Enumeration of unlabeled graph classes

Jordi Castellví (CRM)

Work in collaboration with Michael Drmota and Clément Requilé

BMS-BGSMath Junior Meeting 2026 - Cerdanyola del Vallès

Labeled vs unlabeled

A graph with n vertices is **labeled** if each vertex carries a different label in $\{1, 2, \ldots, n\}$.

Labeled vs unlabeled

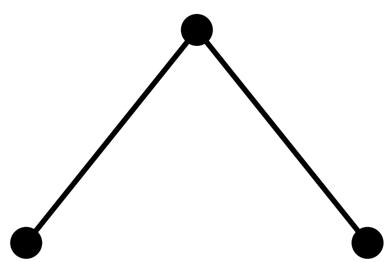
A graph with n vertices is **labeled** if each vertex carries a different label in $\{1, 2, \ldots, n\}$.

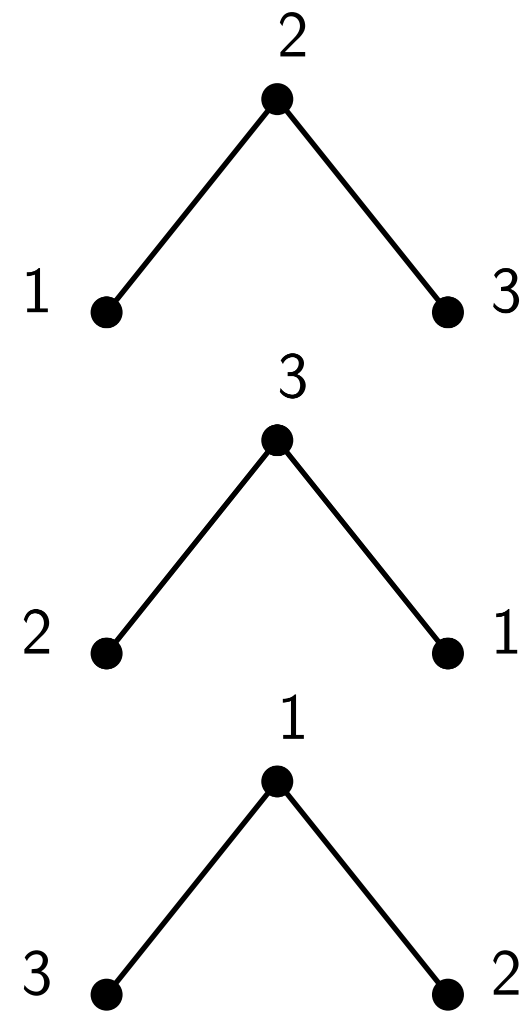
In an **unlabeled** graph, the vertices cannot be distinguished by any labels.

Labeled vs unlabeled

A graph with n vertices is **labeled** if each vertex carries a different label in $\{1, 2, \ldots, n\}$.

In an **unlabeled** graph, the vertices cannot be distinguished by any labels.





The symbolic method

Our goal is to determine the number of graphs in the family with size n.

The symbolic method

Our goal is to determine the number of graphs in the family with size n.

Definition. A **combinatorial class** is a pair $(\mathcal{A}, |\cdot|)$ where

- $\mathcal{A}$ is a family of combinatorial objects,
- $|\cdot| : \mathcal{A} \to \mathbb{N}$ is a size function,
- The number of objects with size n is $a_n < \infty$.

The symbolic method

Our goal is to determine the number of graphs in the family with size n.

Definition. A **combinatorial class** is a pair $(\mathcal{A}, |\cdot|)$ where

- $\mathcal{A}$ is a family of combinatorial objects,
- $|\cdot| : \mathcal{A} \to \mathbb{N}$ is a size function,
- The number of objects with size n is $a_n < \infty$.

Definition. The **ordinary generating function** (OGF) of $(\mathcal{A}, |\cdot|)$ is the formal power series

$$A(x) = \sum_{n \geq 0} a_n x^n.$$

Suitable for unlabeled classes.

Definition. The **exponential generating function** (EGF) of $(\mathcal{A}, |\cdot|)$ is the formal power series

$$A(x) = \sum_{n \geq 0} \frac{a_n}{n!} x^n.$$

Suitable for labeled classes.

The symbolic method

Operations between **classes** translate into relations involving their **generating functions**.

Combinatorial class	**Generating function**
$\mathcal{A} \cup \mathcal{B}$	$A(x) + B(x)$
$\mathcal{A} \times \mathcal{B}$	$A(x) \cdot B(x)$
$\mathcal{A} \circ \mathcal{B}$	$A(B(x))$
$\mathsf{Seq}(\mathcal{A})$	$\frac{1}{1-A(x)}$
$\mathsf{Set}(\mathcal{A})$	$\exp(A(x))$

The symbolic method

Operations between **classes** translate into relations involving their **generating functions**.

Combinatorial class	**Generating function**
$\mathcal{A} \cup \mathcal{B}$	$A(x) + B(x)$
$\mathcal{A} \times \mathcal{B}$	$A(x) \cdot B(x)$
$\mathcal{A} \circ \mathcal{B}$	$A(B(x))$
$\mathsf{Seq}(\mathcal{A})$	$\frac{1}{1-A(x)}$
$\mathsf{Set}(\mathcal{A})$	$\exp(A(x))$

The goal is to obtain (a system of) equations that determine the GF of our class.

The symbolic method

Operations between **classes** translate into relations involving their **generating functions**.

Combinatorial class	**Generating function**
$\mathcal{A} \cup \mathcal{B}$	$A(x) + B(x)$
$\mathcal{A} \times \mathcal{B}$	$A(x) \cdot B(x)$
$\mathcal{A} \circ \mathcal{B}$	$A(B(x))$
$\mathsf{Seq}(\mathcal{A})$	$\frac{1}{1-A(x)}$
$\mathsf{Set}(\mathcal{A})$	$\exp(A(x))$

The goal is to obtain (a system of) equations that determine the GF of our class.

The growth of the coefficients of generating functions can be studied by considering them as complex variable functions and analysing their singularities. This is called **Analytic Combinatorics** [Flajolet & Sedgewick (2009)]

Labeled trees

Let $\mathcal{T}$ be the class of labeled trees.

Labeled trees

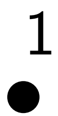
Let $\mathcal{T}$ be the class of labeled trees.

$$T(x) = \frac{1}{1!}x + \frac{1}{2!}x^2 + \frac{3}{3!}x^3 + \frac{16}{4!}x^4 \cdots .$$

Labeled trees

Let $\mathcal{T}$ be the class of labeled trees.

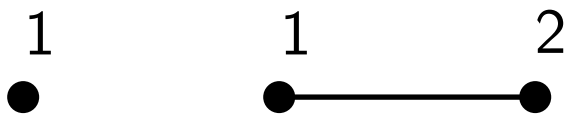
$$T(x) = \frac{1}{1!}x + \frac{1}{2!}x^2 + \frac{3}{3!}x^3 + \frac{16}{4!}x^4 \cdots .$$



Labeled trees

Let $\mathcal{T}$ be the class of labeled trees.

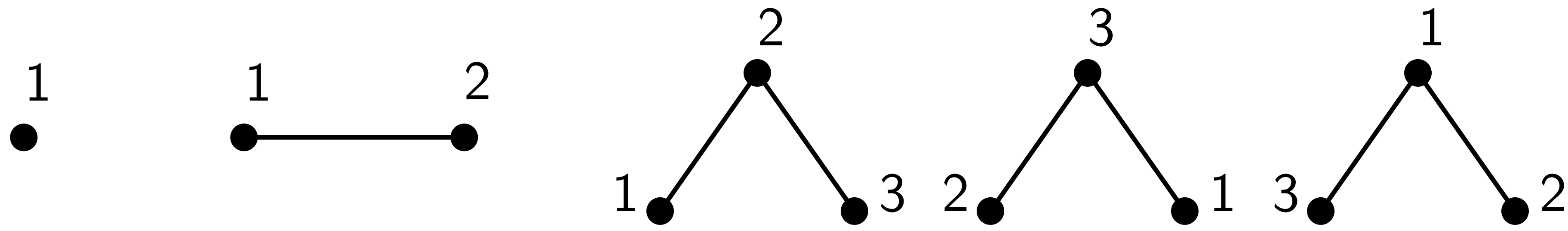
$$T(x) = \frac{1}{1!}x + \frac{1}{2!}x^2 + \frac{3}{3!}x^3 + \frac{16}{4!}x^4 \cdots .$$



Labeled trees

Let $\mathcal{T}$ be the class of labeled trees.

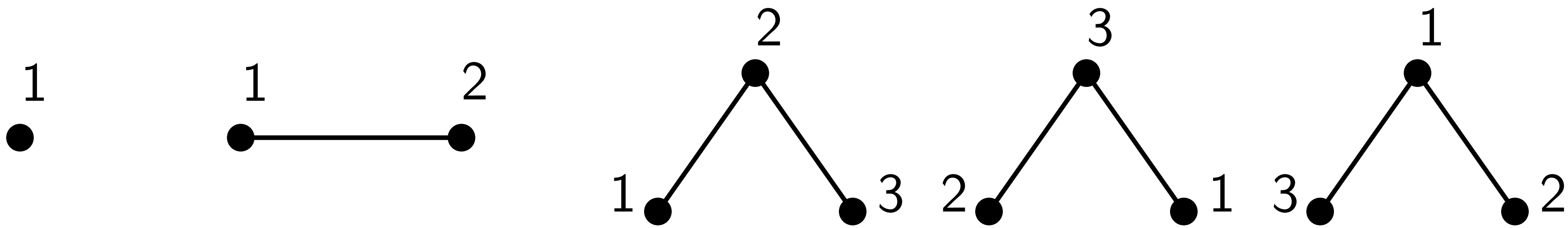
$$T(x) = \frac{1}{1!}x + \frac{1}{2!}x^2 + \frac{3}{3!}x^3 + \frac{16}{4!}x^4 \cdots .$$



Labeled trees

Let $\mathcal{T}$ be the class of labeled trees.

$$T(x) = \frac{1}{1!}x + \frac{1}{2!}x^2 + \frac{3}{3!}x^3 + \frac{16}{4!}x^4 \cdots .$$

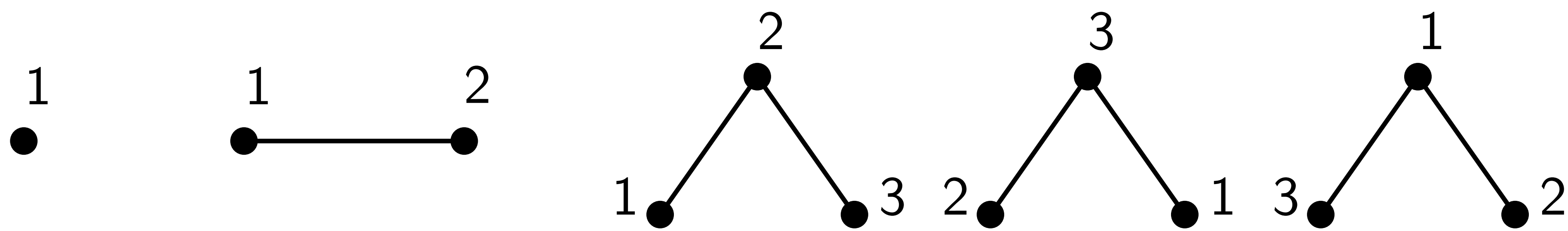


Cayley's formula: $|\mathcal{T}_n| = n^{n-2}$ [Borchardt (1860)]

Labeled trees

Let $\mathcal{T}$ be the class of labeled trees.

$$T(x) = \frac{1}{1!}x + \frac{1}{2!}x^2 + \frac{3}{3!}x^3 + \frac{16}{4!}x^4 \cdots .$$



Cayley's formula: $|\mathcal{T}_n| = n^{n-2}$ [Borchardt (1860)]

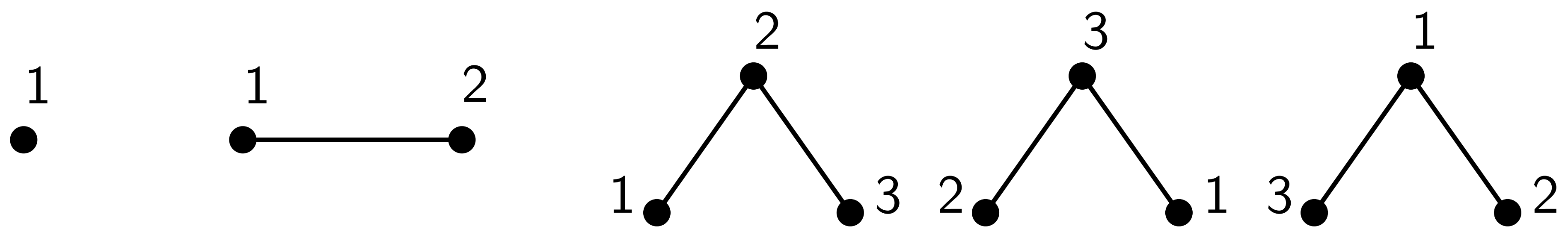
Rooting. Let $\mathcal{T}^\bullet$ be the class of rooted labeled trees.
Since all vertices are distinguishable, there are n ways to root a tree with n vertices. Thus,

$$T^\bullet(x) = \sum_{n\geq 0} n\frac{t_n}{n!}\,x^n = xT'(x).$$

Labeled trees

Let $\mathcal{T}$ be the class of labeled trees.

$$T(x) = \frac{1}{1!}x + \frac{1}{2!}x^2 + \frac{3}{3!}x^3 + \frac{16}{4!}x^4 \cdots .$$



Cayley's formula: $|\mathcal{T}_n| = n^{n-2}$ [Borchardt (1860)]

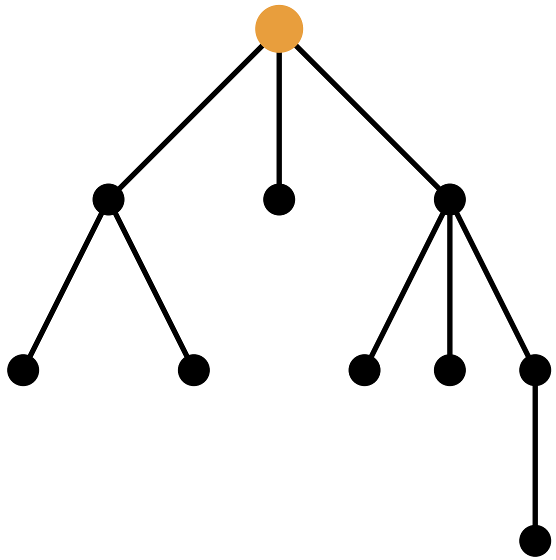
Rooting. Let $\mathcal{T}^\bullet$ be the class of rooted labeled trees.
Since all vertices are distinguishable, there are n ways to root a tree with n vertices. Thus,

$$T^\bullet(x) = \sum_{n\geq 0} n\frac{t_n}{n!}\,x^n = xT'(x).$$

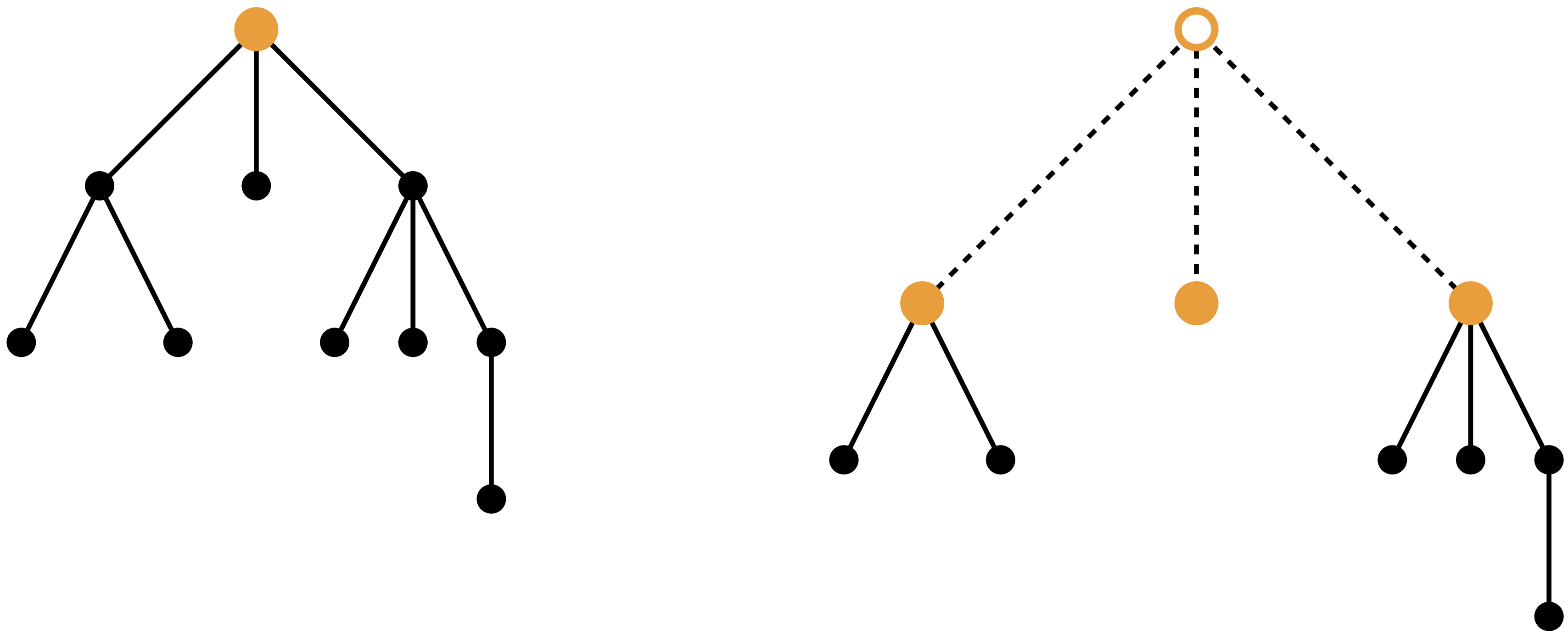
Unrooting. To do the inverse operation, we can simply integrate:

