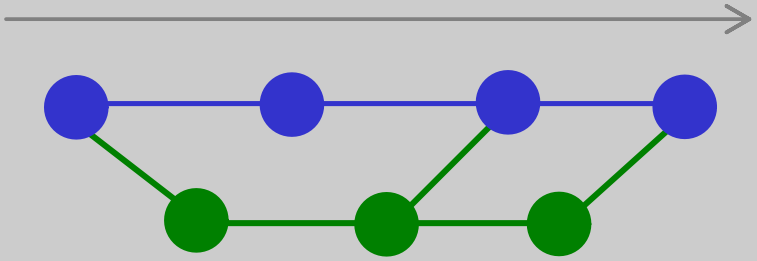
Feature branch graphs rules :

| OK | OK n't |
|---|---|
|  |  |
|  |  |
|  |  |



Broaden this class of graphs by allowing feature branches to be "reborn" when they go to "die" on the main branch,

e.g.



# PHOENIX GRAPHS

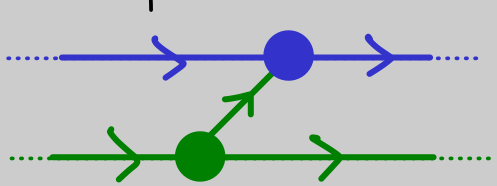
## Definition

Phoenix graph = DAG with:

- a linear main branch
- zero or more feature branches (DAGs starting on a main vertex, and ending on one or more feature vertices further to the right)

- indegree(●) $\in \{1,2\}$ ($=0$ for leftmost)
- outdegree(●) $\geq 1$ ($=0$ for rightmost)
- indegree(●) $= 1$
- outdegree(●) $\in \{1,2\}$, $=2$ iff one child of each colour

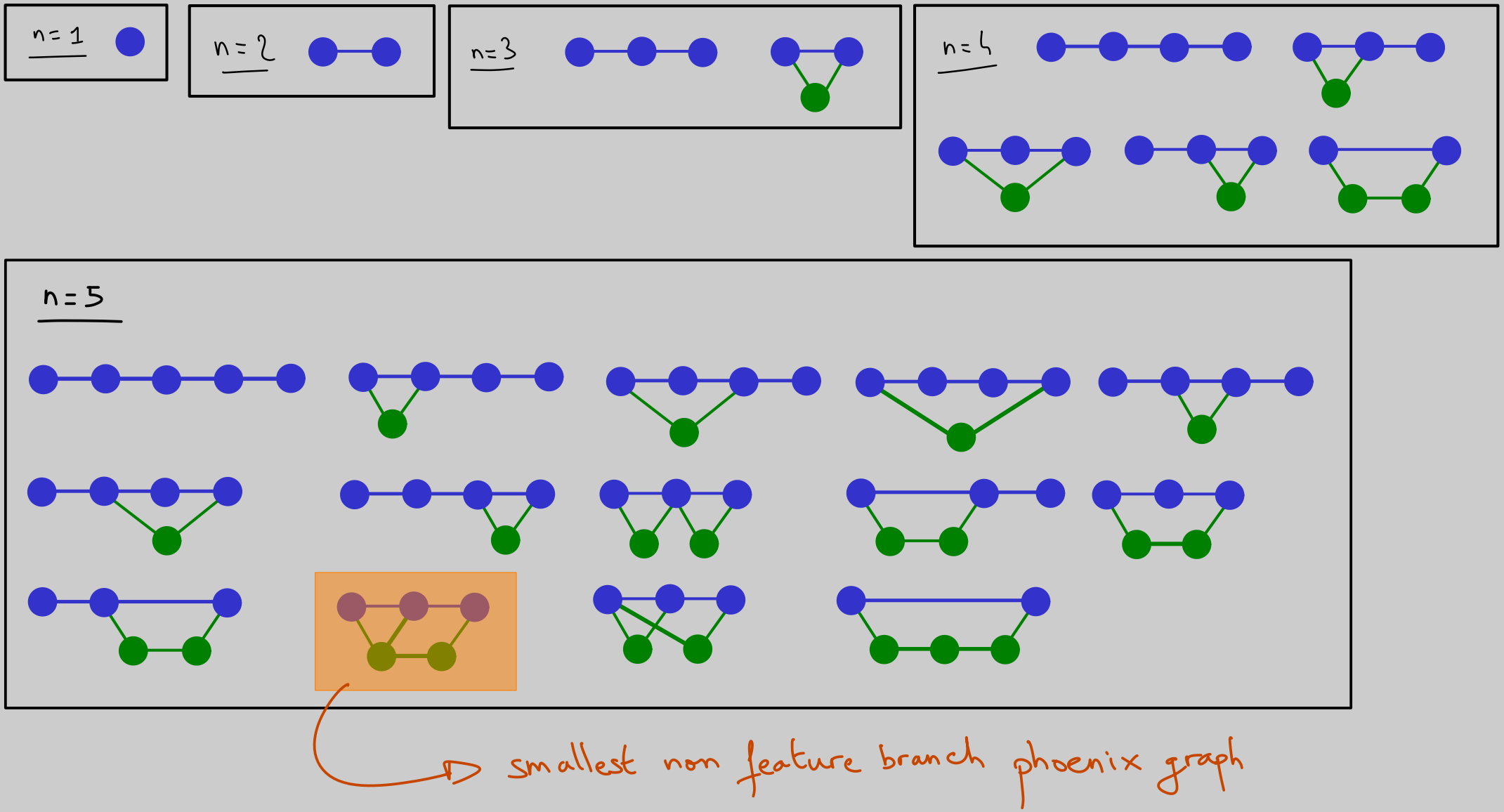
i.e a feature branch graph, but the pattern



is allowed.

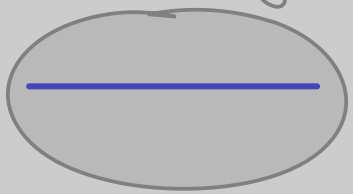
# EXAMPLES

Size $n$ of a phoenix graph = number of vertices

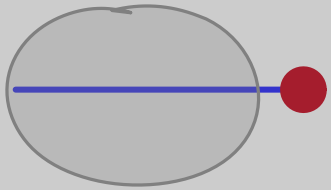


# RECURSIVE DECOMPOSITION

Phoenix graph

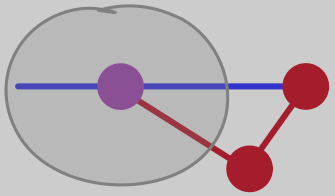


= [empty] or

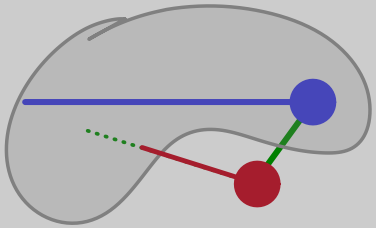


or

k-1 choices

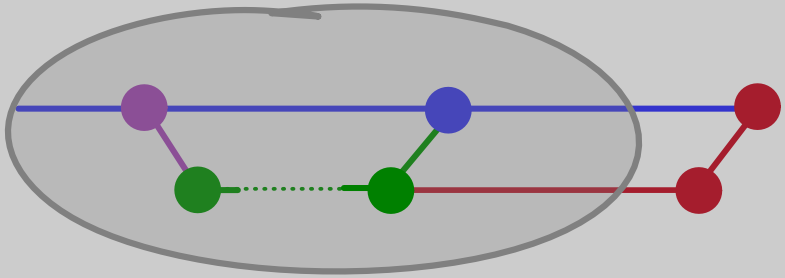


or



or

b choices



$$g_{n,k,b} = g_{n-1,k-1,b} + (k-1)\,g_{n-2,k-1,b-1} + (g_{n-1,k,b} - g_{n-2,k-1,b}) + b \cdot g_{n-2,k-1,b}$$

number of phoenix graphs with $n$ vertices, $k$ main vertices, $b$ feature branches

# GENERATING FUNCTION

★ $g_{n,k,b} = g_{n-1,k-1,b} + (k-1)\,g_{n-2,k-1,b-1} + g_{n-1,k,b} + (b-1)\,g_{n-2,k-1,b}$

★ $\widetilde{G}(z,u,v) := \sum_{n,k,b \geq 0} g_{n,k,b}\, z^n u^k v^b$

["$\widetilde{G}$" for short]

↳ not analytic...