$$T(x) = \int T^\bullet(x)/x \; dx.$$

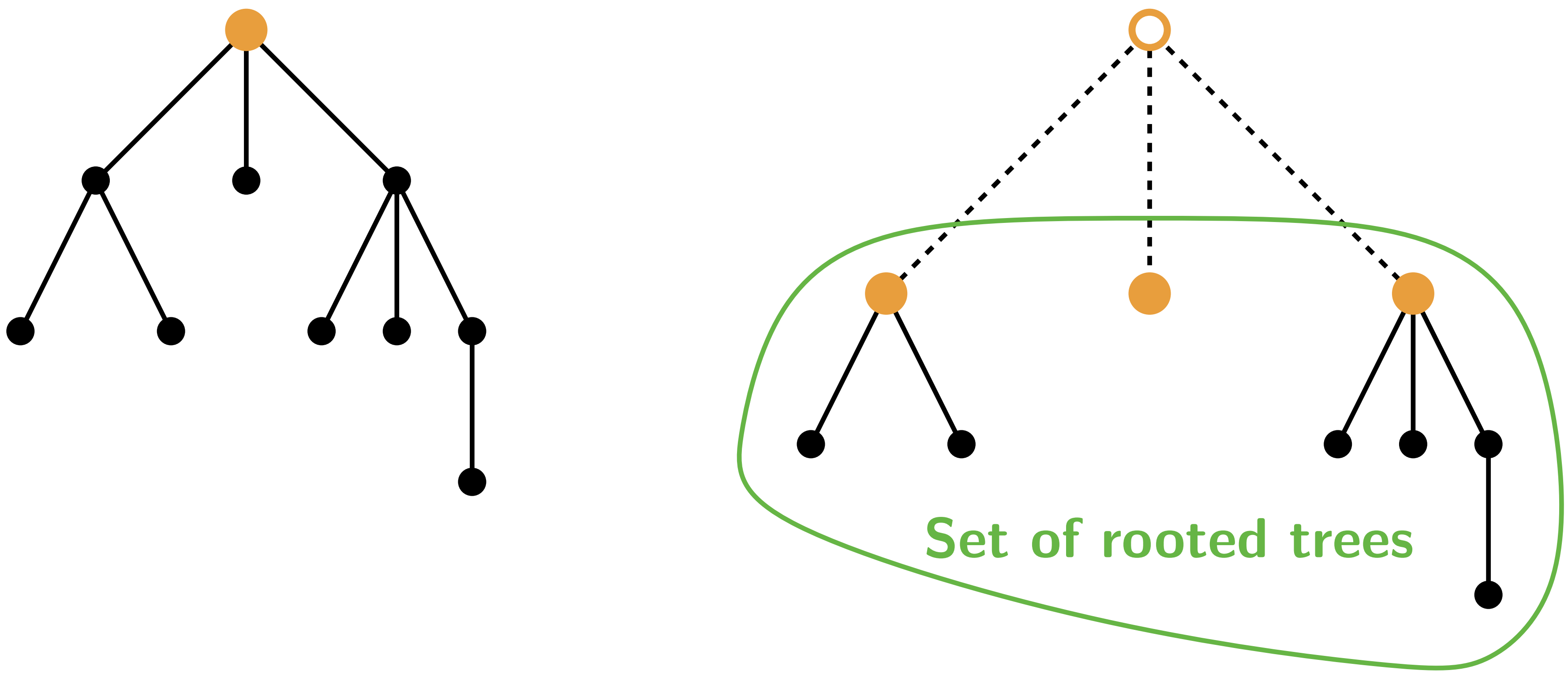
Labeled trees



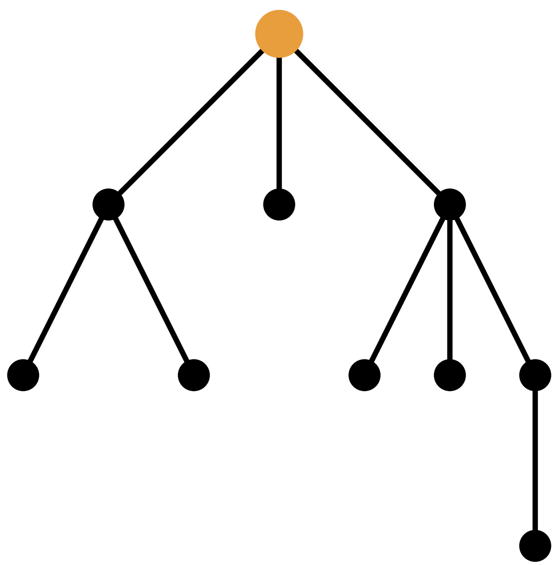
Labeled trees

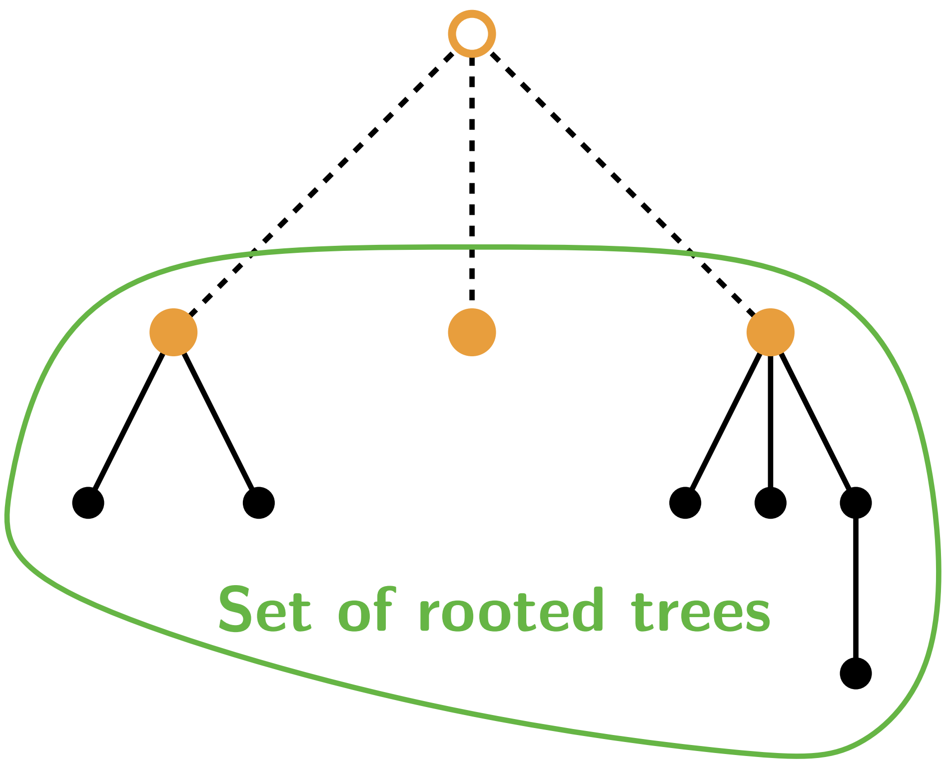


Labeled trees



Labeled trees

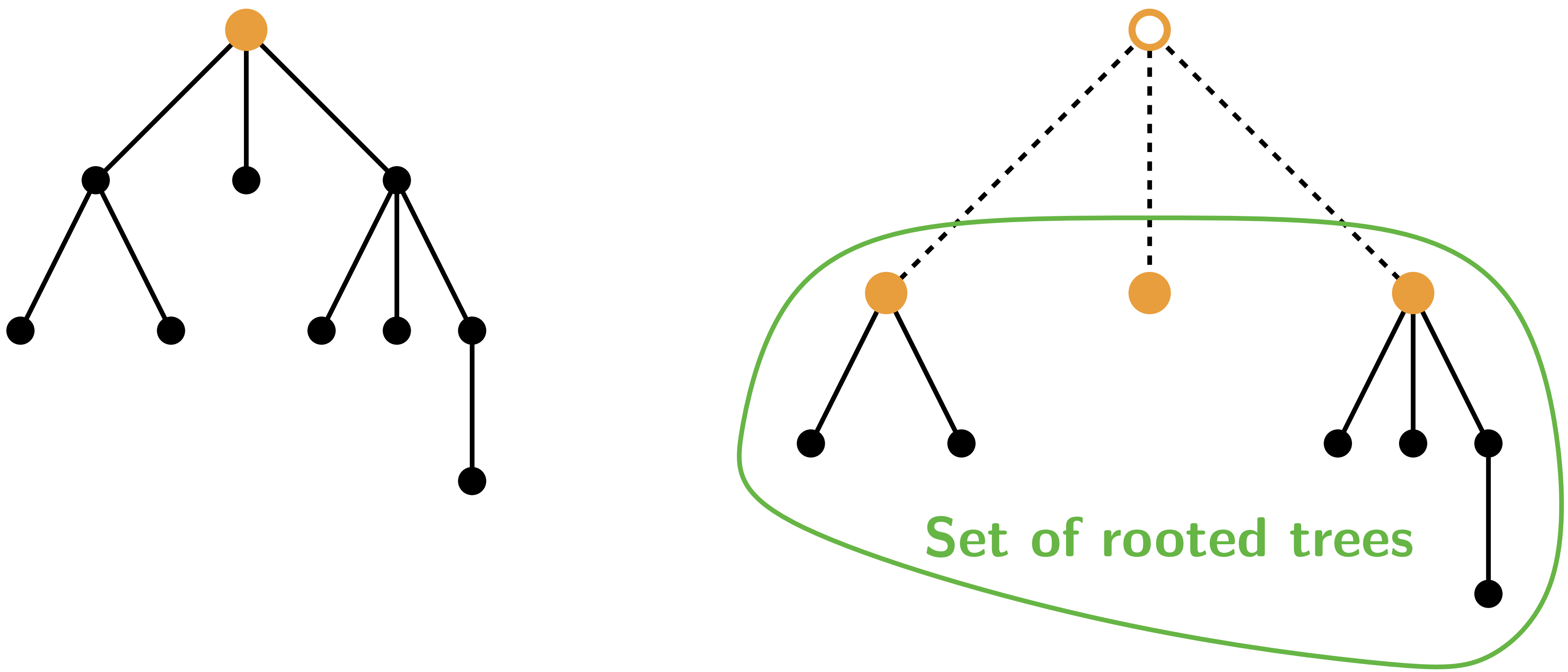




Implicit equation: $\mathcal{T}^\bullet = \{\text{root}\} \times \mathsf{Set}(\mathcal{T}^\bullet)$, which gives

$$T^\bullet(x) = x \exp\left(T^\bullet(x)\right) = x + x^2 + \frac{3x^3}{2} + \cdots .$$

Labeled trees



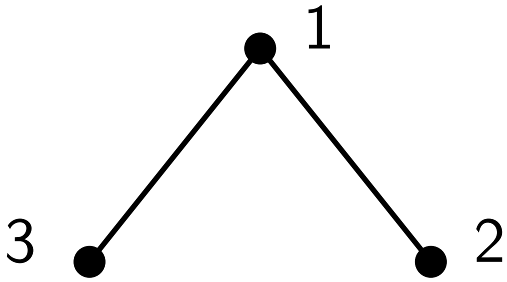
Implicit equation: $\mathcal{T}^{\bullet} = \{\text{root}\} \times \text{Set}(\mathcal{T}^{\bullet})$, which gives

$$T^{\bullet}(x) = x \exp\left(T^{\bullet}(x)\right) = x + x^2 + \frac{3x^3}{2} + \cdots .$$

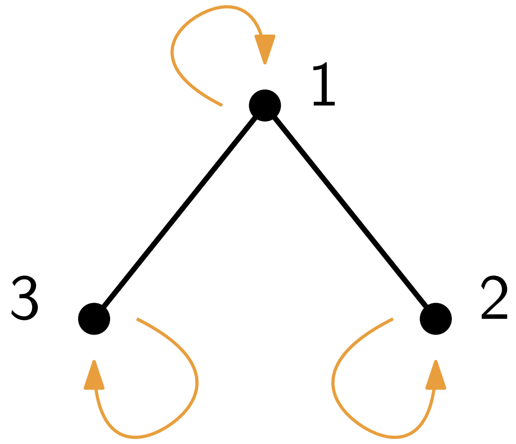
By using the **Lagrange inversion formula** we obtain:

$$|\mathcal{T}_n^{\bullet}| = n![x^n]T^{\bullet}(x) = n^{n-1} \implies |\mathcal{T}_n| = |\mathcal{T}_n^{\bullet}|/n = n^{n-2}.$$

Pólya theory

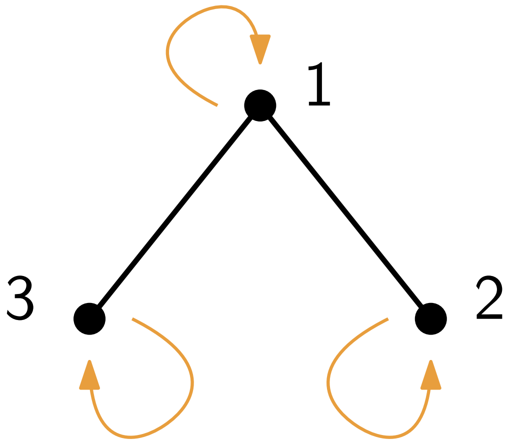


Pólya theory

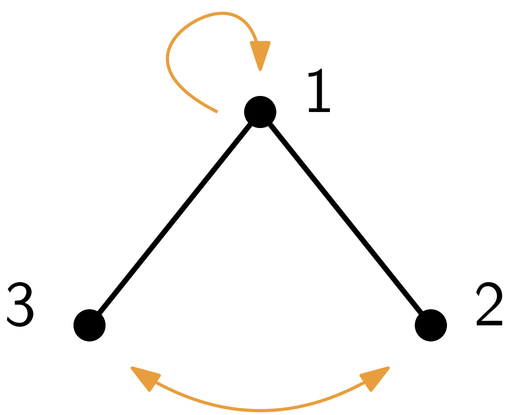


$(1)(2)(3) \longrightarrow s_1^3$

Pólya theory

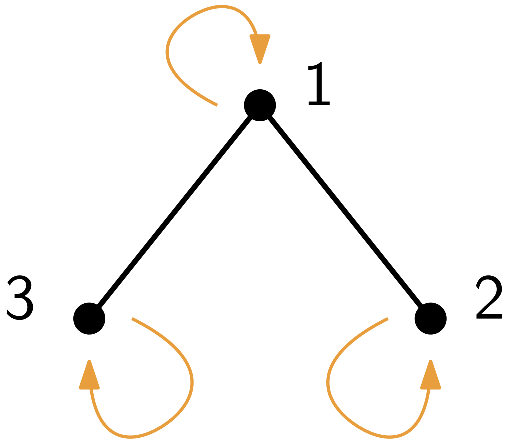


$(1)(2)(3) \longrightarrow s_1^3$

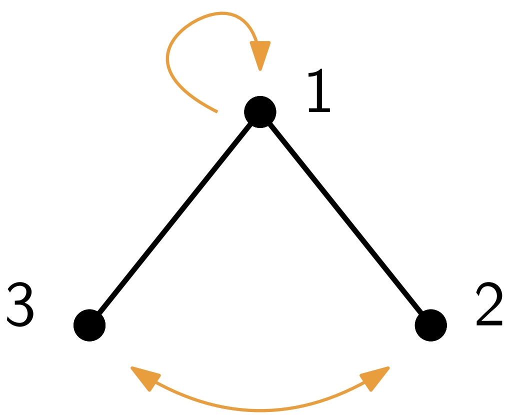


$(1)(23) \longrightarrow s_1 s_2$

Pólya theory



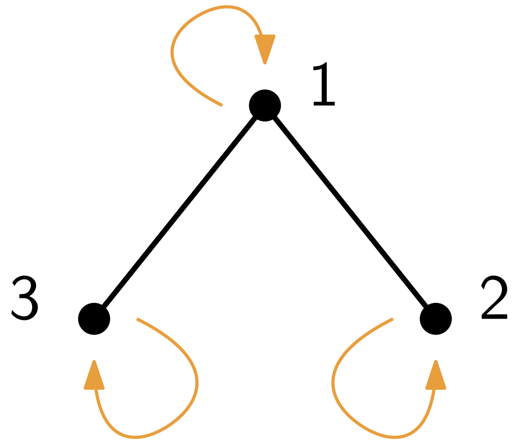
$$(1)(2)(3) \longrightarrow s_1^3$$



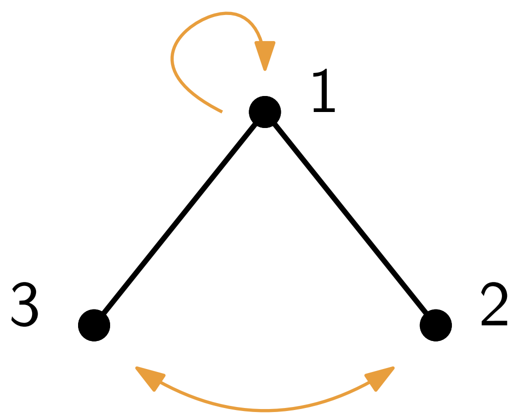
$$(1)(23) \longrightarrow s_1 s_2$$

$$\frac{1}{3!}\left(s_1^3 + s_1 s_2\right)$$

Pólya theory



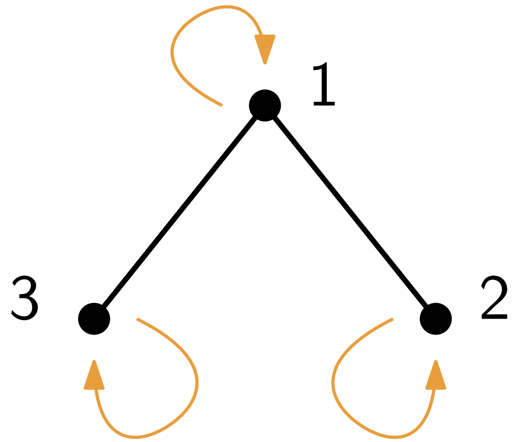
$$(1)(2)(3) \longrightarrow s_1^3$$



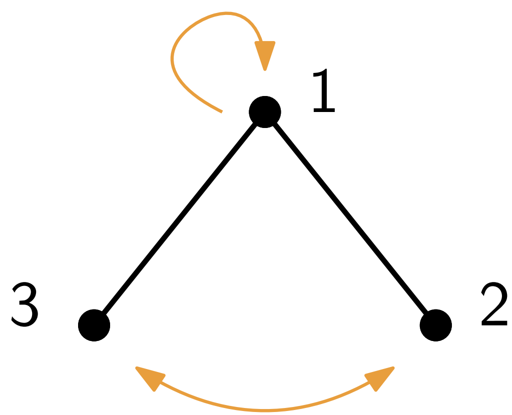
$$(1)(23) \longrightarrow s_1 s_2$$

$$\left.\begin{array}{l} (1)(2)(3) \longrightarrow s_1^3 \\ (1)(23) \longrightarrow s_1 s_2 \end{array}\right\} \quad \begin{array}{c} \dfrac{1}{3!}(s_1^3 + s_1 s_2) \\ \Big\downarrow \ \text{3 labeled graphs in the class} \\ \dfrac{3}{3!}(s_1^3 + s_1 s_2) \end{array}$$

Pólya theory



$(1)(2)(3) \longrightarrow s_1^3$



$(1)(23) \longrightarrow s_1 s_2$

$$\frac{1}{3!}(s_1^3 + s_1 s_2)$$

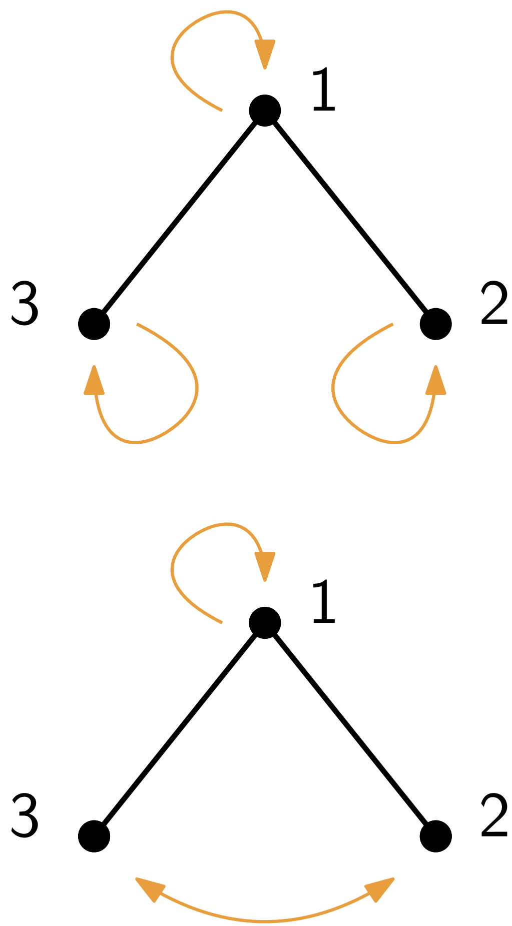
3 labeled graphs in the class

$$\frac{3}{3!}(s_1^3 + s_1 s_2)$$

Cycle index sum

$Z_{\mathcal{G}}(s_1, s_2, s_3, \ldots)$

Pólya theory



$$\left.\begin{array}{l} (1)(2)(3) \longrightarrow s_1^3 \\ \\ (1)(23) \longrightarrow s_1 s_2 \end{array}\right\} \quad \frac{1}{3!}(s_1^3 + s_1 s_2)$$

$$\Big\downarrow \text{ 3 labeled graphs in the class}$$

$$\frac{3}{3!}(s_1^3 + s_1 s_2)$$

Cycle index sum

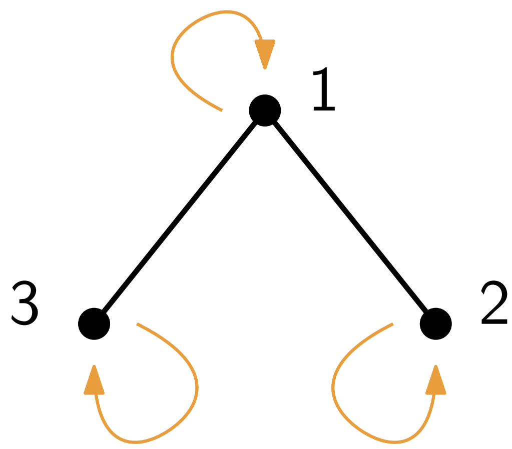
$Z_{\mathcal{G}}(s_1, s_2, s_3, \dots)$

Theorem [Pólya (1937)]

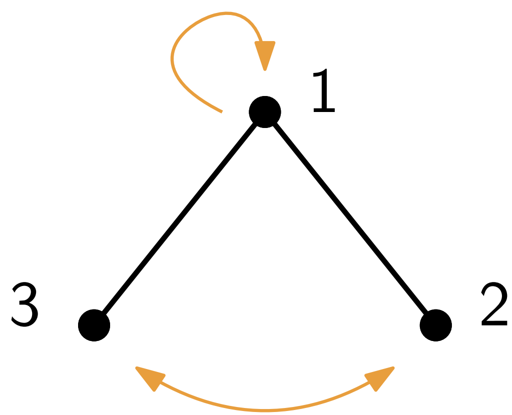
The OGF of the unlabeled class $\tilde{\mathcal{G}}$ is given by

$$\tilde{G}(x) = Z_{\mathcal{G}}(x, x^2, x^3, \dots).$$

Pólya theory



$(1)(2)(3) \longrightarrow s_1^3$



$(1)(23) \longrightarrow s_1 s_2$

$$\frac{1}{3!}(s_1^3 + s_1 s_2)$$

3 labeled graphs in the class

$$\frac{3}{3!}(s_1^3 + s_1 s_2)$$

Cycle index sum

$Z_{\mathcal{G}}(s_1, s_2, s_3, \dots)$

Theorem [Pólya (1937)]
The OGF of the unlabeled class $\tilde{\mathcal{G}}$ is given by

$$\tilde{G}(x) = Z_{\mathcal{G}}(x, x^2, x^3, \dots).$$

In this example,

$$G(x) = \frac{3}{3!}(x^3 + x \cdot x^2) = x^3$$

Pólya theory

Operations between **classes** translate into relations involving their **cycle index sums**.

Class	Cycle index sum
$\mathcal{A} \cup \mathcal{B}$	$Z_{\mathcal{A}}(s_1, s_2, s_3, \dots) + Z_{\mathcal{B}}(s_1, s_2, s_3, \dots)$
$\mathcal{A} \times \mathcal{B}$	$Z_{\mathcal{A}}(s_1, s_2, s_3, \dots) \cdot Z_{\mathcal{B}}(s_1, s_2, s_3, \dots)$
$\mathcal{A} \circ \mathcal{B}$	$Z_{\mathcal{A}}(Z_{\mathcal{B}}(s_1, s_2, s_3, \dots), Z_{\mathcal{B}}(s_2, s_4, s_6, \dots), \dots)$
Seq$(\mathcal{A})$	$\frac{1}{1-Z_{\mathcal{A}}(s_1,s_2,s_3,\dots)}$
Set$(\mathcal{A})$	$\exp\left(\sum_{k\geq 1} \frac{1}{k} Z_{\mathcal{A}}(s_k, s_{2k}, s_{3k}, \dots)\right)$

Unlabeled trees

Pólya trees: rooted, unlabeled trees.

Unlabeled trees

Pólya trees: rooted, unlabeled trees.

Theorem. [Pólya (1937)]
The OGF $P(x)$ of Pólya trees is given by

$$P(x) = x \exp\left(P(x) + \frac{P(x^2)}{2} + \frac{P(x^3)}{3} + \dots\right).$$

As $n \to \infty$ we have

$$[x^n]P(x) \sim \frac{b\sqrt{\rho}}{2\sqrt{\pi}} \cdot n^{-3/2} \cdot \rho^{-n},$$

with $b \approx 2.681127$ and $\rho \approx 0.338219$.

Unlabeled trees

Pólya trees: rooted, unlabeled trees.

Theorem. [Pólya (1937)]
The OGF $P(x)$ of Pólya trees is given by

$$P(x) = x \exp(P(x) + \frac{P(x^2)}{2} + \frac{P(x^3)}{3} + \dots).$$

As $n \to \infty$ we have

$$[x^n]P(x) \sim \frac{b\sqrt{\rho}}{2\sqrt{\pi}} \cdot n^{-3/2} \cdot \rho^{-n},$$

with $b \approx 2.681127$ and $\rho \approx 0.338219$.

What about unrooted unlabeled trees?

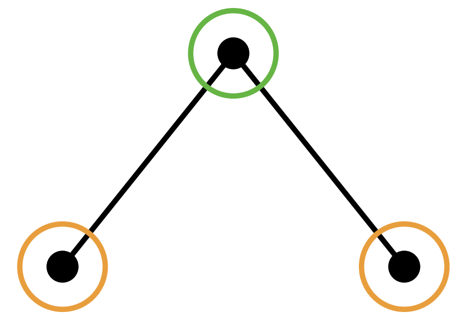
Unlabeled trees

Problem!

Unlabeled trees

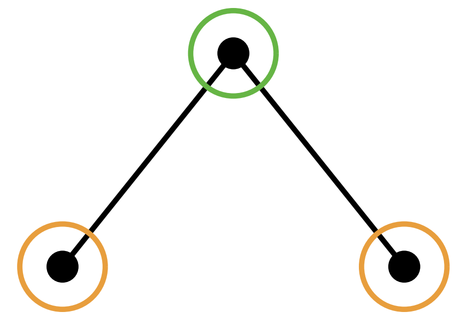
Problem!

Rooting is biased in unlabeled graphs.
Not every unlabelled graph of size n gives rise to n rooted graphs.



Unlabeled trees

Problem!
Rooting is biased in unlabeled graphs.
Not every unlabelled graph of size n gives rise to n rooted graphs.



Theorem. [Otter (1948)]
The OGF $U(x)$ of unlabeled trees is given by

$$U(x) = P(x) + \frac{1}{2}(P(x^2) - P(x)^2).$$

As $n \to \infty$ we have

$$[x^n]P(x) \sim \frac{b^3 \rho^{3/2}}{4\sqrt{\pi}} \cdot n^{-3/2} \cdot \rho^{-n},$$

with $b \approx 2.681127$ and $\rho \approx 0.338219$.

Proof. Using the **dissymmetry theorem**.

The dissymmetry theorem

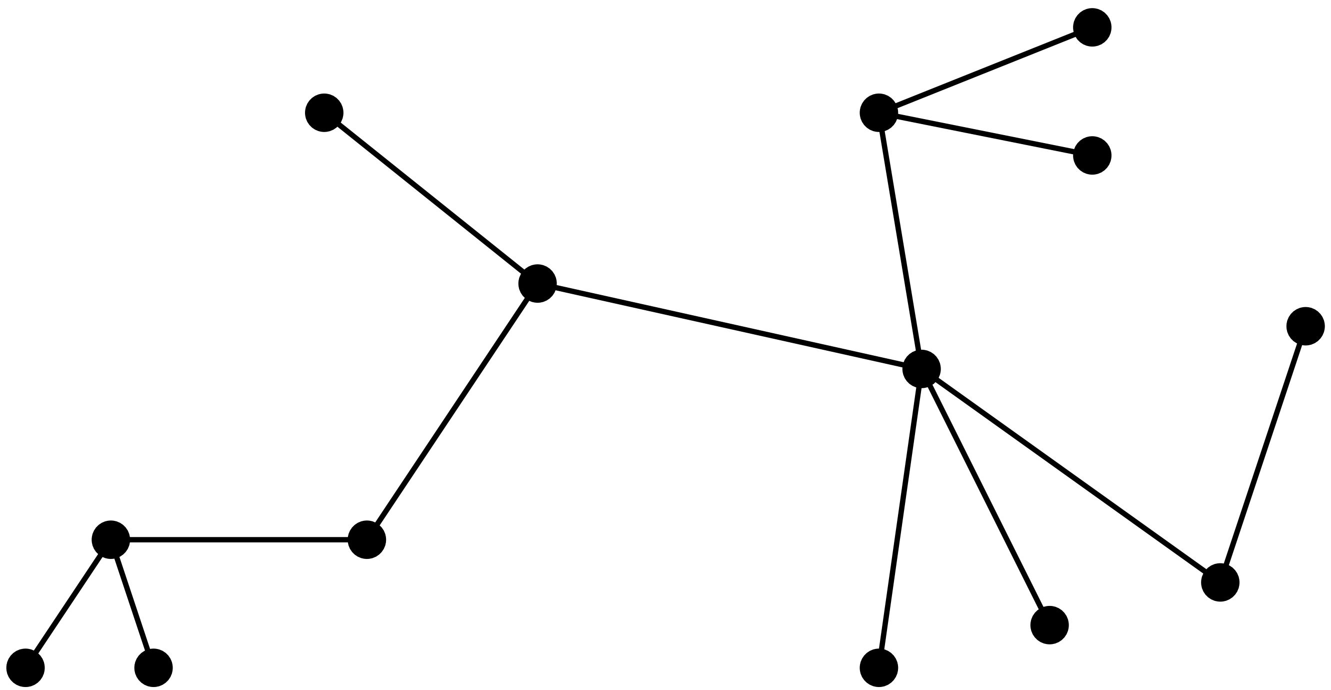
Theorem. Let $\mathcal{T}$ be a class of trees. Then,

$$\mathcal{T} + \mathcal{T}^{\bullet \rightarrow \bullet} \simeq \mathcal{T}^{\bullet} + \mathcal{T}^{\bullet - \bullet}$$

The dissymmetry theorem

Theorem. Let $\mathcal{T}$ be a class of trees. Then,

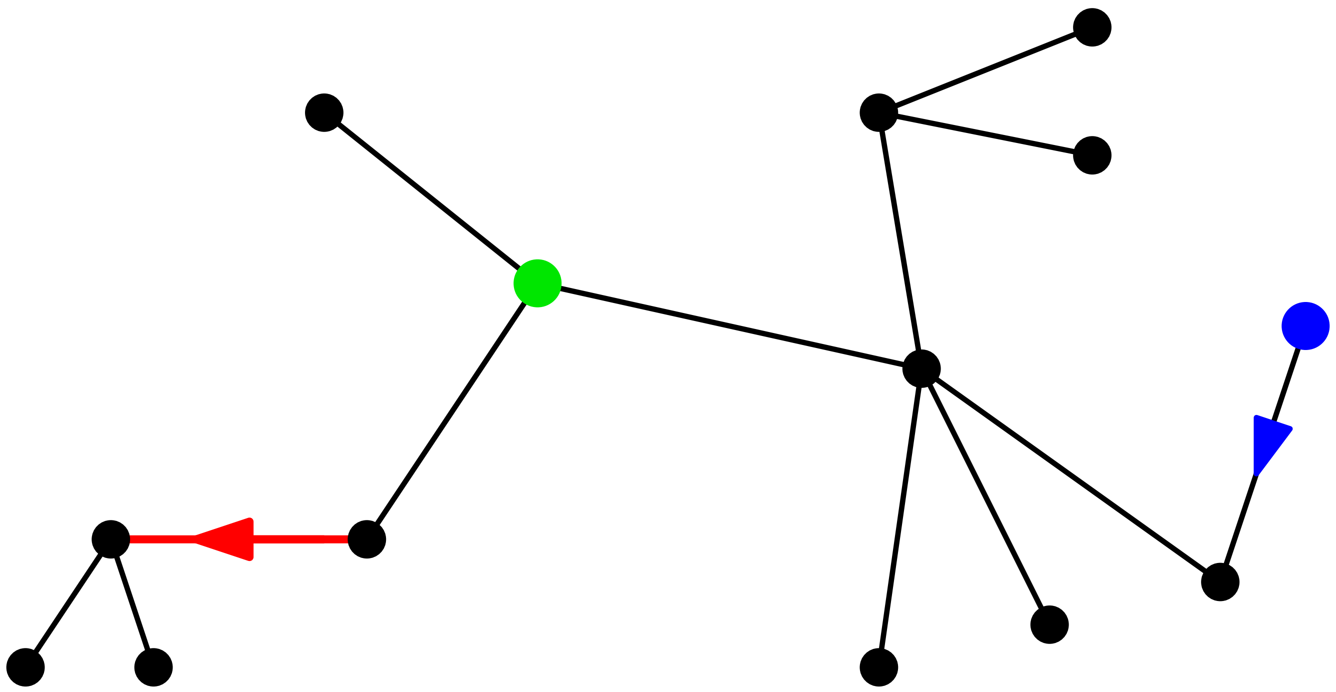
$$\mathcal{T} + \mathcal{T}^{\bullet\to\bullet} \simeq \mathcal{T}^{\bullet} + \mathcal{T}^{\bullet-\bullet}$$