# GENERATING FUNCTION

* $g_{n,k,b} = g_{n-1,k-1,b} + (k-1)\,g_{n-2,k-1,b-1} + g_{n-1,k,b} + (b-1)\,g_{n-2,k-1,b}$

* $\widetilde{G}(z,u,v) := \sum_{n,k,b \geq 0} g_{n,k,b}\, z^n \frac{u^k}{k!} \frac{v^b}{b!}$ ["$\widetilde{G}$" for short]

$$\hookrightarrow z^2\widetilde{G} - z^2 u \partial_u \widetilde{G} - z\partial_v \widetilde{G} + (1-z)\partial_u \partial_v \widetilde{G} - z^2 v \partial_{v,v}^2 \widetilde{G} = 0$$

Sadly, no solution ☹

Moreover, the sequence $\left(\sum_{k,b} g_{n,k,b}\right)_{n \geq 0}$ does not appear in the OEIS.

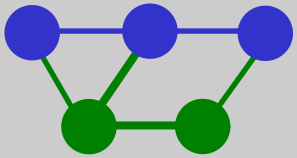
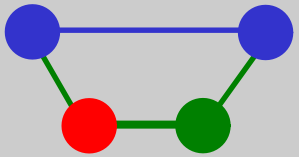
⟶ What else *can* we say about phoenix graphs?

# BABY PHOENIXES

## Definition

Baby phoenix graph = phoenix graph where every feature vertex is part of a merge (i.e one of its children is a main vertex, and no main vertex has its indegree equal to 1

## Example

 is a baby phoenix, but  is not.

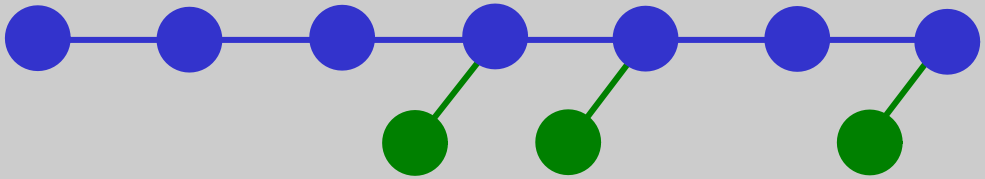
# HOW ARE BABIES MADE?

To construct a baby phoenix graph with $k \geq 1$ main vertices, start with the main branch:

# HOW ARE BABIES MADE?

To construct a baby phoenix graph with $k \geq 1$ main vertices, start with the main branch:

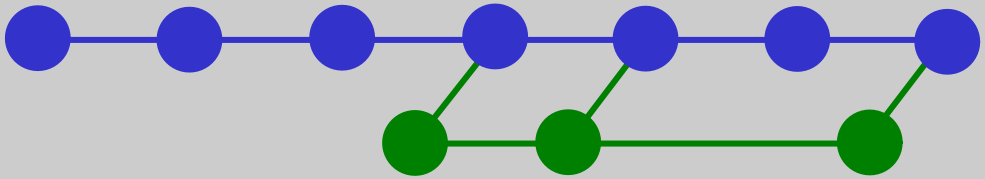
Then, among the $k-1$ rightmost vertices, take a nonempty subset, make each one of them part of a merge (adding feature vertices as needed), make them part of the same feature branch, and finally choose where that branch starts:



# HOW ARE BABIES MADE?

To construct a baby phoenix graph with $k \geq 1$ main vertices, start with the main branch:

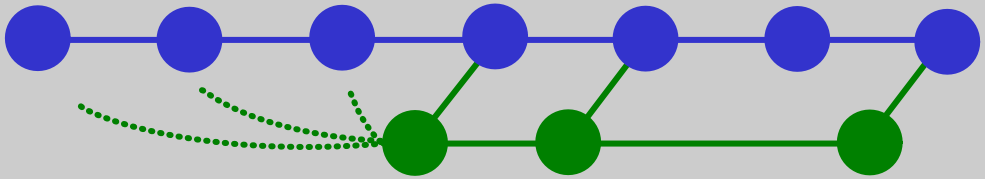
Then, among the $k-1$ rightmost vertices, take a nonempty subset, make each one of them part of a merge (adding feature vertices as needed), make them part of the same feature branch, and finally choose where that branch starts:



# HOW ARE BABIES MADE?

To construct a baby phoenix graph with $k \geq 1$ main vertices, start with the main branch:

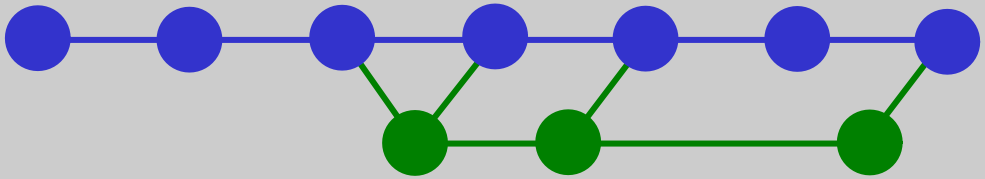
Then, among the $k-1$ rightmost vertices, take a nonempty subset, make each one of them part of a merge (adding feature vertices as needed), make them part of the same feature branch, and finally choose where that branch starts:



# HOW ARE BABIES MADE?

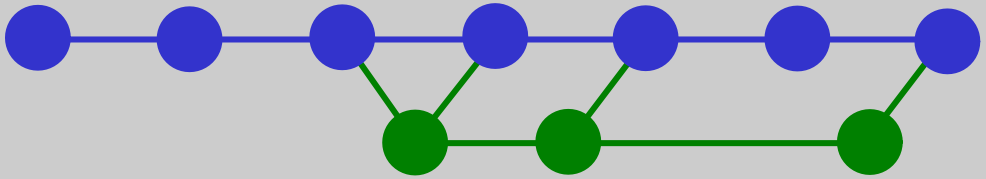
To construct a baby phoenix graph with $k \geq 1$ main vertices, start with the main branch:

Then, among the $k-1$ rightmost vertices, take a nonempty subset, make each one of them part of a merge (adding feature vertices as needed), make them part of the same feature branch, and finally choose where that branch starts:



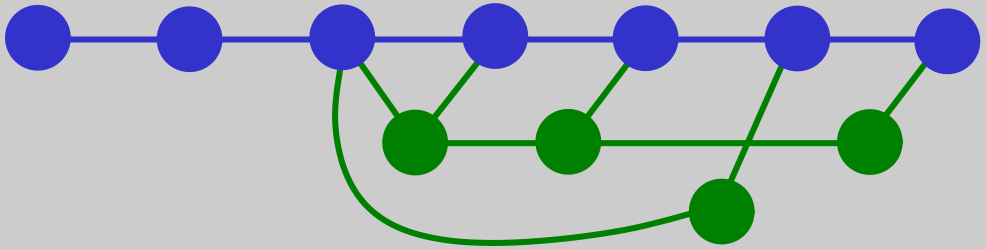
# HOW ARE BABIES MADE?

Repeat this process with the available main vertices to add other feature branches:



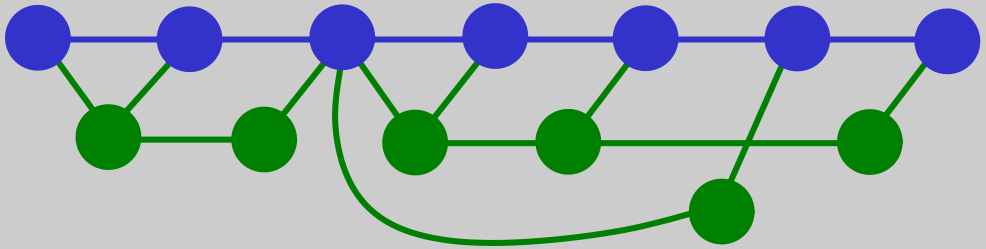
# HOW ARE BABIES MADE?

Repeat this process with the available main vertices to add other feature branches:



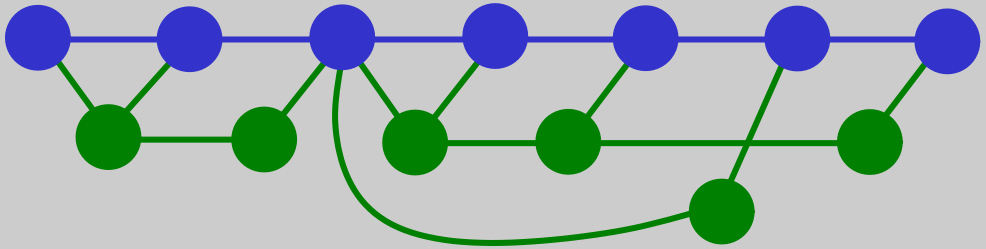
# HOW ARE BABIES MADE?