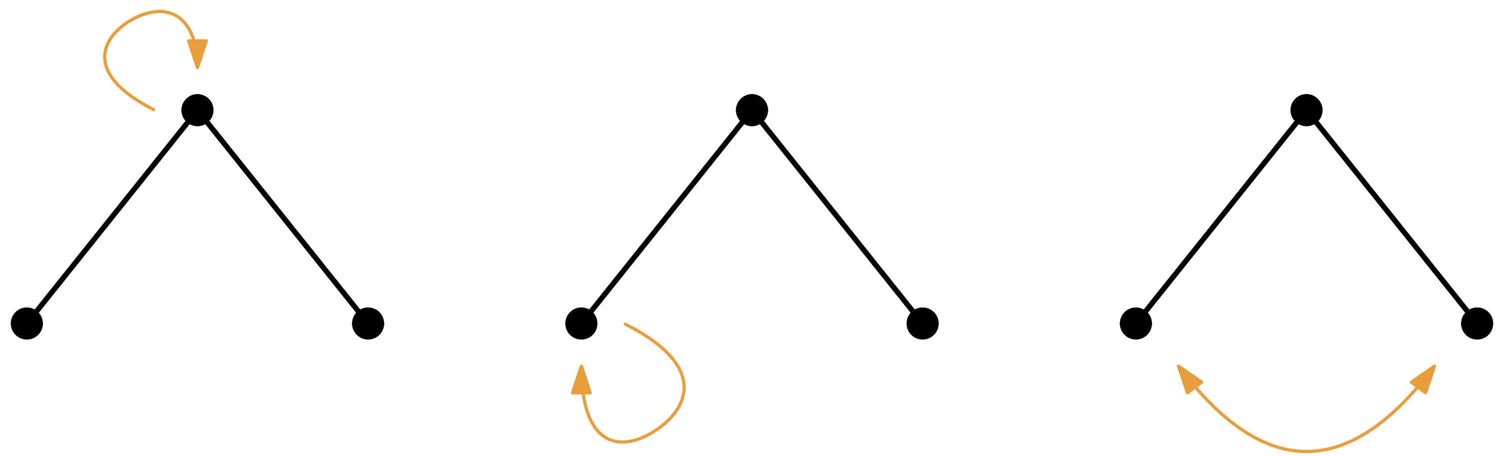
The dissymmetry theorem

Theorem. Let $\mathcal{T}$ be a class of trees. Then,

$$\mathcal{T} + \mathcal{T}^{\bullet\to\bullet} \simeq \mathcal{T}^{\bullet} + \mathcal{T}^{\bullet-\bullet}$$

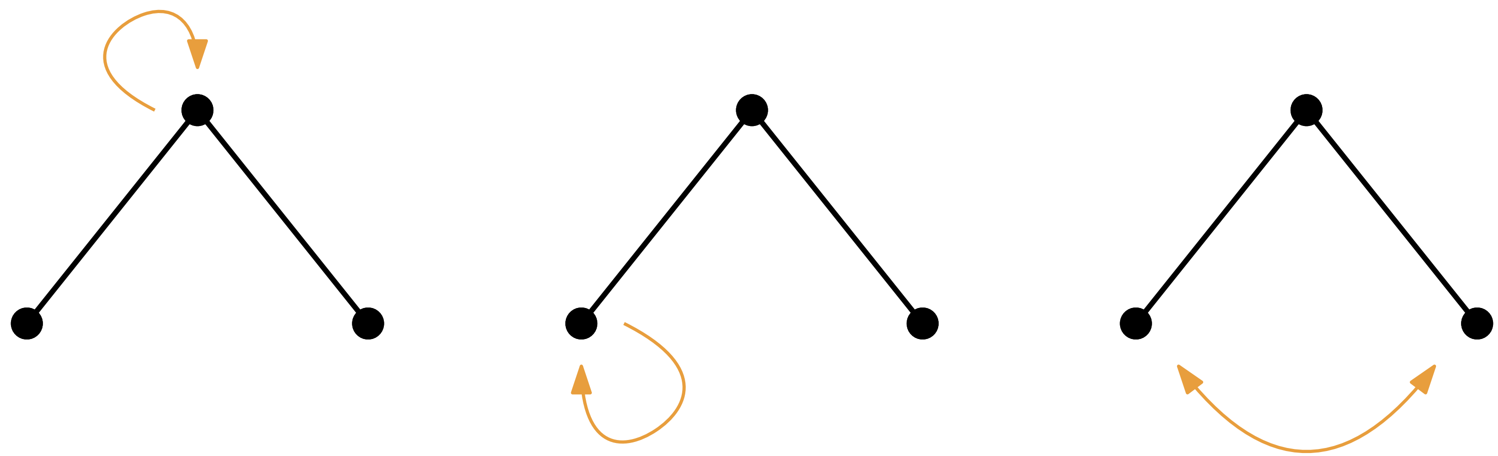


Cycle-pointing



Definition. A **cycle-pointed graph** is a pair (G, c) where $G \in \mathcal{G}$ is an unlabelled graph and c is a cycle of some automorphism of G.

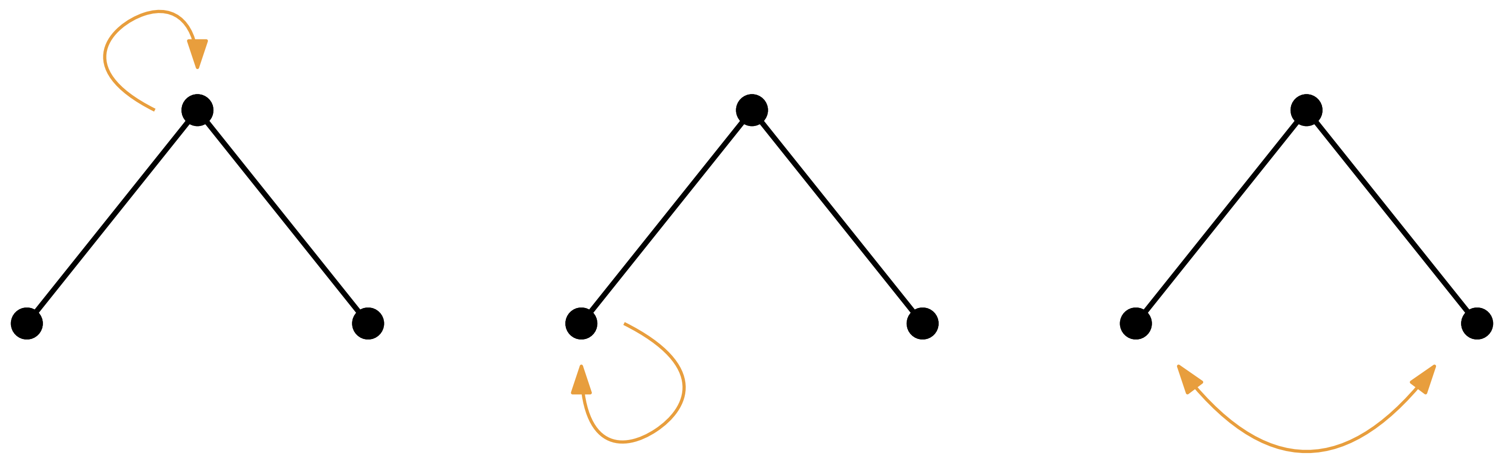
Cycle-pointing



Definition. A **cycle-pointed graph** is a pair (G, c) where $G \in \mathcal{G}$ is an unlabelled graph and c is a cycle of some automorphism of G.

Theorem. [Bodirsky, Fusy, Kang & Vigerske (2007)]
Every unlabelled graph $G \in \mathcal{G}$ of size n admits exactly n cycle-pointings.

Cycle-pointing

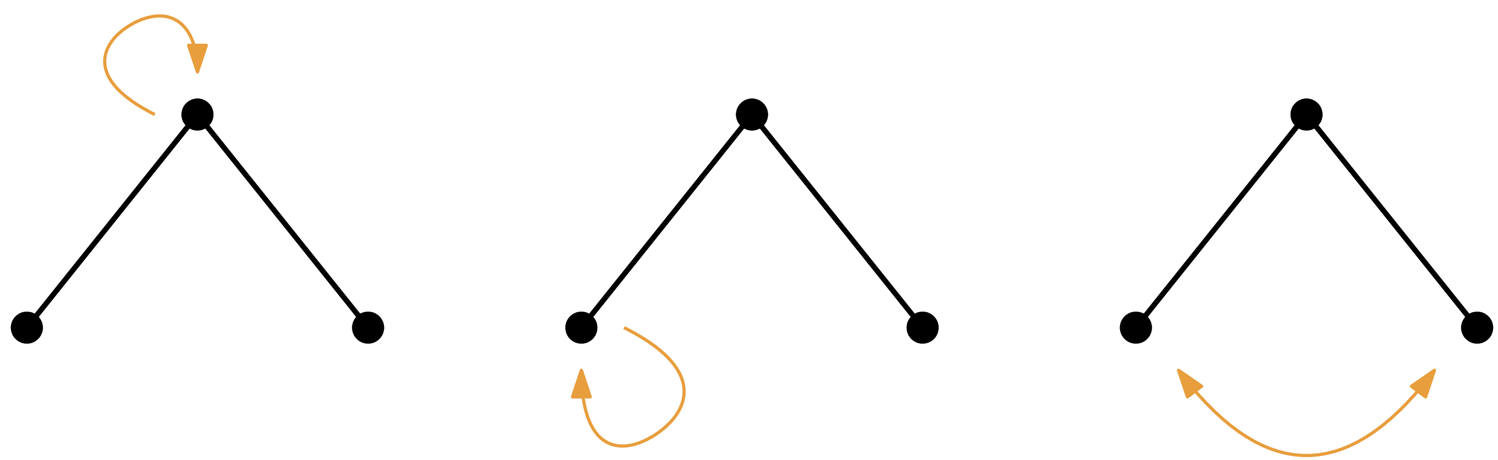


Definition. A **cycle-pointed graph** is a pair (G, c) where $G \in \mathcal{G}$ is an unlabelled graph and c is a cycle of some automorphism of G.

Theorem. [Bodirsky, Fusy, Kang & Vigerske (2007)]
Every unlabelled graph $G \in \mathcal{G}$ of size n admits exactly n cycle-pointings.

An unbiased rooting (pointing) operator!

Cycle-pointing



Definition. A **cycle-pointed graph** is a pair (G, c) where $G \in \mathcal{G}$ is an unlabelled graph and c is a cycle of some automorphism of G.

Theorem. [Bodirsky, Fusy, Kang & Vigerske (2007)]
Every unlabelled graph $G \in \mathcal{G}$ of size n admits exactly n cycle-pointings.

An unbiased rooting (pointing) operator!

They extend Pólya theory to cycle-pointed graphs. In particular, they manage to unroot Pólya trees via cycle-pointing and they recover Otter's formula.

Beyond trees

What is a tree?

Beyond trees

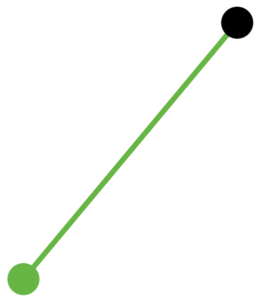
What is a tree?

Iteratively add a new vertex connected to an existing vertex.

Beyond trees

What is a tree?

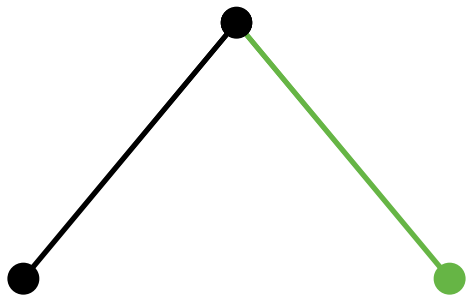
Iteratively add a new vertex connected to an existing vertex.



Beyond trees

What is a tree?

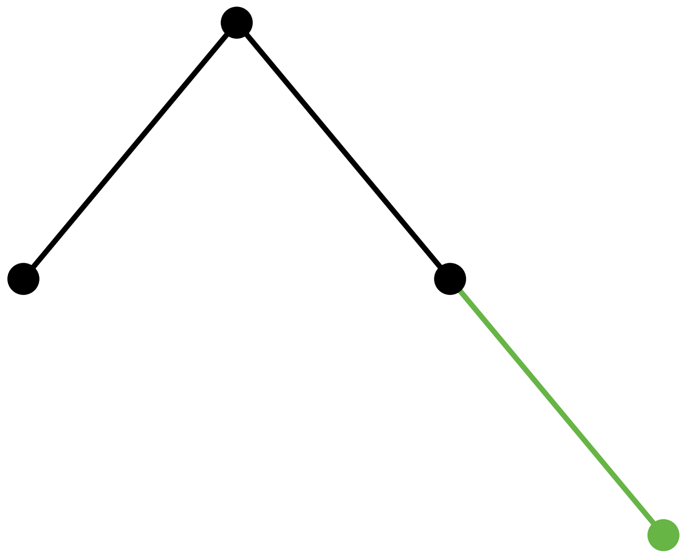
Iteratively add a new vertex connected to an existing vertex.



Beyond trees

What is a tree?

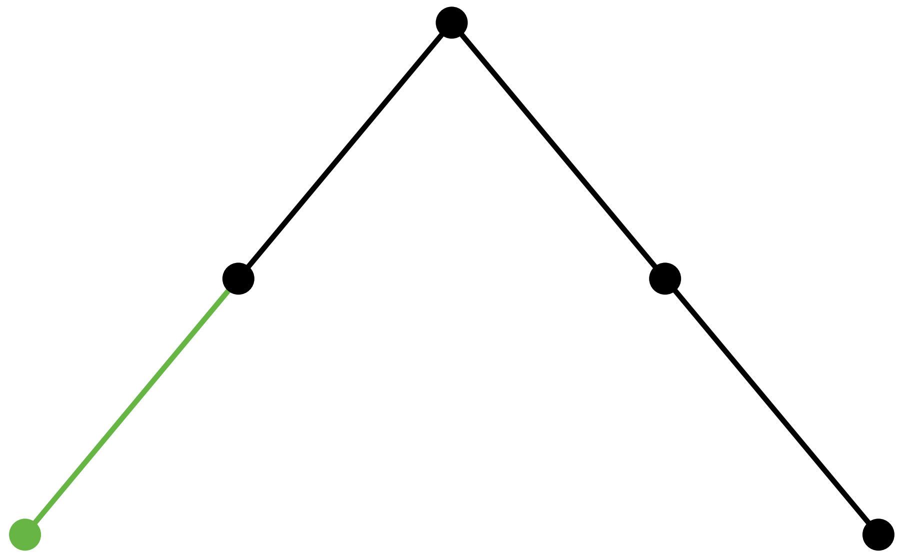
Iteratively add a new vertex connected to an existing vertex.



Beyond trees

What is a tree?

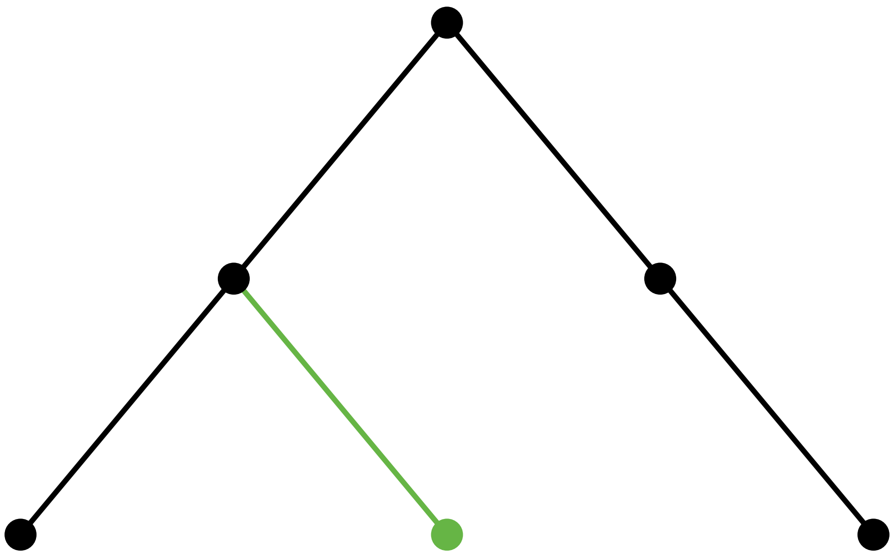
Iteratively add a new vertex connected to an existing vertex.



Beyond trees

What is a tree?

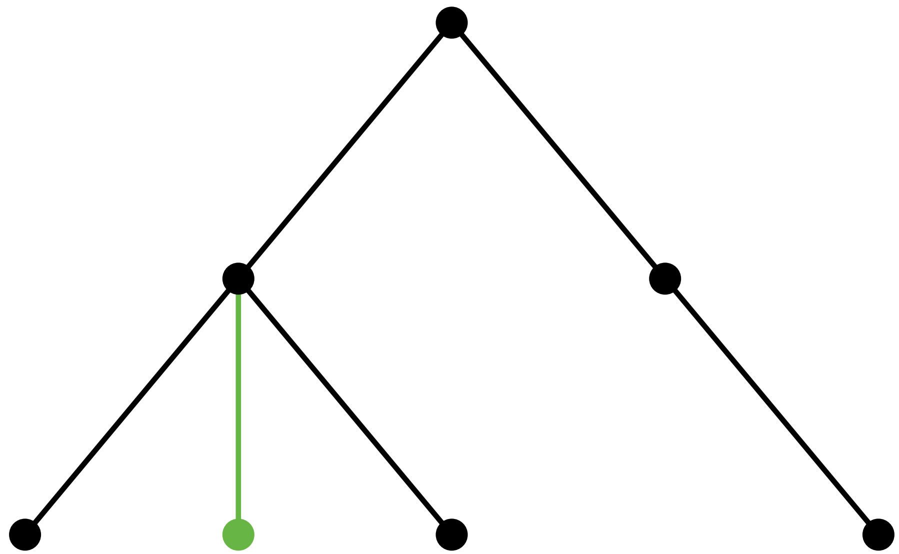
Iteratively add a new vertex connected to an existing vertex.