Repeat this process with the available main vertices to add other feature branches:



# HOW ARE BABIES MADE?

Repeat this process with the available main vertices to add other feature branches:



Notes

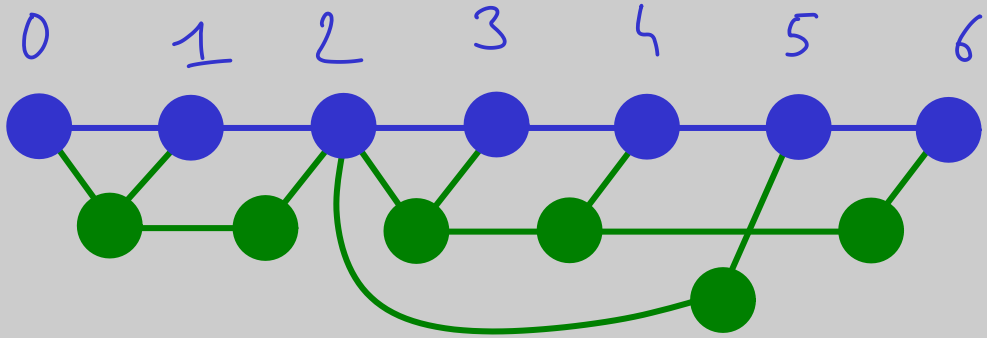
* A baby phoenix graph with $k \geq 1$ main vertices necessarily has $k-1$ feature vertices.
* Counting sequence: 1, 1, 3, 14, 89, 716, ...
  ↪ OEIS A007549

# BIJECTION WITH SET PARTITIONS

Labelling the main vertices from 0 to $k-1$, we can map a baby phoenix graph with $k$ main vertices to a set partition of size $k-1$:

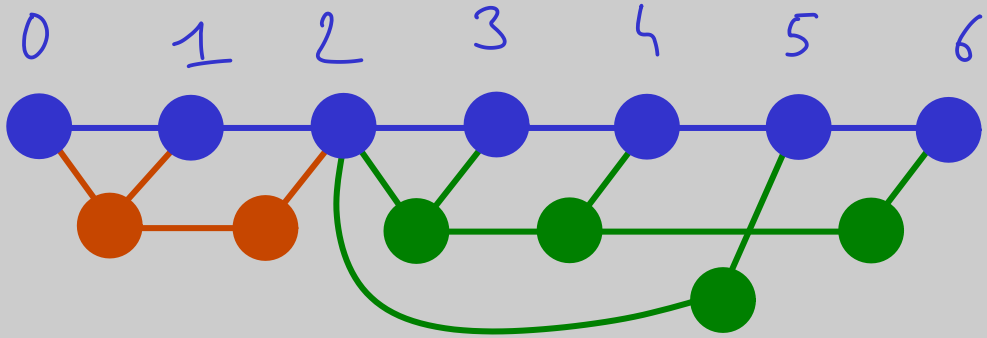
# BIJECTION WITH SET PARTITIONS

Labelling the main vertices from 0 to $k-1$, we can map a baby phoenix graph with $k$ main vertices to a set partition of size $k-1$:



# BIJECTION WITH SET PARTITIONS

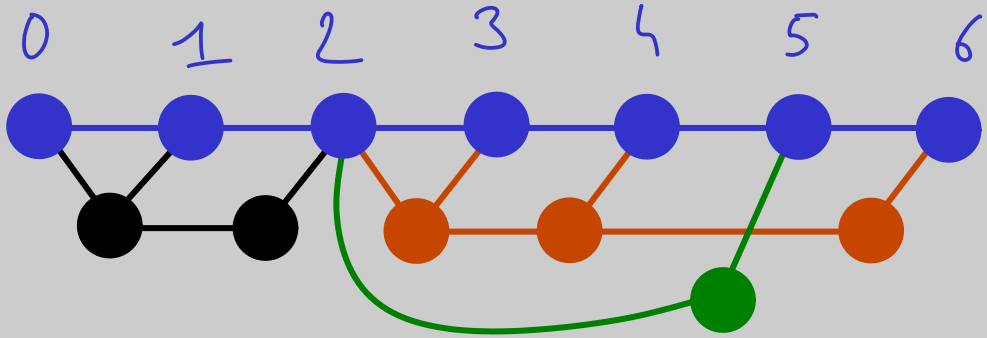
Labelling the main vertices from 0 to $k-1$, we can map a baby phoenix graph with $k$ main vertices to a set partition of size $k-1$:




$\{1, 2\}$

# BIJECTION WITH SET PARTITIONS

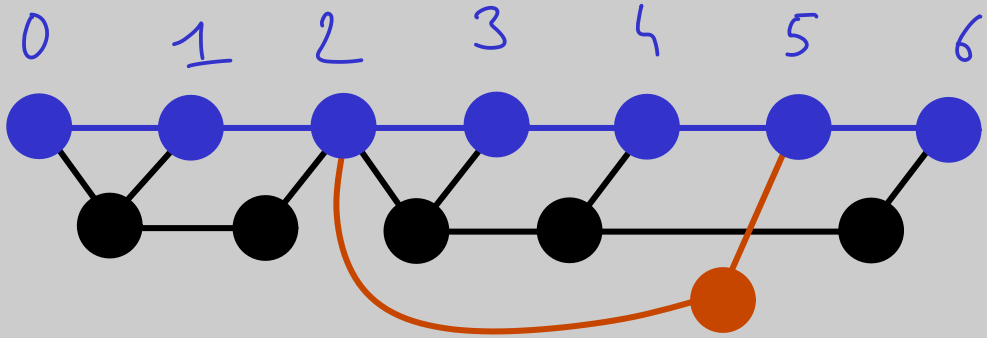
Labelling the main vertices from 0 to $k-1$, we can map a baby phoenix graph with $k$ main vertices to a set partition of size $k-1$:




$\{1, 2\}$ $\{3, 4, 6\}$

# BIJECTION WITH SET PARTITIONS

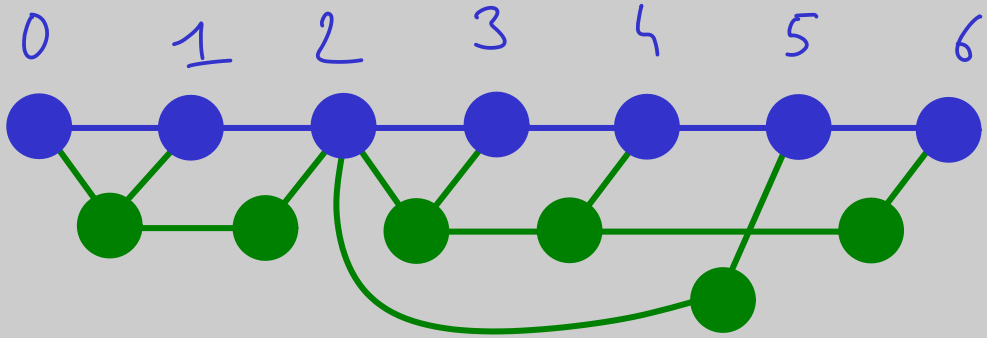
Labelling the main vertices from 0 to $k-1$, we can map a baby phoenix graph with $k$ main vertices to a set partition of size $k-1$:




$\{1,2\}$ $\{3,4,6\}$ $\{5\}$

# BIJECTION WITH SET PARTITIONS

Labelling the main vertices from 0 to $k-1$, we can map a baby phoenix graph with $k$ main vertices to a set partition of size $k-1$:




$$[6] = \{1,2\} \sqcup \{3,4,6\} \sqcup \{5\}$$

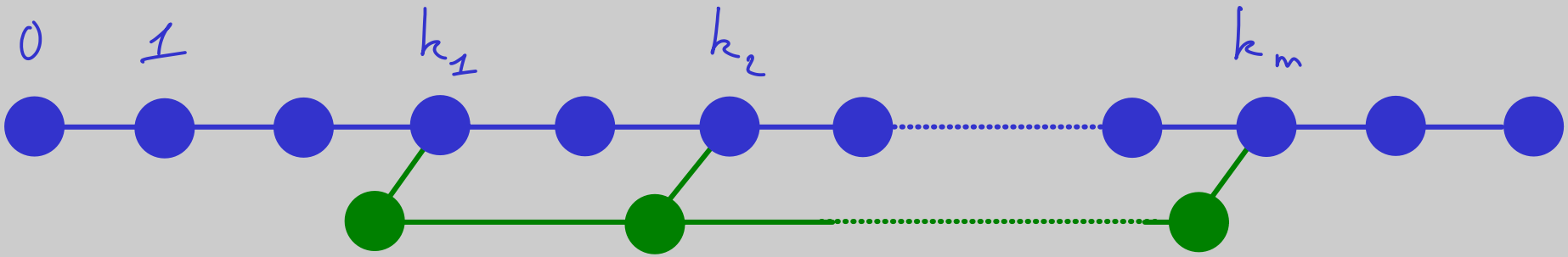
↳ This is a surjection. How do we make it a bijection?