Beyond trees

What is a tree?

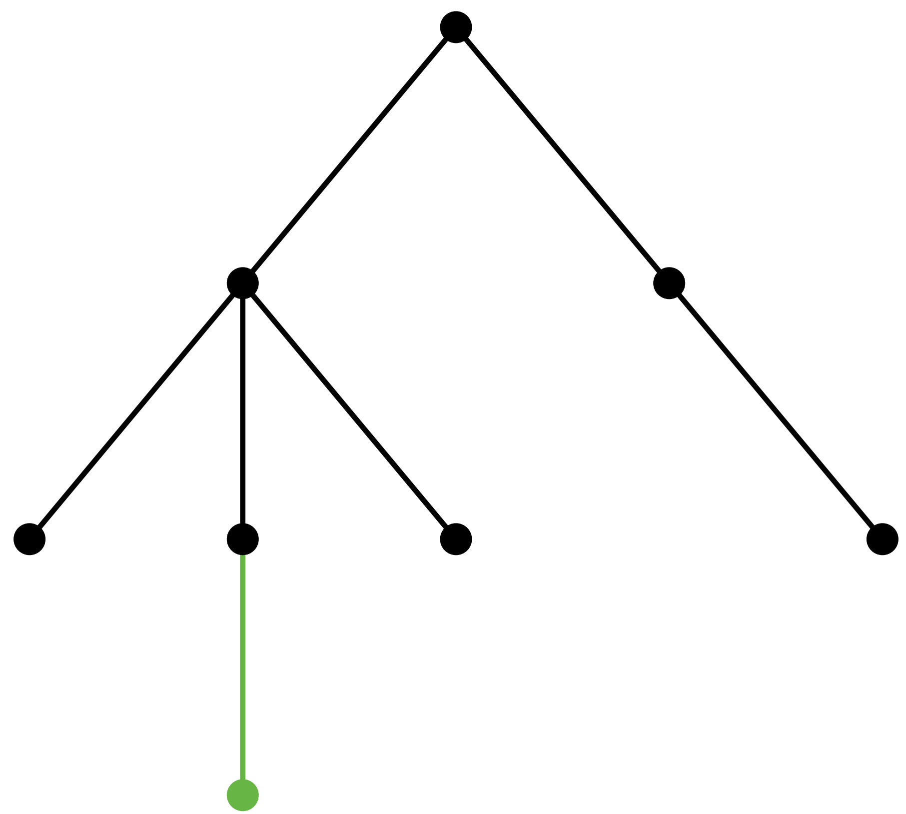
Iteratively add a new vertex connected to an existing vertex.



Beyond trees

What is a tree?

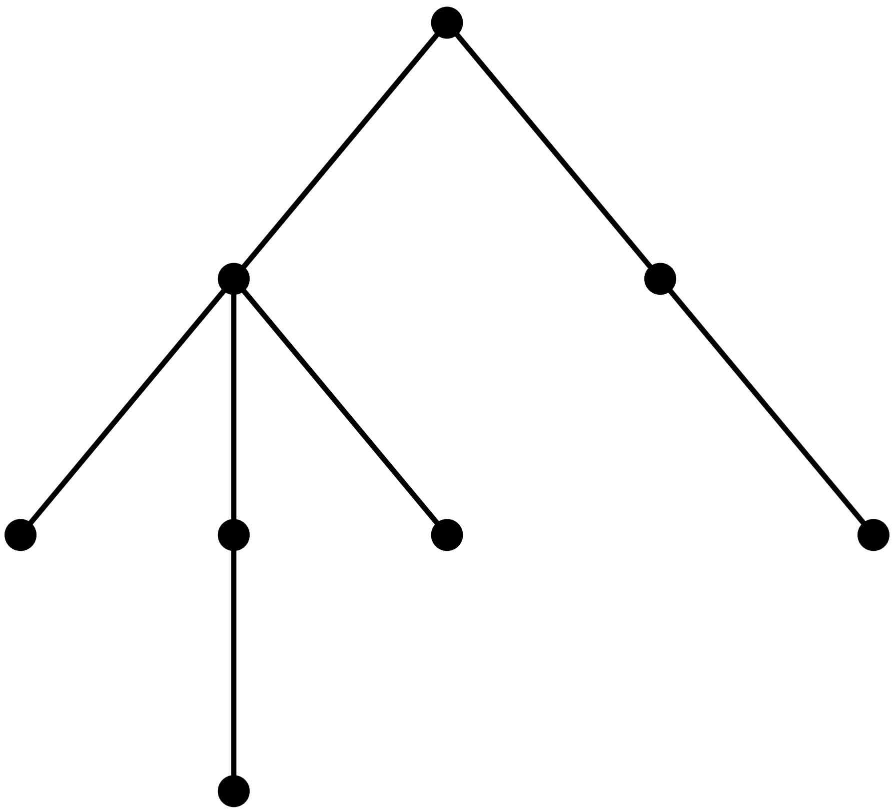
Iteratively add a new vertex connected to an existing vertex.



Beyond trees

What is a tree?

Iteratively add a new vertex connected to an existing vertex.



Beyond trees

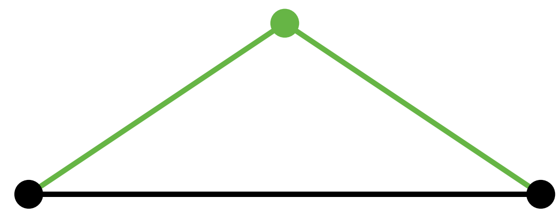
Iteratively add a new vertex connected to the vertices of an existing **edge**.

Beyond trees

Iteratively add a new vertex connected to the vertices of an existing **edge**.

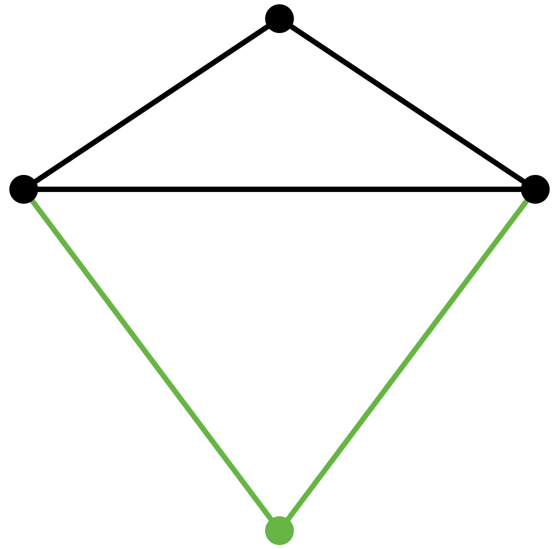
Beyond trees

Iteratively add a new vertex connected to the vertices of an existing **edge**.



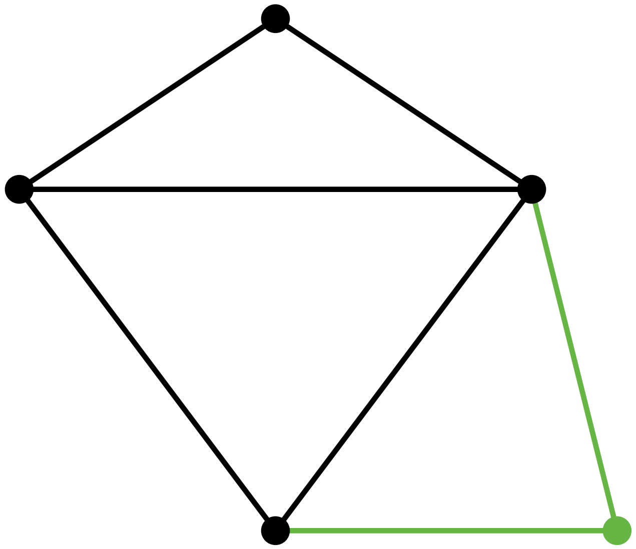
Beyond trees

Iteratively add a new vertex connected to the vertices of an existing **edge**.



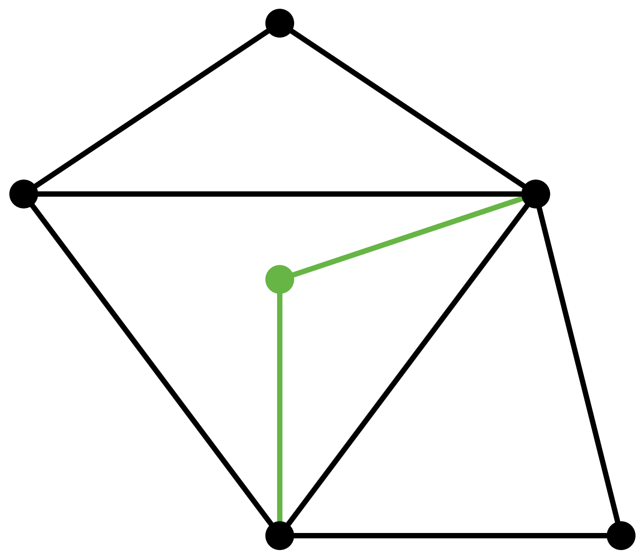
Beyond trees

Iteratively add a new vertex connected to the vertices of an existing **edge**.



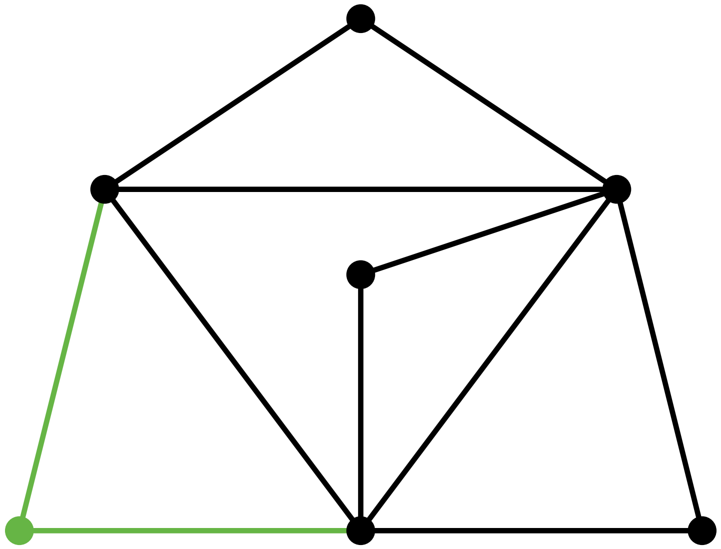
Beyond trees

Iterativelyadd a new vertex connected to the vertices of an existing **edge**.



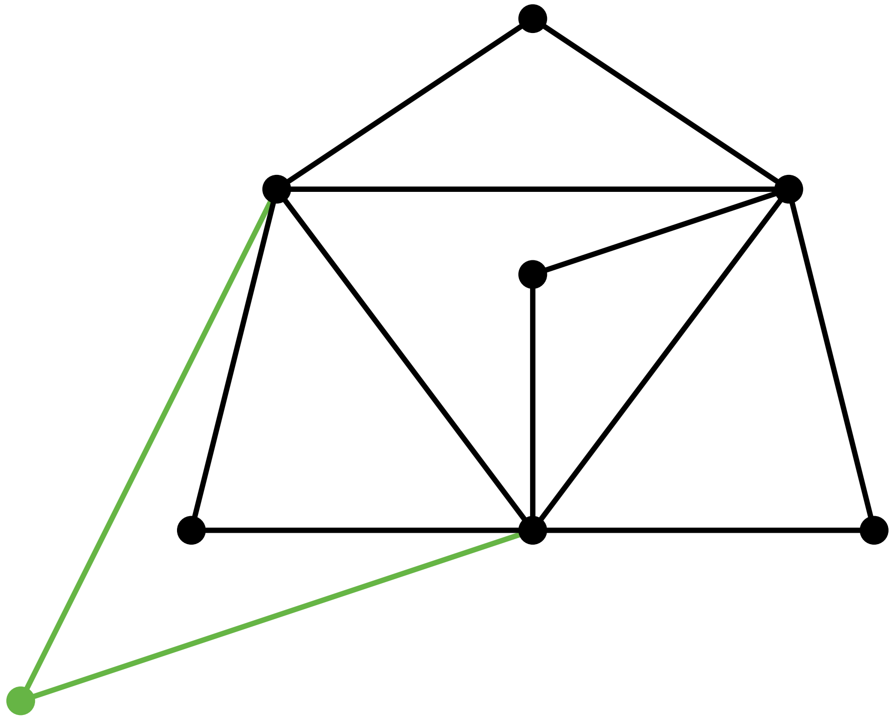
Beyond trees

Iteratively add a new vertex connected to the vertices of an existing **edge**.



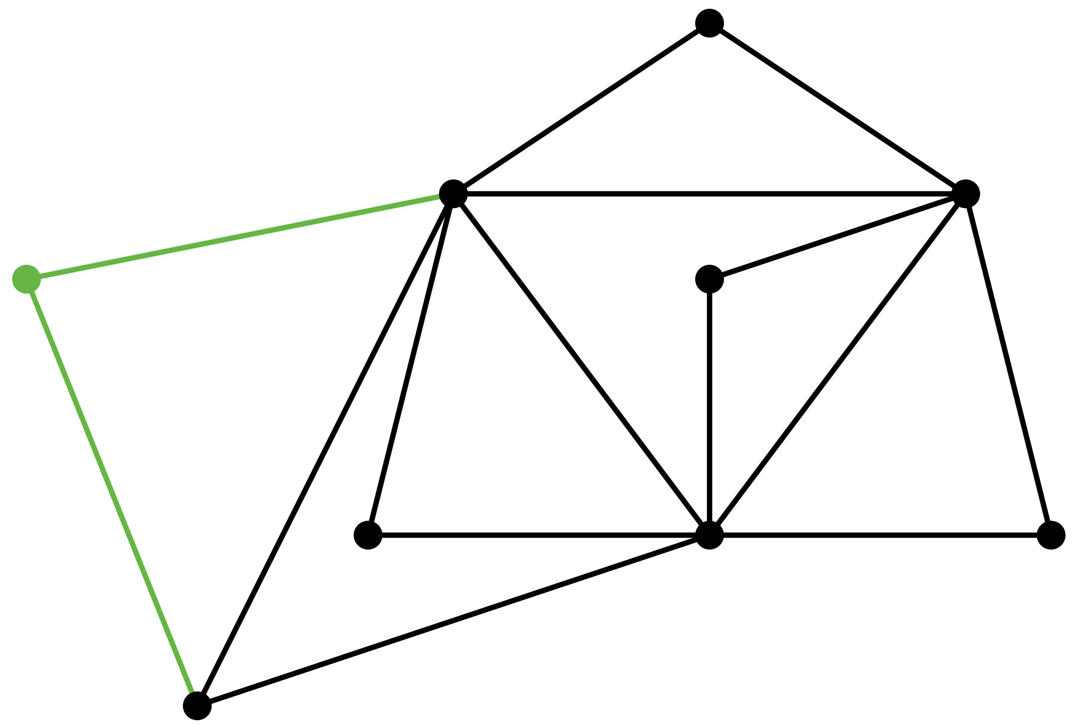
Beyond trees

Iteratively add a new vertex connected to the vertices of an existing **edge**.



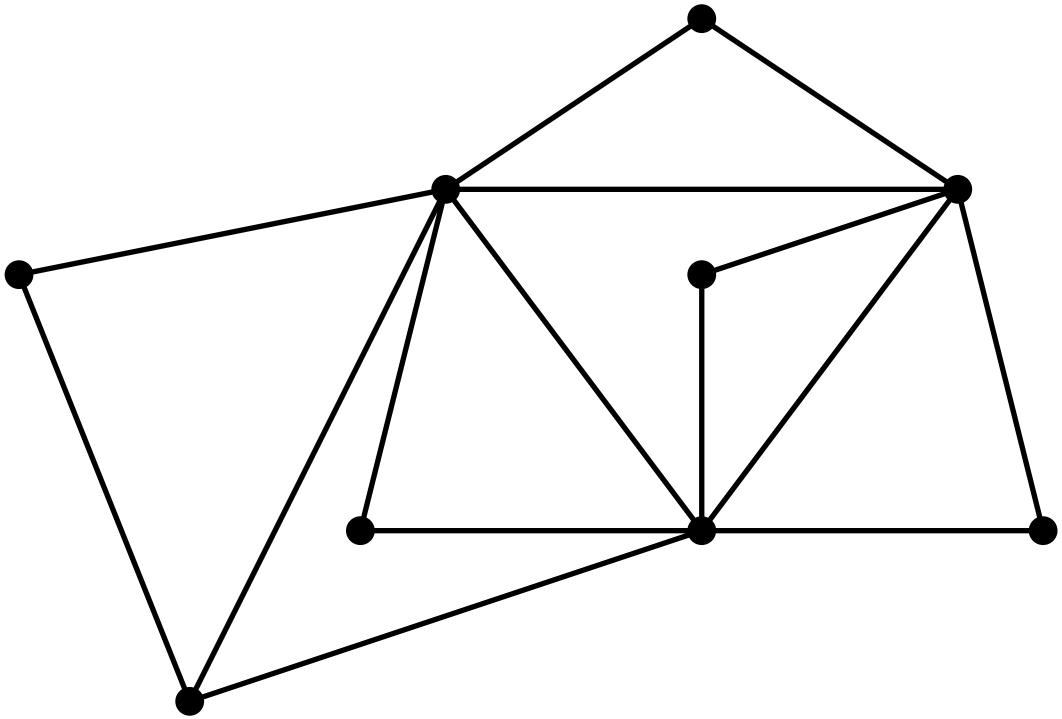
Beyond trees

Iteratively add a new vertex connected to the vertices of an existing **edge**.