# BIJECTION WITH SET PARTITIONS

How many ways are there to obtain a given part $\{k_1, k_2, \ldots, k_m\}$ (here, $k_1 < k_2 < \ldots < k_m$)?

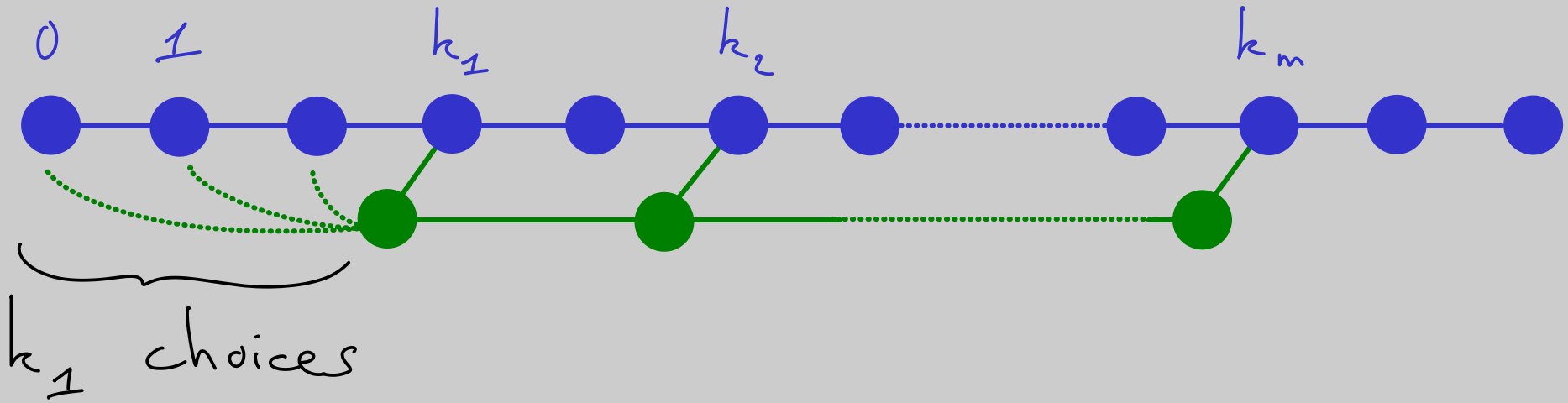
# BIJECTION WITH SET PARTITIONS

How many ways are there to obtain a given part $\{k_1, k_2, \ldots, k_m\}$ (here, $k_1 < k_2 < \ldots < k_m$)?



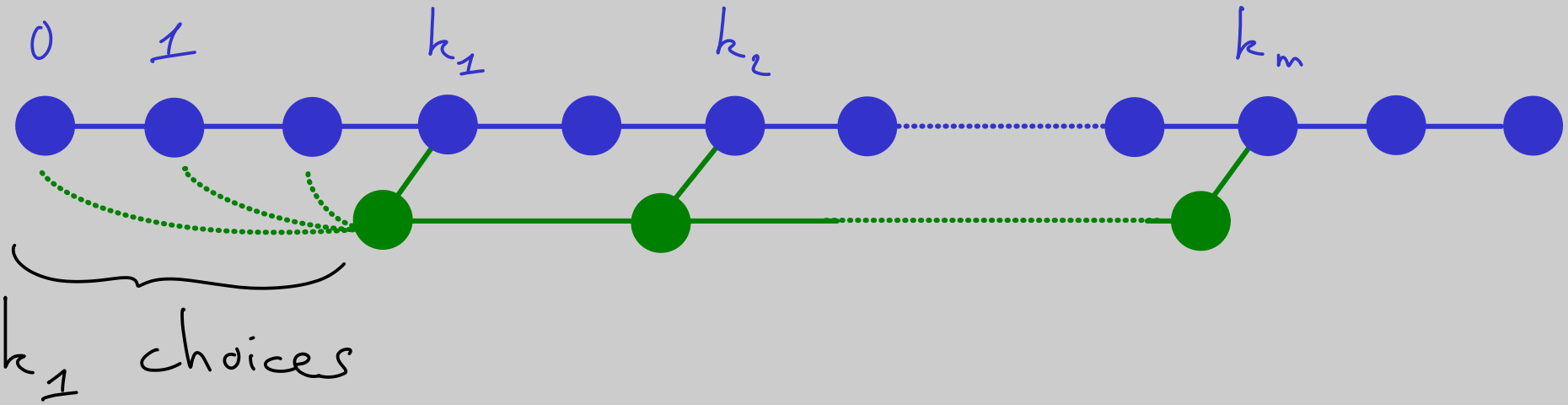
# BIJECTION WITH SET PARTITIONS

How many ways are there to obtain a given part $\{k_1, k_2, \dots, k_m\}$ (here, $k_1 < k_2 < \dots < k_m$)?



# BIJECTION WITH SET PARTITIONS

How many ways are there to obtain a given part $\{k_1, k_2, \dots, k_m\}$ (here, $k_1 < k_2 < \dots < k_m$)?




↳ Baby phoenix graphs with $k$ main vertices

$\simeq$

Set partitions of size $k-1$,

weighted by the product of part minima

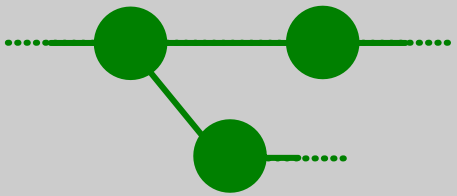
# FORK ANYWHERE

["FA" for short]

**Definition**

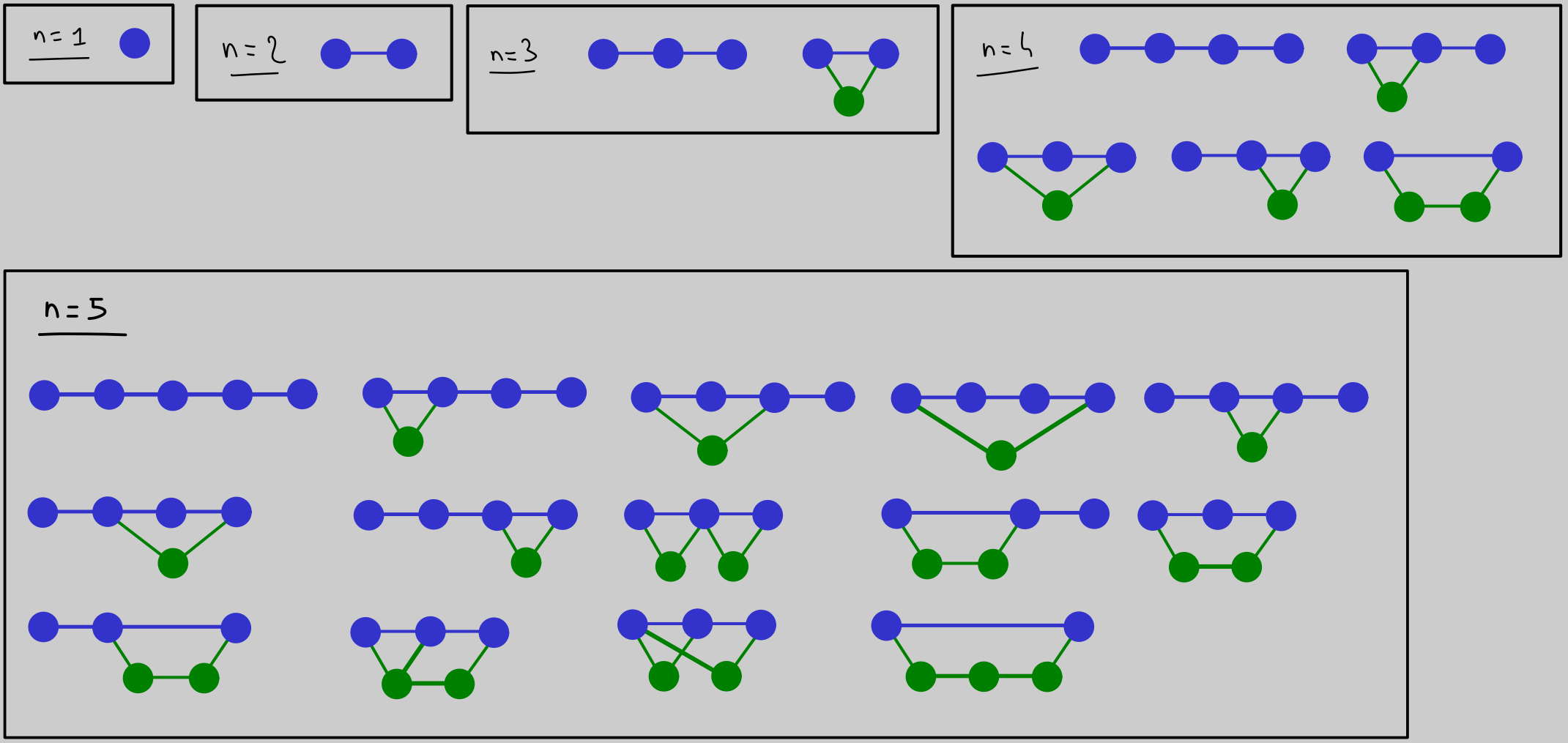
Fork anywhere graph = phoenix graph, but vertices on feature branches can have their outdegree equal to 2 without restriction on their children's types,