Beyond trees

Iteratively add a new vertex connected to the vertices of an existing **edge**.



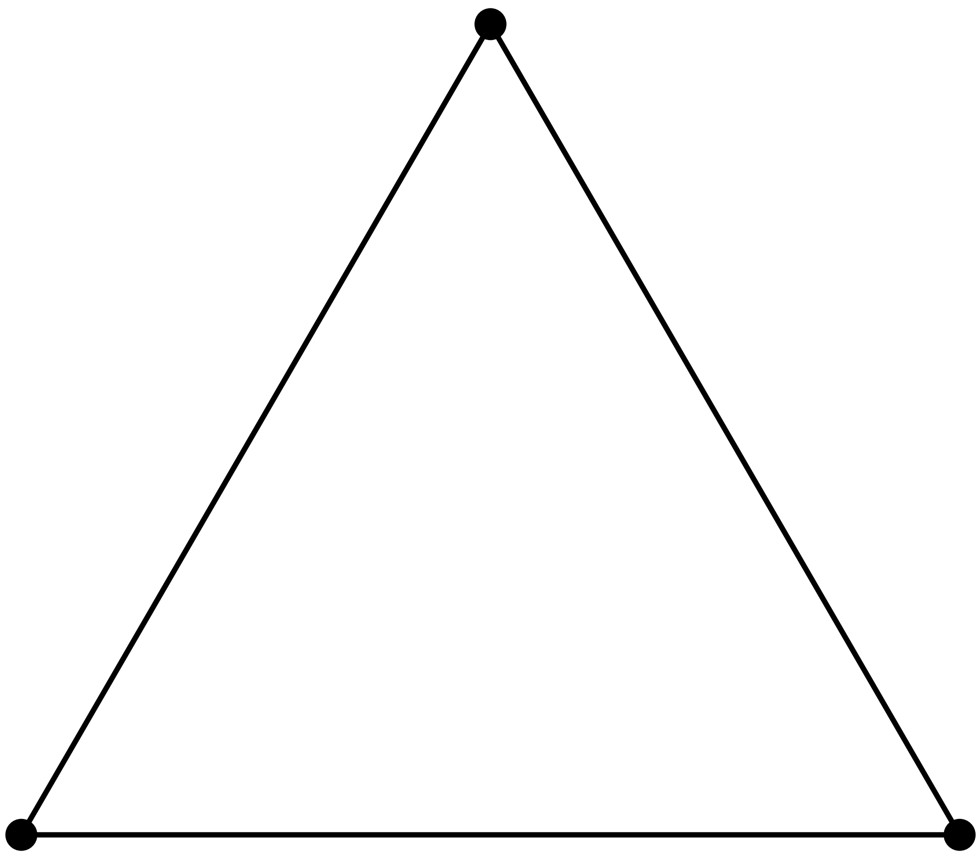
2-trees

Beyond trees

Iteratively add a new vertex connected to the vertices of an existing **triangle**.

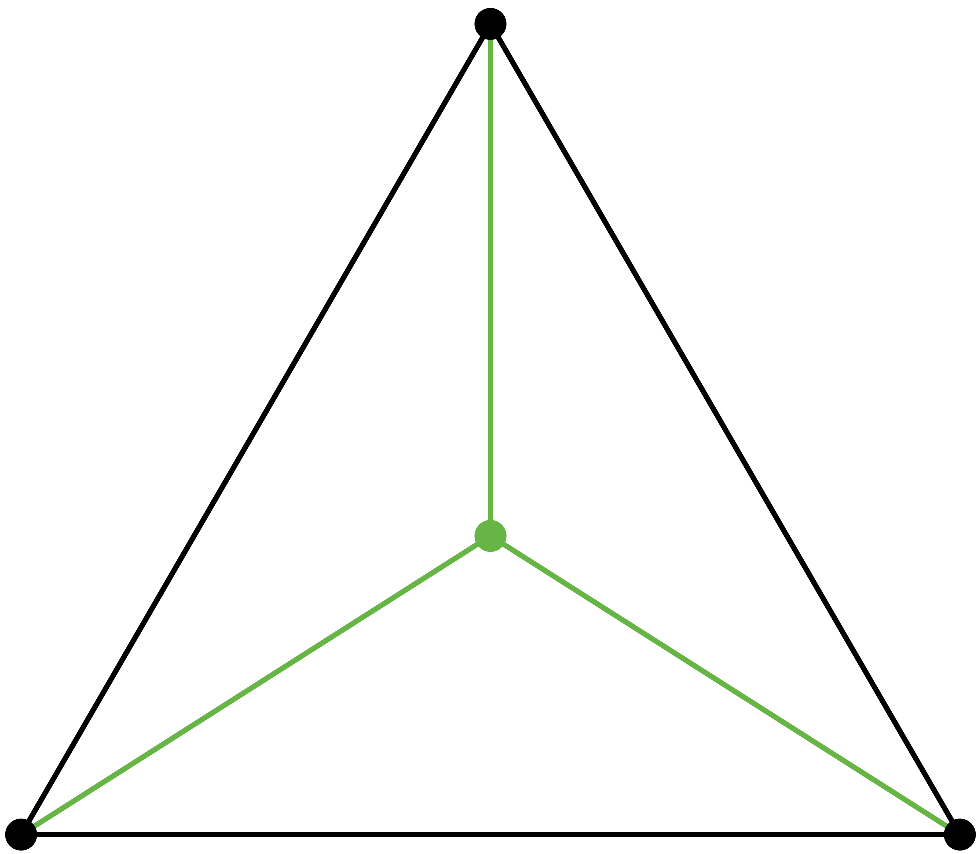
Beyond trees

Iteratively add a new vertex connected to the vertices of an existing **triangle**.



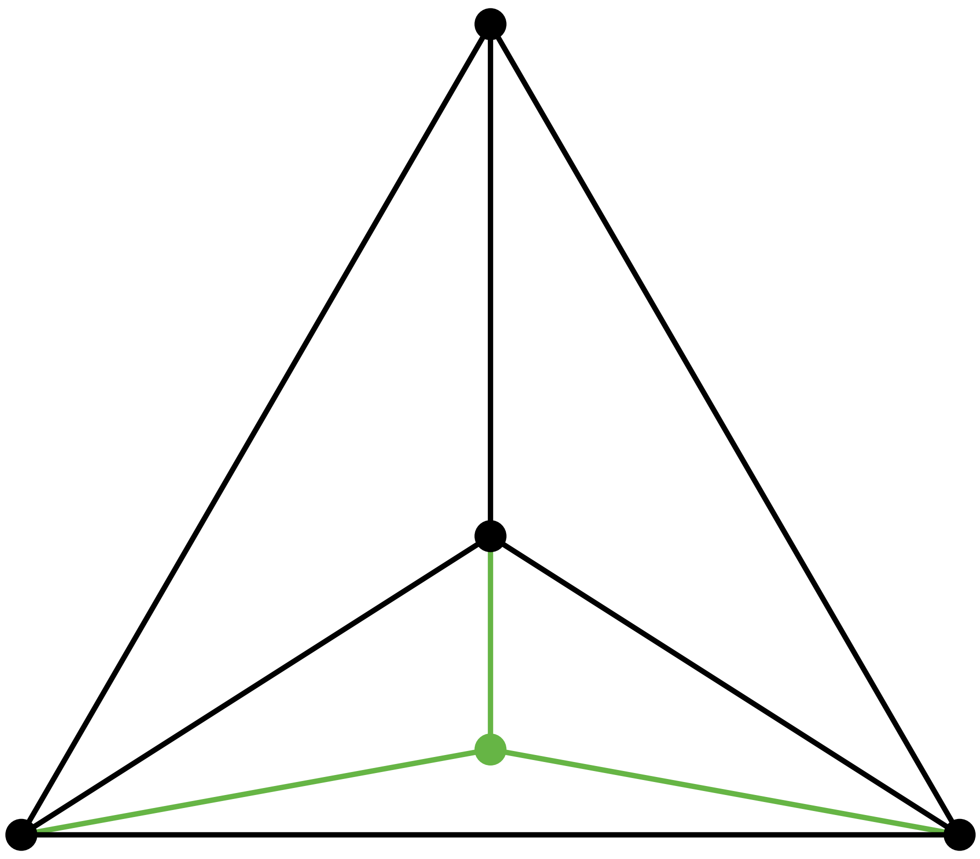
Beyond trees

Iteratively add a new vertex connected to the vertices of an existing **triangle**.



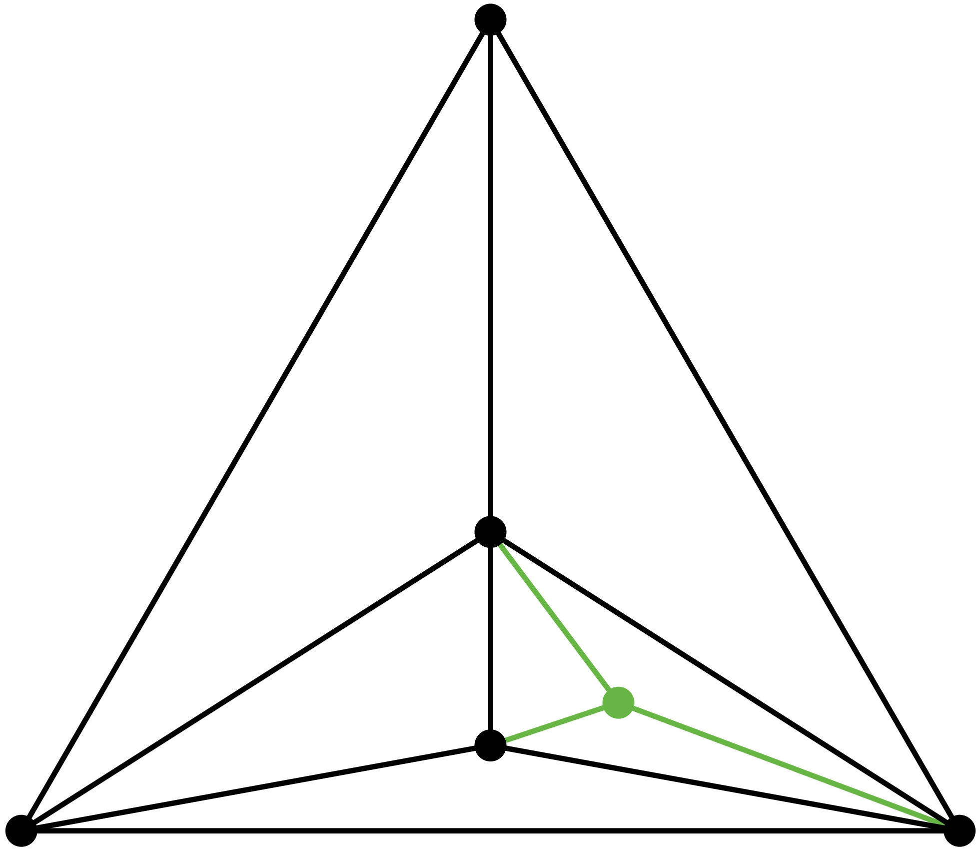
Beyond trees

Iteratively add a new vertex connected to the vertices of an existing **triangle**.



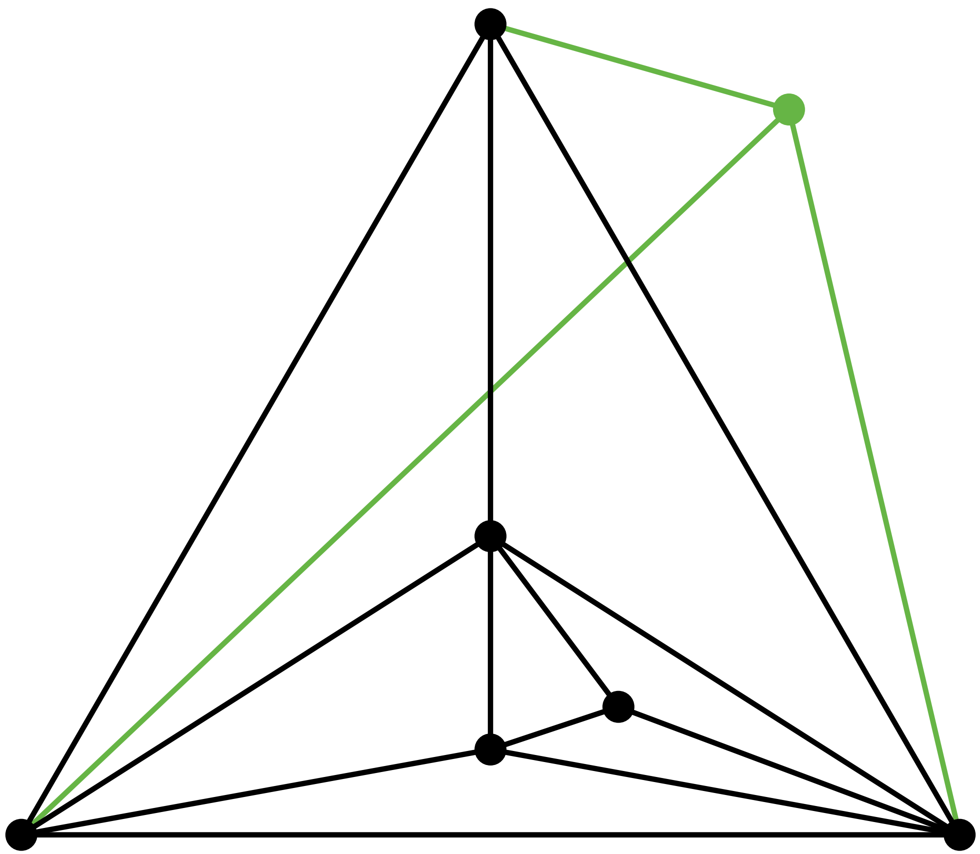
Beyond trees

Iteratively add a new vertex connected to the vertices of an existing **triangle**.



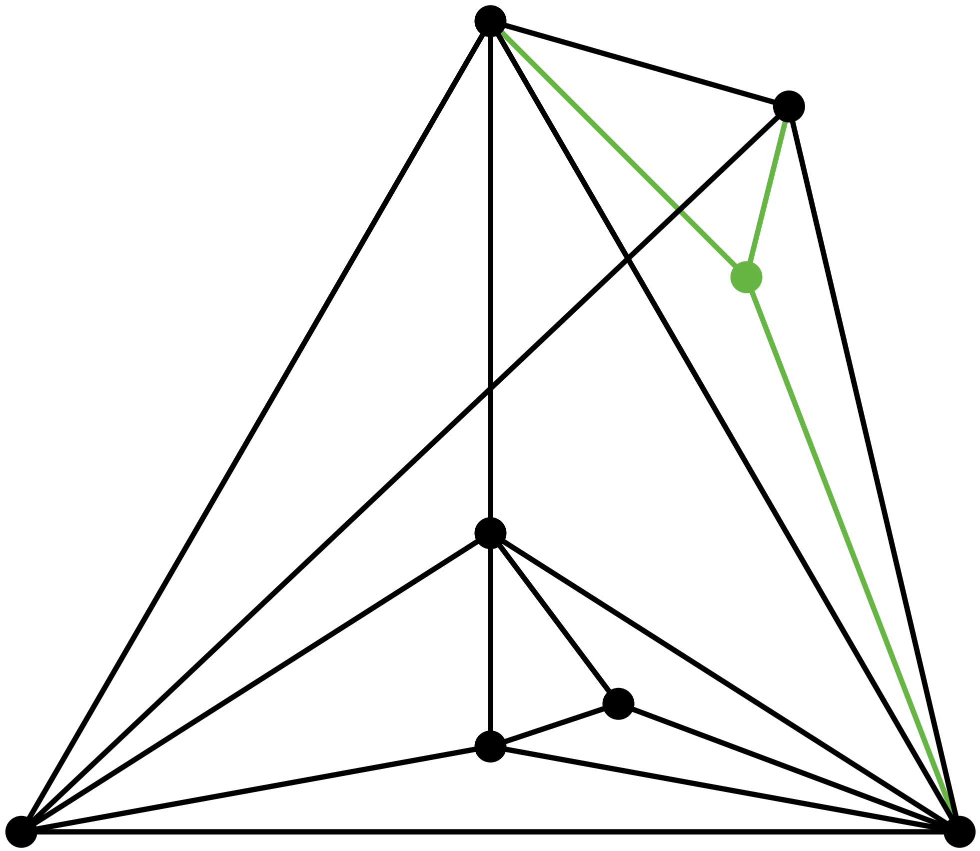
Beyond trees

Iteratively add a new vertex connected to the vertices of an existing **triangle**.