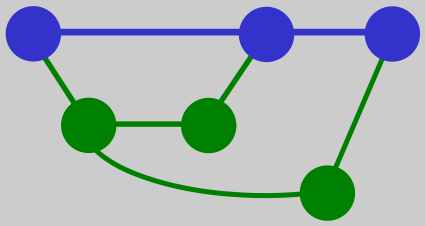
i.e. a phoenix graph, but the pattern



is allowed.

# EXAMPLES

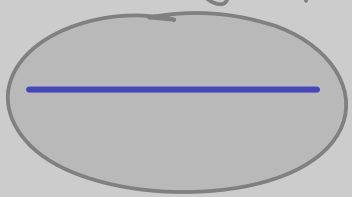


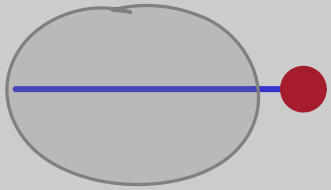
is the smallest non phoenix FA graph.

# RECURSIVE DECOMPOSITION

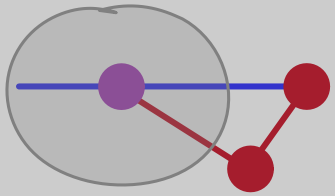
FA graph



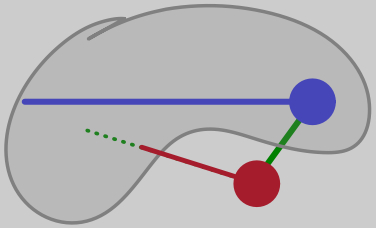
= [empty] or



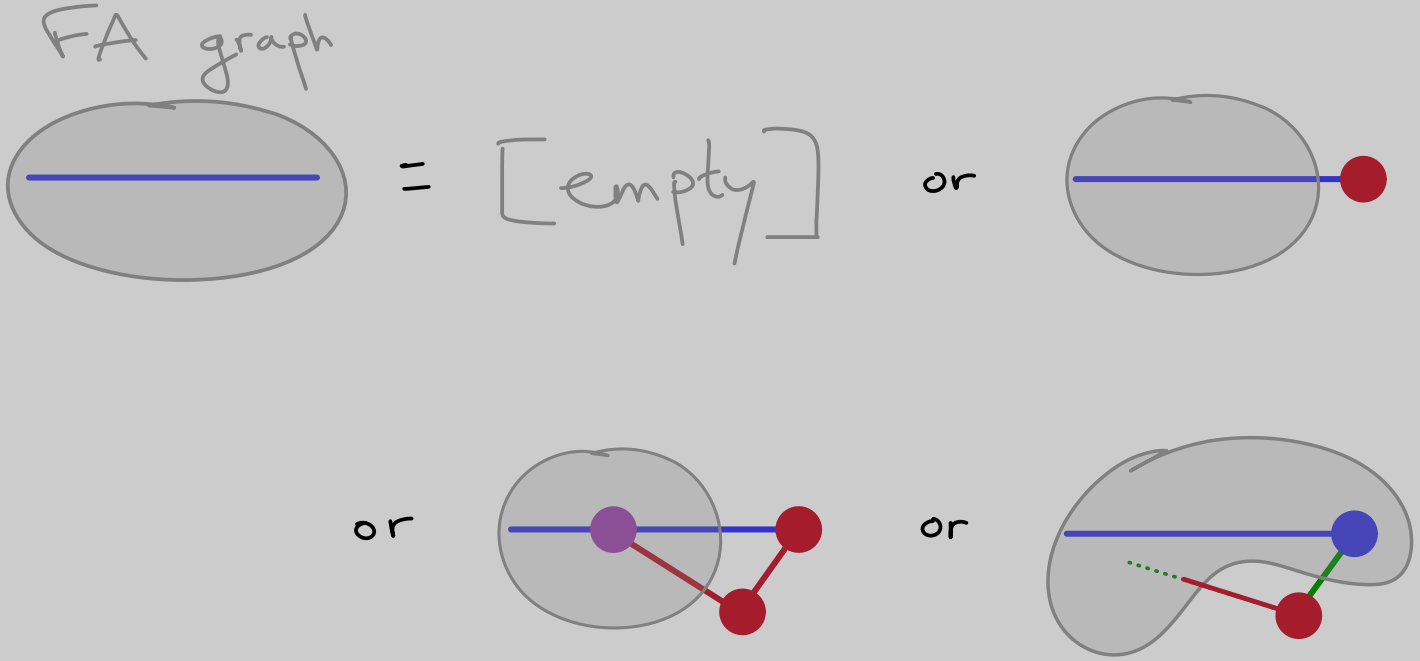
or



or



# RECURSIVE DECOMPOSITION



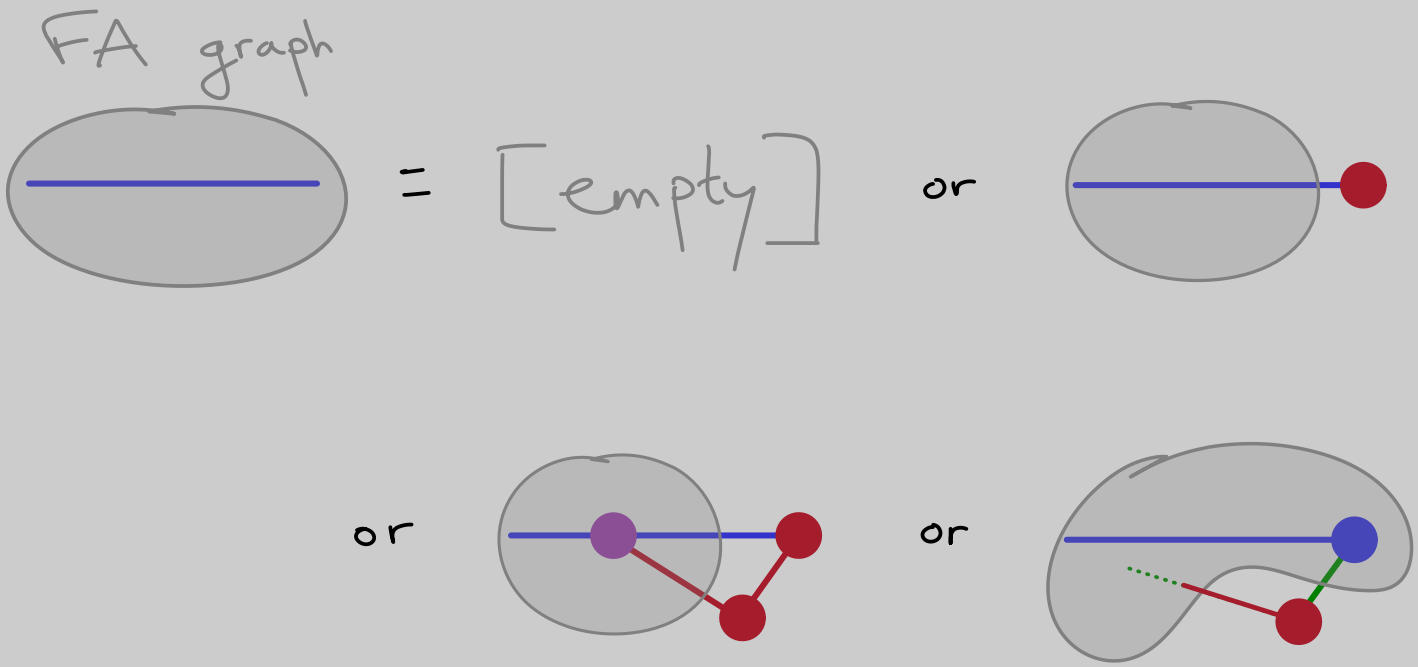

OGF "$G$":

$$G(z) = \sum_{n \geq 0} g_n z^n$$

EGF "$\tilde{G}$":

$$\tilde{G}(z) = \sum_{n \geq 0} \frac{g_n}{n!} z^n$$

# RECURSIVE DECOMPOSITION




OGF "$G$":

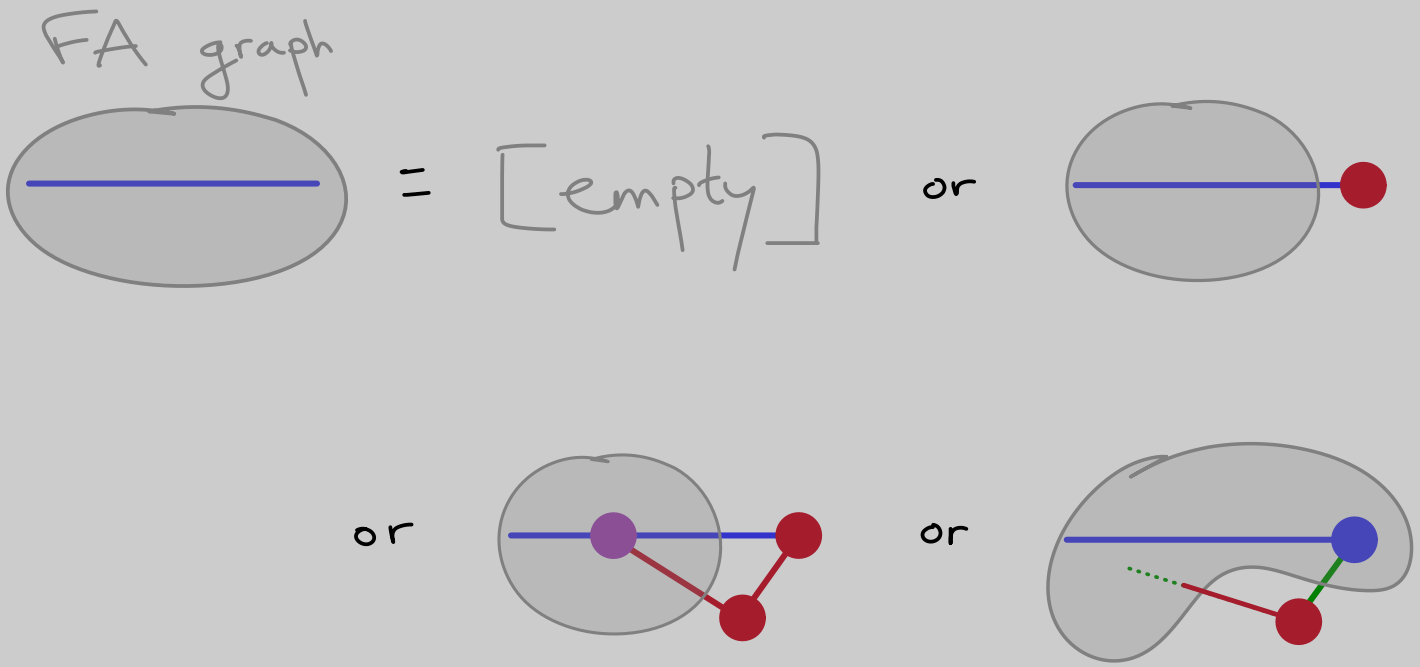
$$G(z) = \sum_{n \geq 0} g_n z^n$$

EGF "$\tilde{G}$":

$$\tilde{G}(z) = \sum_{n \geq 0} \frac{g_n}{n!} z^n$$

$$\hookrightarrow G(z) = 1 + zG(z) + z^2 \cdot zG'(z) + z \cdot \left( G(z) - zG(z) - 1 \right)$$

# RECURSIVE DECOMPOSITION




OGF "$G$":

$$G(z) = \sum_{n \geq 0} g_n z^n$$

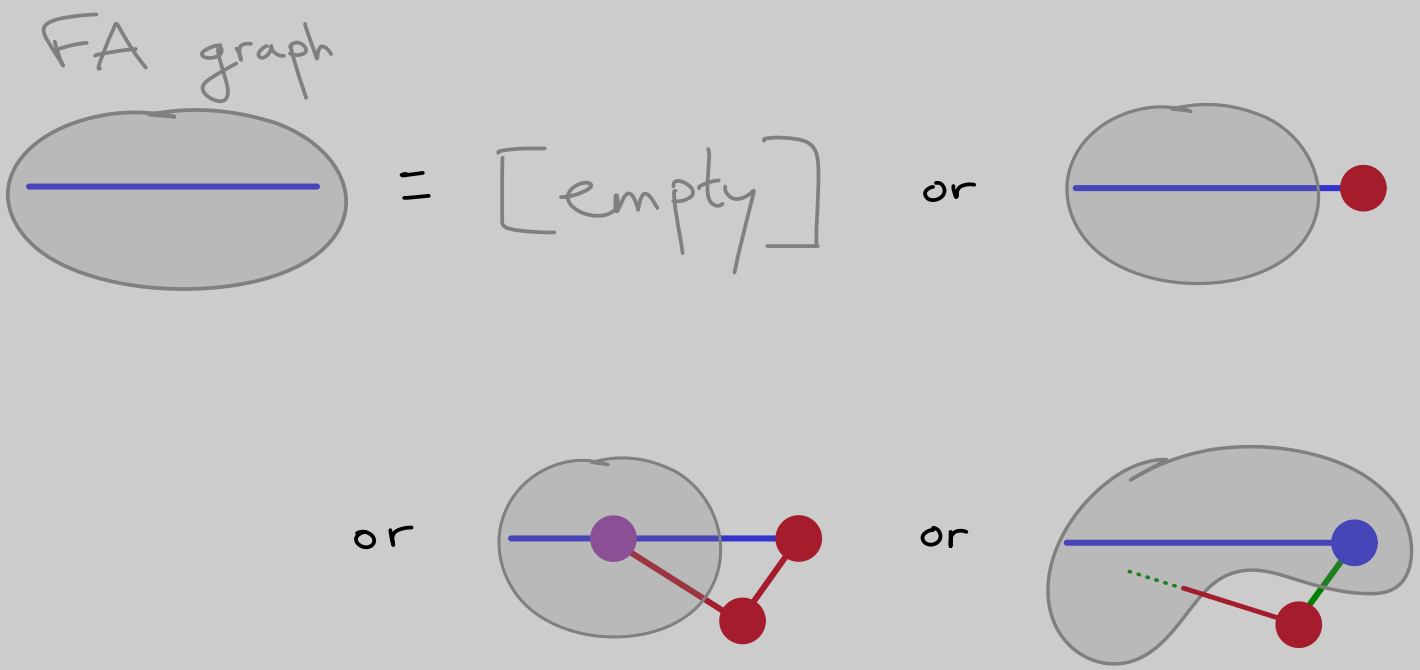
EGF "$\tilde{G}$":

$$\tilde{G}(z) = \sum_{n \geq 0} \frac{g_n}{n!} z^n$$

$$\hookrightarrow G(z) = 1 + zG(z) + z^2 \cdot zG'(z) + z \cdot (G(z) - zG(z) - 1)$$

$$\rightarrow \frac{z^2}{2} \tilde{G}''(z) = \frac{z^2}{2} e^z e^{z + z^2/2}$$

# RECURSIVE DECOMPOSITION




OGF "$G$":

$$G(z) = \sum_{n \geq 0} g_n z^n$$

EGF "$\tilde{G}$":

$$\tilde{G}(z) = \sum_{n \geq 0} \frac{g_n}{n!} z^n$$

$$\hookrightarrow G(z) = 1 + zG(z) + z^2 \cdot zG'(z) + z \cdot (G(z) - zG(z) - 1)$$

$$\rightarrow \frac{z^2}{2}\tilde{G}''(z) = \frac{z^2}{2} e^z e^{z + z^2/2}$$

$$\hookrightarrow \hat{G} := \tilde{G}'' \text{ satisfies } \hat{G}(z) = \exp\left(2z + \frac{1}{2}z^2\right)$$