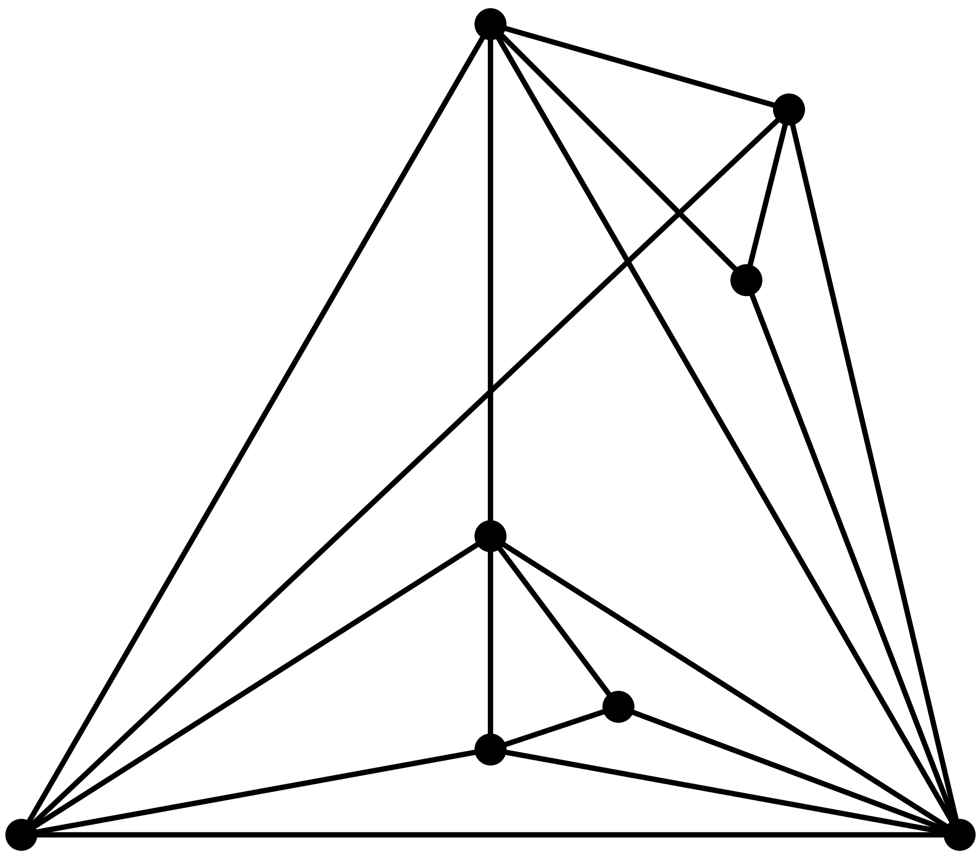
Beyond trees

Iteratively add a new vertex connected to the vertices of an existing **triangle**.



Beyond trees

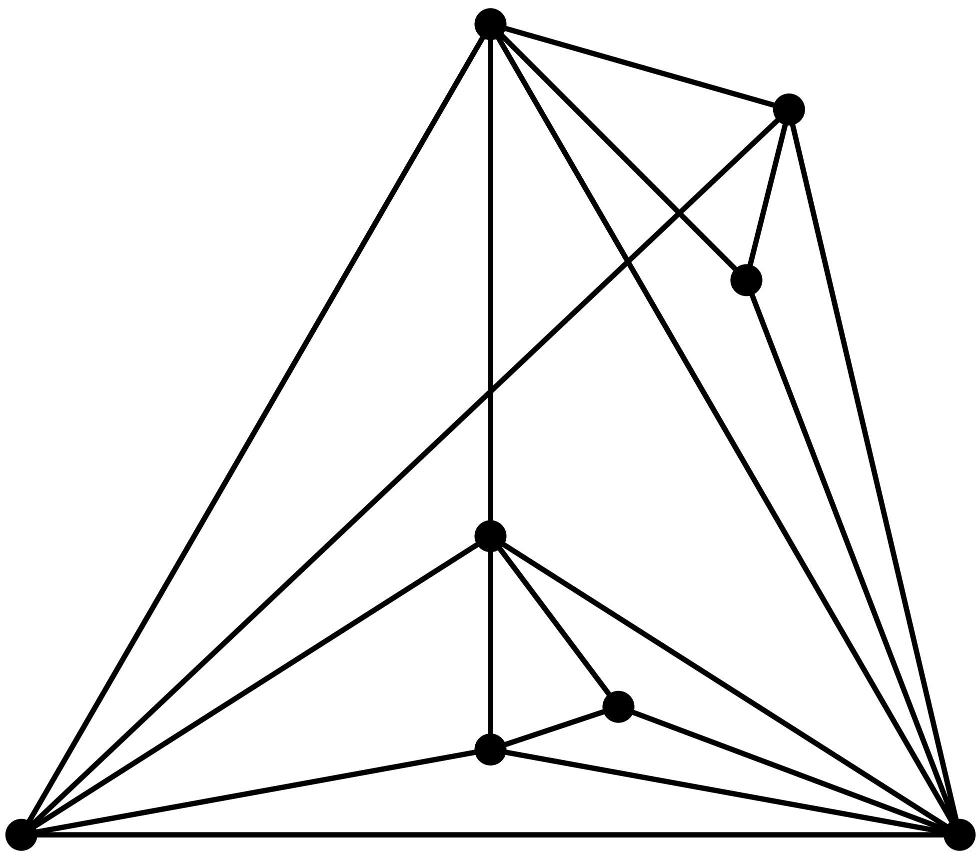
Iteratively add a new vertex connected to the vertices of an existing **triangle**.



3-trees

Beyond trees

Iteratively add a new vertex connected to the vertices of an existing **triangle**.



3-trees

Definition. A k-**tree** is a graph obtained from a $(k+1)$-clique by successively adding a new vertex connected to all vertices of an existing k-clique.

Chordal graphs

Iteratively add a new vertex connected to the vertices of an existing **clique** (complete subgraph).

Chordal graphs

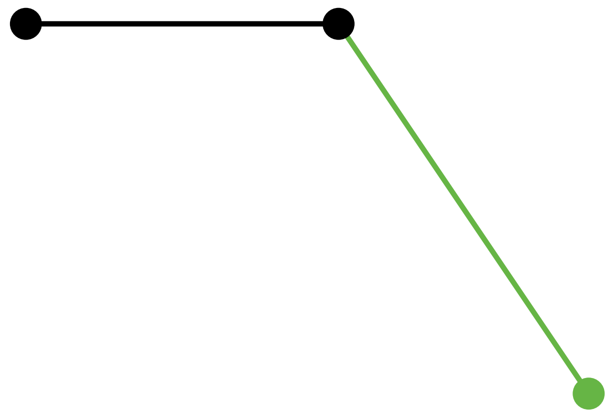
Iteratively add a new vertex connected to the vertices of an existing **clique** (complete subgraph).

Chordal graphs

Iterately add a new vertex connected to the vertices of an existing **clique** (complete subgraph).

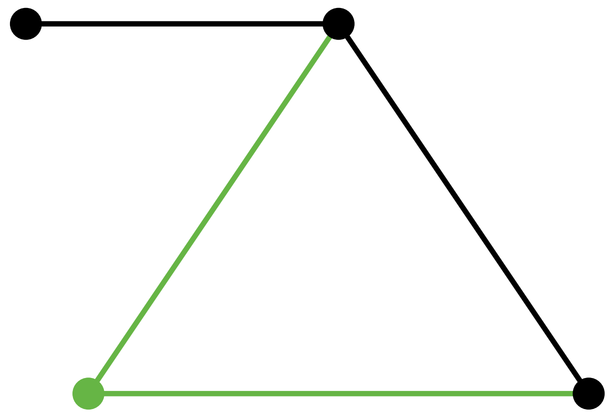
Chordal graphs

Iteratively add a new vertex connected to the vertices of an existing **clique** (complete subgraph).



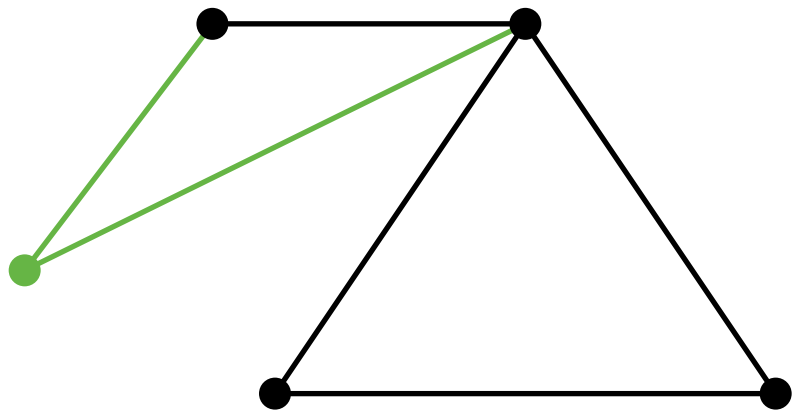
Chordal graphs

Iteratively add a new vertex connected to the vertices of an existing **clique** (complete subgraph).



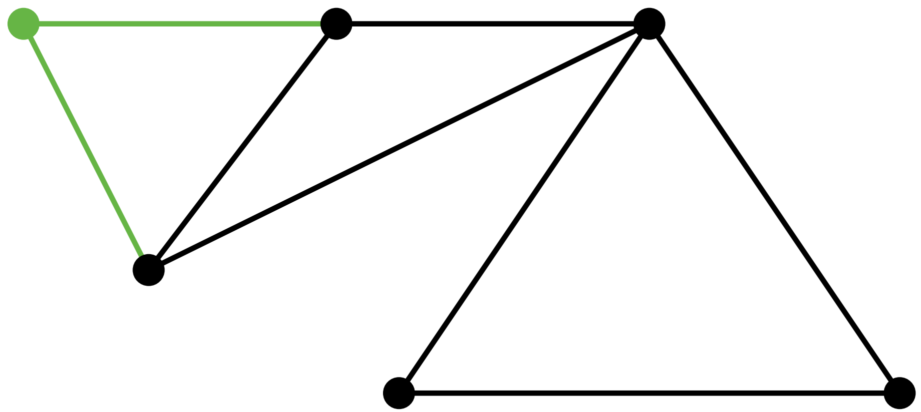
Chordal graphs

Iteratively add a new vertex connected to the vertices of an existing **clique** (complete subgraph).



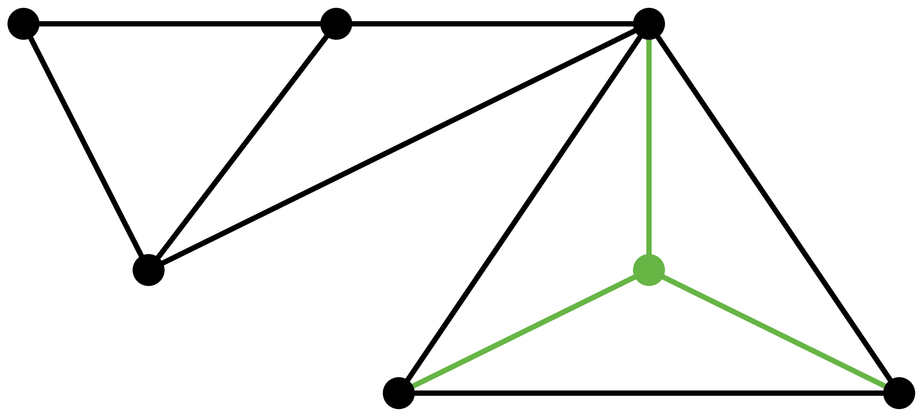
Chordal graphs

Iteratively add a new vertex connected to the vertices of an existing **clique** (complete subgraph).



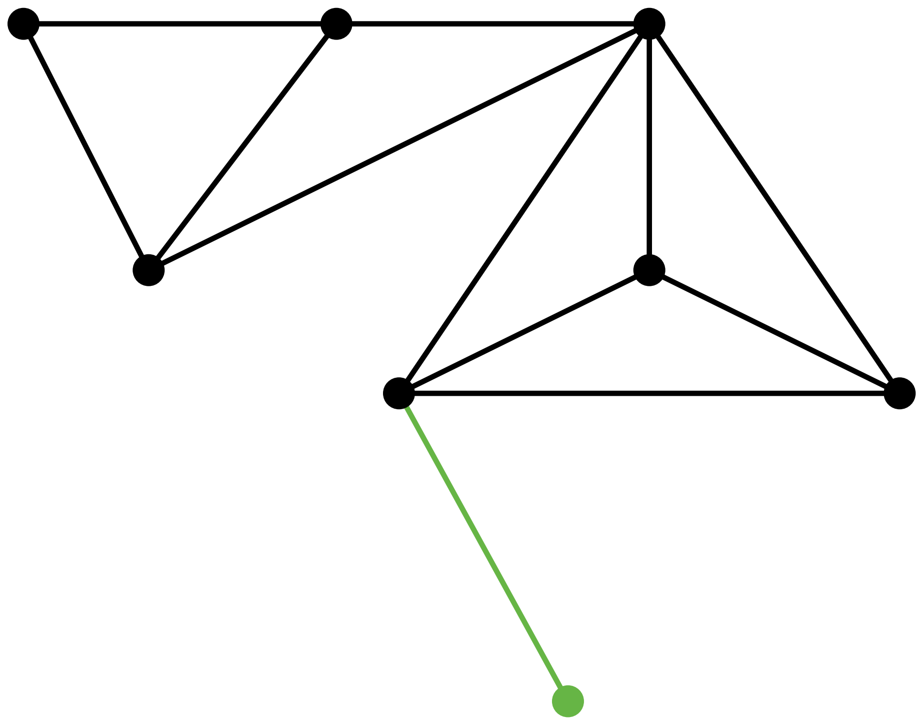
Chordal graphs

Iteratively add a new vertex connected to the vertices of an existing **clique** (complete subgraph).



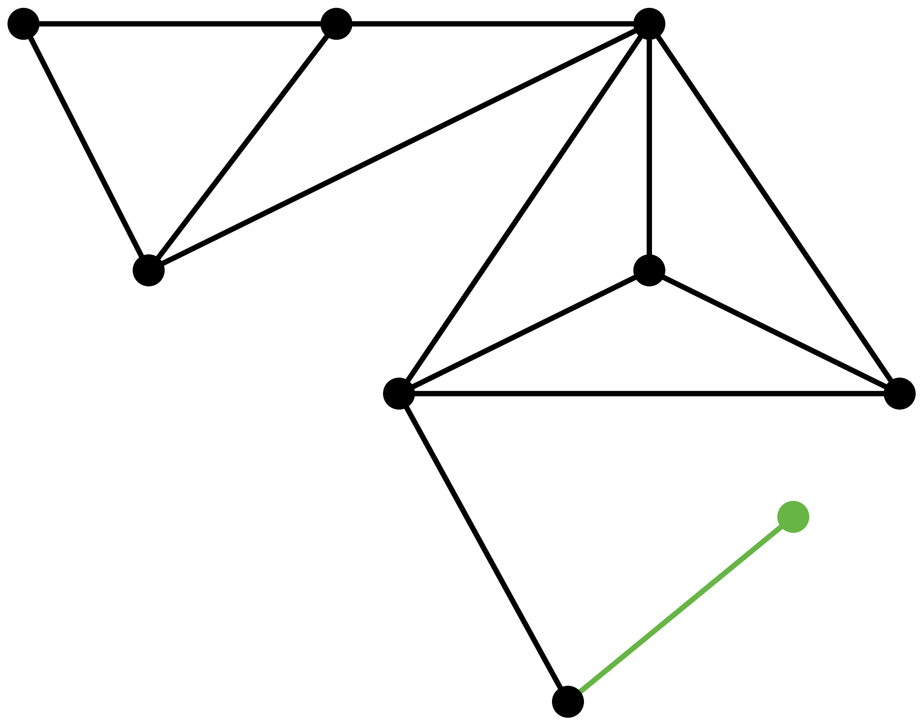
Chordal graphs

Iteratively add a new vertex connected to the vertices of an existing **clique** (complete subgraph).



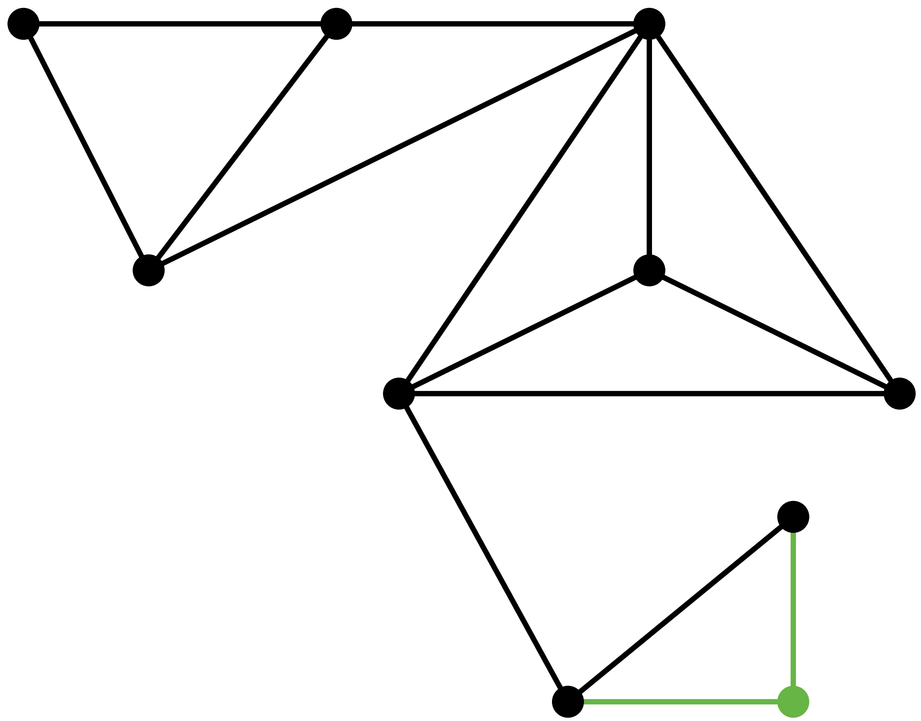
Chordal graphs

Iteratively add a new vertex connected to the vertices of an existing **clique** (complete subgraph).