GF

$$\hat{G}(z) = \exp\left(2z + \frac{1}{2}z^2\right)$$

# GF

$$\hat{G}(z) = \exp\left(2z + \frac{1}{2}z^2\right)$$

↳ symbolic: $\mathcal{G} = \mathrm{SET}\left(\mathcal{Z}_\bullet \sqcup \mathcal{Z}_\circ \sqcup \mathrm{CYC}_2(\mathcal{Z}_\circ)\right)$

# GF

$$\hat{G}(z) = \exp\left(2z + \frac{1}{2}z^2\right)$$

↳ symbolic: $\mathcal{G} = \mathrm{SET}\left(\mathcal{Z}_\bullet \sqcup \mathcal{Z}_\circ \sqcup \mathrm{CYC}_2(\mathcal{Z}_\circ)\right)$

↳ can be interpreted as involutions with 2 types of fixed points

# GF

$$\hat{G}(z) = \exp\left(2z + \frac{1}{2}z^2\right)$$

↳ symbolic: $\mathcal{G} = \mathrm{SET}\left(\mathcal{Z}_\bullet \sqcup \mathcal{Z}_\circ \sqcup \mathrm{CYC}_2(\mathcal{Z}_\circ)\right)$

↳ can be interpreted as involutions with 2 types of fixed points

↳ Bijection ?

# GF

$$\hat{G}(z) = \exp\left(2z + \frac{1}{2}z^2\right)$$

↳ symbolic : $\mathcal{G} = \mathrm{SET}\left(\mathcal{Z}_\bullet \sqcup \mathcal{Z}_\circ \sqcup \mathrm{CYC}_2(\mathcal{Z}_\circ)\right)$

↳ can be interpreted as involutions with 2 types of fixed points

↳ Bijection ?

↳ Random generation of FA ?

# GF

$$\hat{G}(z) = \exp\left(2z + \frac{1}{2}z^2\right)$$

↳ symbolic : $\mathcal{G} = \mathrm{SET}\left(\mathcal{Z}_\bullet \sqcup \mathcal{Z}_\circ \sqcup \mathrm{CYC}_2(\mathcal{Z}_\circ)\right)$

↳ can be interpreted as involutions with 2 types of fixed points

↳ Bijection ?

↳ Random generation of FA ?

↳ asymptotics : GF has no singularity

# GF

$$\hat{G}(z) = \exp\left(2z + \frac{1}{2}z^2\right)$$

↳ symbolic : $\mathcal{G} = \mathrm{SET}\left(\mathcal{Z}_\bullet \sqcup \mathcal{Z}_\circ \sqcup \mathrm{CYC}_2(\mathcal{Z}_\circ)\right)$

↳ can be interpreted as involutions with 2 types of fixed points

↳ Bijection ?

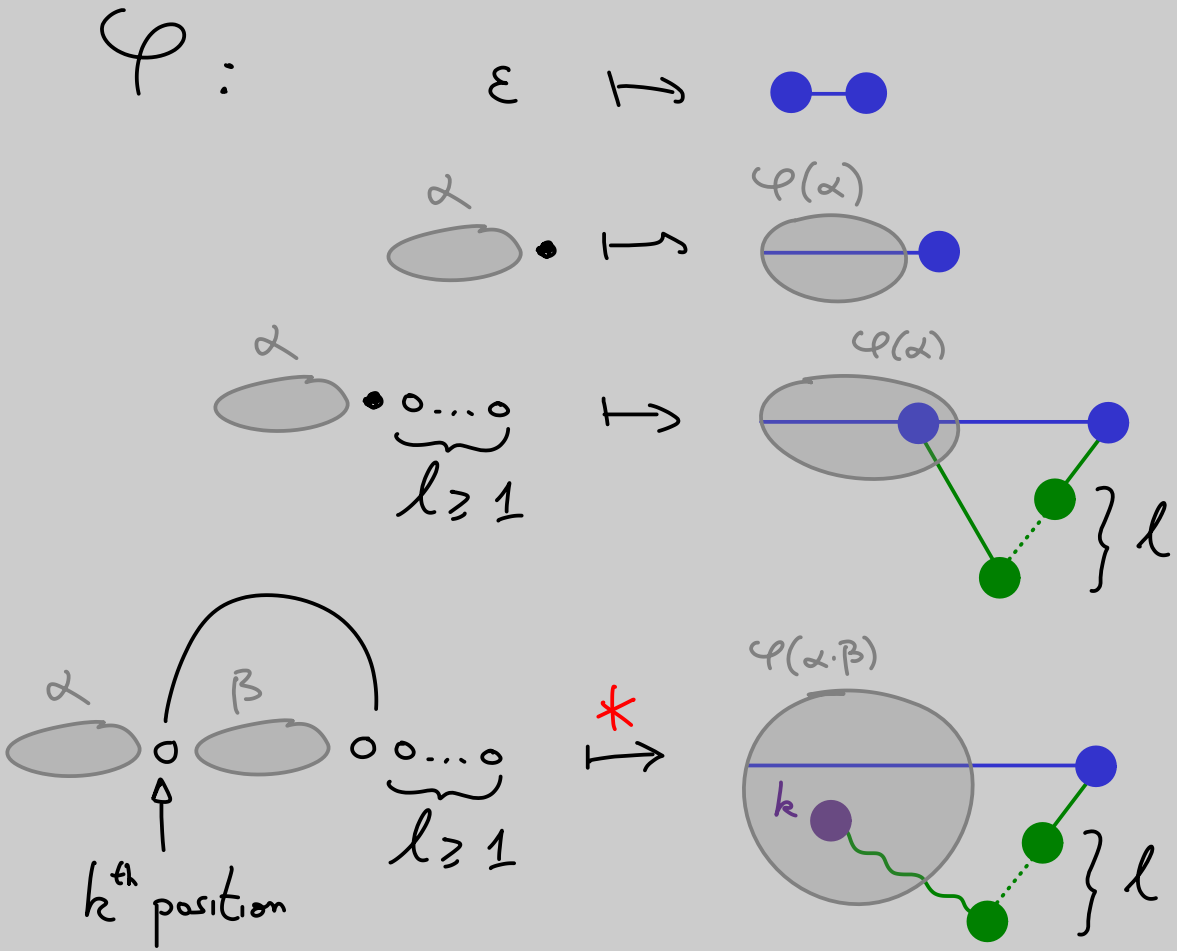
↳ Random generation of FA ?

↳ asymptotics : GF has no singularity

↳ Saddle point method ?

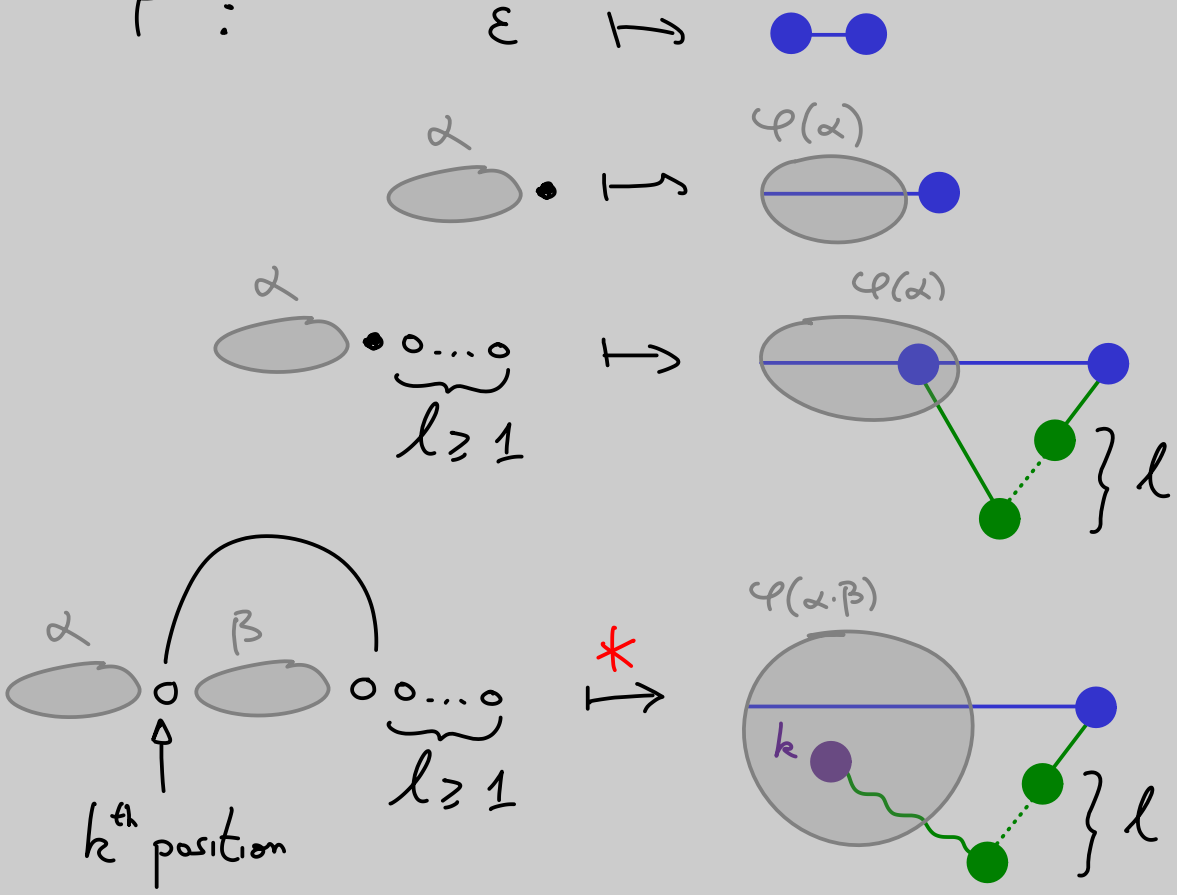
# BIJECTION(S)

## Example

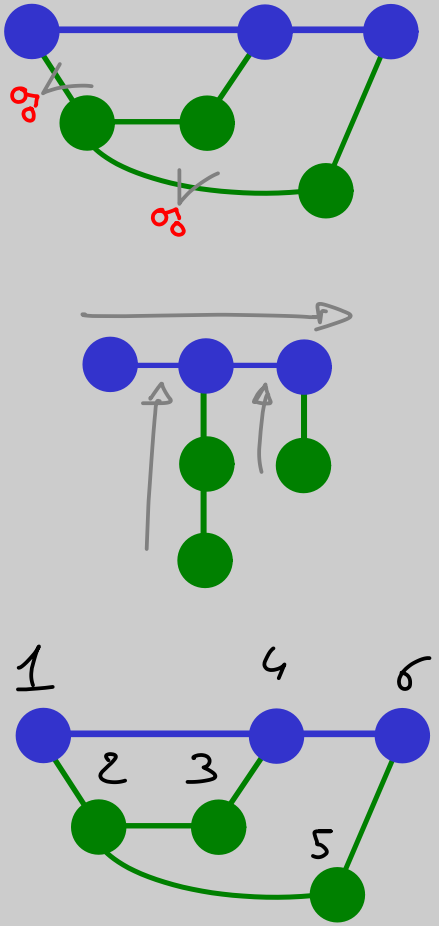


# BIJECTION(S)

## Example




* FA graph vertex labelling :



# BIJECTION(S)

## Open problem

feature branch $\subset$ phoenix $\subset$ fork anywhere

a certain bijection

involutions with 2 types of fixed points

# BIJECTION(S)

## Open problem

feature branch $\subset$ phoenix $\subset$ fork anywhere

↓ same bijection

↓ same bijection

↓ a certain bijection

? $\subset$ ? $\subset$ involutions with 2 types of fixed points

# BIJECTION(S)

## Open problem

feature branch $\subset$ phoenix $\subset$ fork anywhere

↓ same bijection

↓ same bijection

↓ a certain bijection

? $\subset$ ? $\subset$ involutions with 2 types of fixed points

Ideally, we would want the ?'s to be subclasses defined by pattern avoidance.

# RANDOM GENERATION

Boltzmann sampler for involutions:

$$\forall x > 0, \quad \mathbb{P}_x\left(\begin{array}{c} i \\ \uparrow \\ \text{involution} \end{array}\right) = \frac{x^{|i|}}{|i|! \cdot \hat{G}(x)}$$

# RANDOM GENERATION

Boltzmann sampler for involutions:

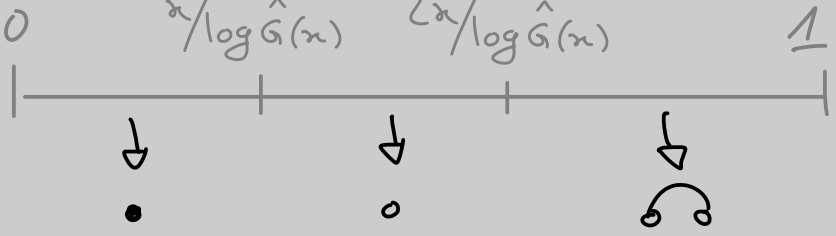
$$\forall x > 0, \quad \mathbb{P}_x\left(i\right) = \frac{x^{|i|}}{|i|! \cdot \hat{G}(x)}$$

($i$: involution)

$$\hat{G}(x) = \exp\left(2x + \frac{1}{2}x^2\right)$$

↳ ⋆ size $\rightsquigarrow$ Poisson$\left(2x + \frac{1}{2}x^2\right)$

⋆ draw each of the nodes according to $u \rightsquigarrow \mathcal{U}_{[0,1]}$



# RANDOM GENERATION

Boltzmann → involution with 2 types of fixed points

↓

FA graph ← any working bijection

```
%%time

boltz_involution(10**6)
```

```
CPU times: user 1.58 s, sys: 37.6 ms, total: 1.62 s
Wall time: 1.62 s
```

# ASYMPTOTICS

Cauchy formula: $[z^n]\,\hat{G}(z) \underset{n\to\infty}{\sim} \frac{1}{2i\pi} \oint \frac{\hat{G}(z)}{z^{n+1}}\,dz$

# ASYMPTOTICS

Cauchy formula: $[z^n]\,\hat{G}(z) \underset{n\to\infty}{\sim} \frac{1}{2i\pi} \oint \frac{\hat{G}(z)}{z^{n+1}}\,dz$

approximation given by saddle point method

# ASYMPTOTICS

Cauchy formula: $[z^n]\,\hat{G}(z) \underset{n\to\infty}{\sim} \frac{1}{2i\pi} \oint \frac{\hat{G}(z)}{z^{n+1}}\,dz$

approximation given by saddle point method

↳ approximation:

$$[z^n]\,\hat{G}(z) \underset{n\to\infty}{\sim} \frac{1}{2\sqrt{\pi n}}\, e^{-3/2}\, n^{-n/2}\, e^{n/2+2\sqrt{n}}$$

# ASYMPTOTICS

Cauchy formula: $[z^n]\,\hat{G}(z) \underset{n\to\infty}{\sim} \frac{1}{2i\pi} \oint \frac{\hat{G}(z)}{z^{n+1}}\,dz$

approximation given by saddle point method

↳ approximation:

$$[z^n]\,\hat{G}(z) \underset{n\to\infty}{\sim} \frac{1}{2\sqrt{\pi n}}\, e^{-3/2}\, n^{-n/2}\, e^{n/2+2\sqrt{n}}$$

differences with classic involutions

# OPEN QUESTIONS

* Study other classes of git graphs (realism, ease of computation, ...)

* Evaluate how "close" a typical DAG is to all of these git graph classes

# OPEN QUESTIONS

- ✡ Study other classes of git graphs (realism, ease of computation, ...)
- ✡ Evaluate how "close" a typical DAG is to all of these git graph classes

Thank you :)