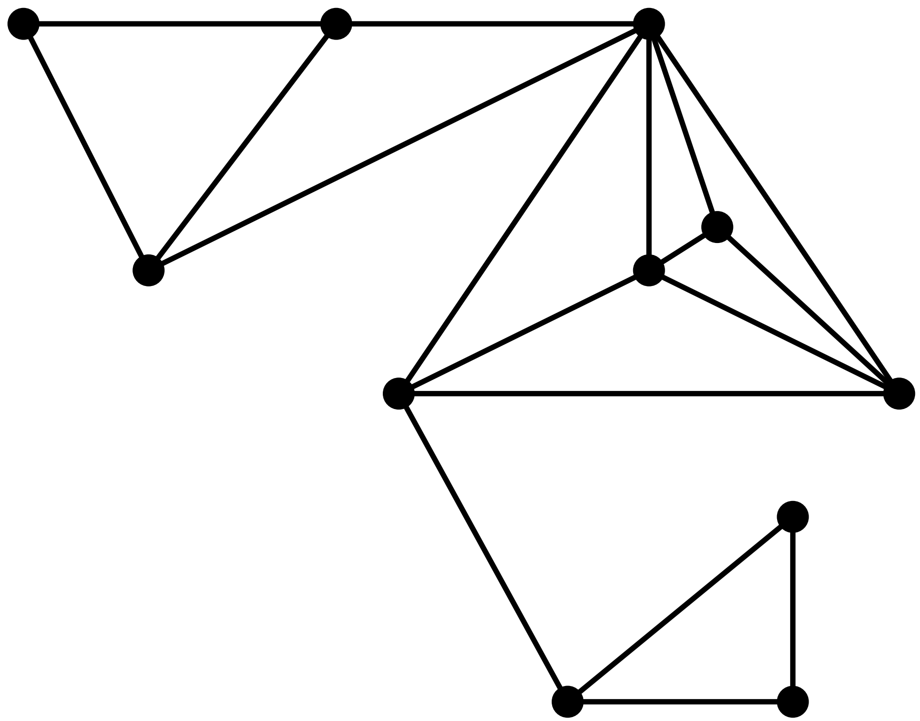
Chordal graphs

Iteratively add a new vertex connected to the vertices of an existing **clique** (complete subgraph).



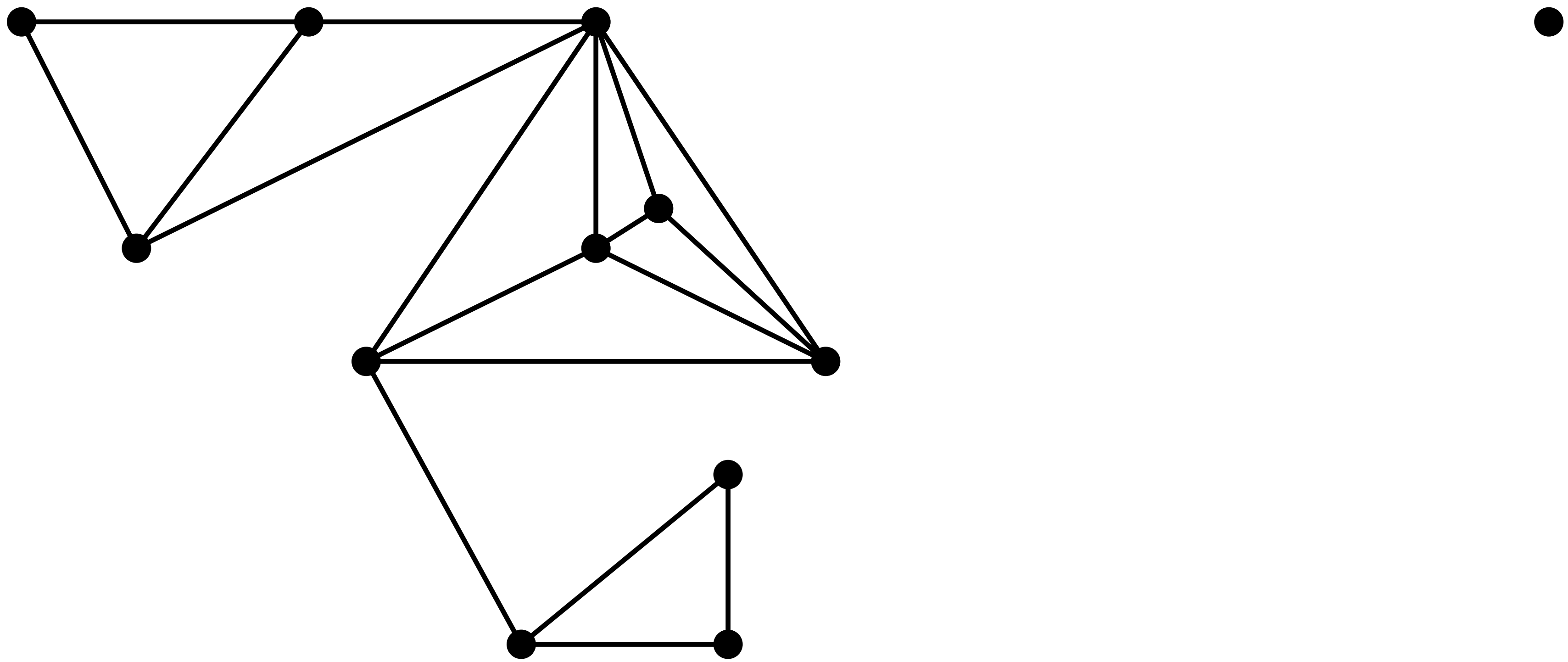
Chordal graphs

Iterately add a new vertex connected to the vertices of an existing **clique** (complete subgraph).



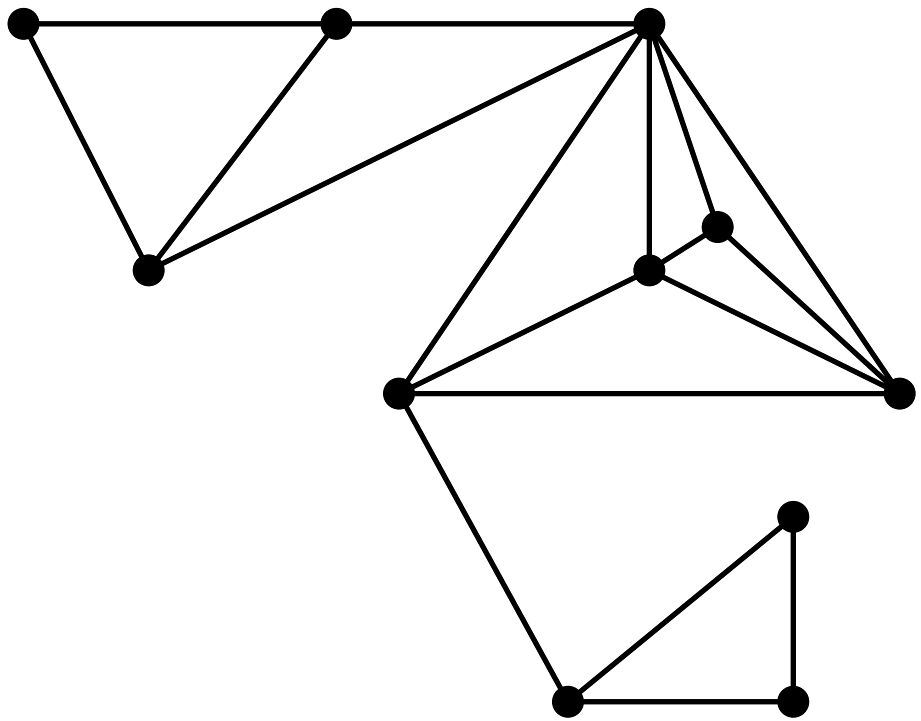
Chordal graphs

Iteratively add a new vertex connected to the vertices of an existing **clique** (complete subgraph).



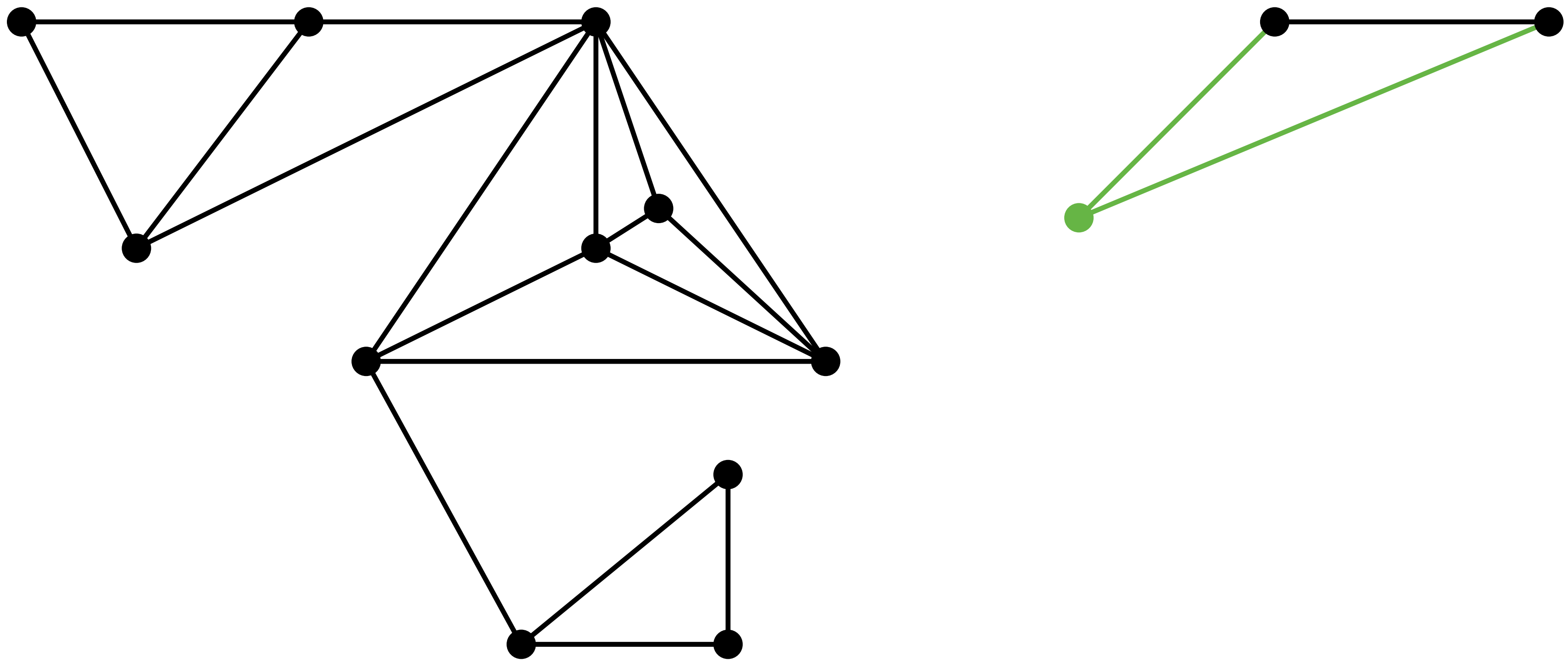
Chordal graphs

Iteratively add a new vertex connected to the vertices of an existing **clique** (complete subgraph).



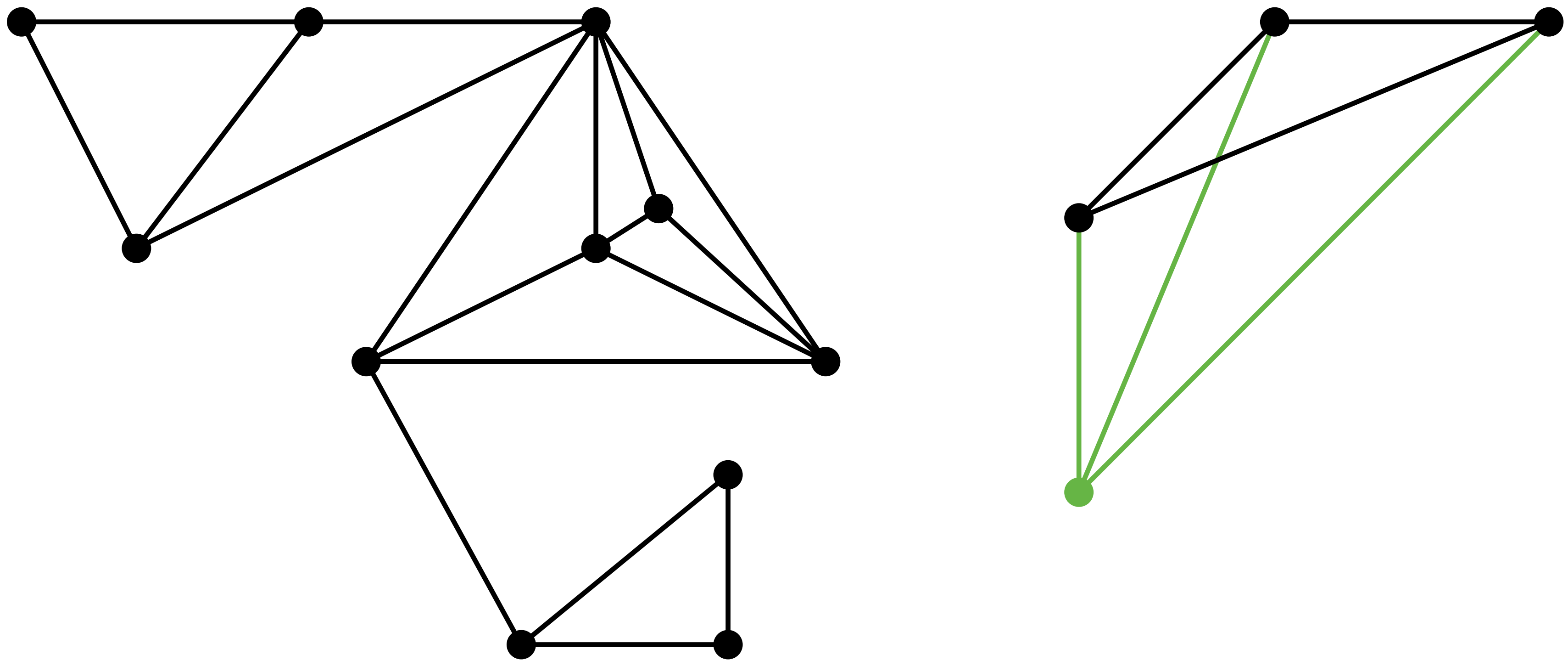
Chordal graphs

Iteratively add a new vertex connected to the vertices of an existing **clique** (complete subgraph).



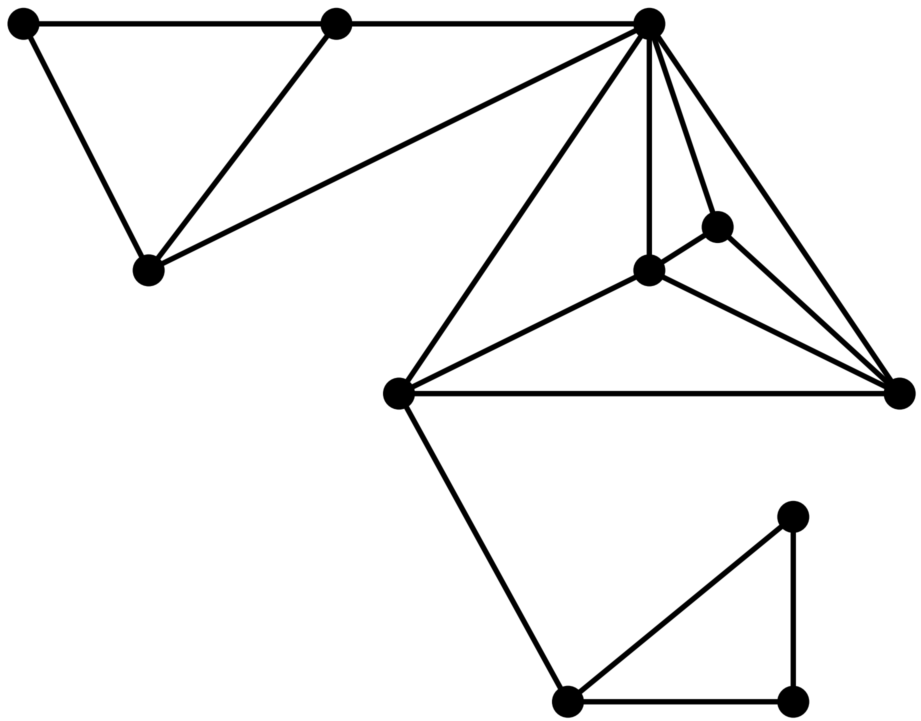
Chordal graphs

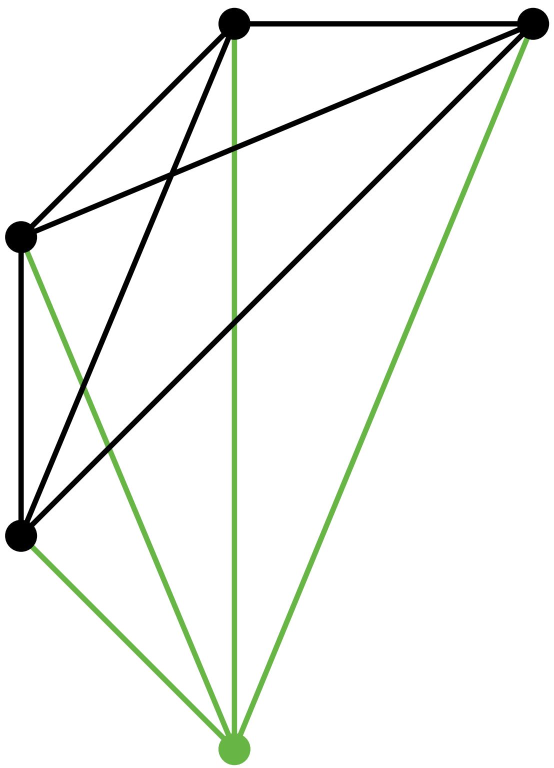
Iteratively add a new vertex connected to the vertices of an existing **clique** (complete subgraph).



Chordal graphs

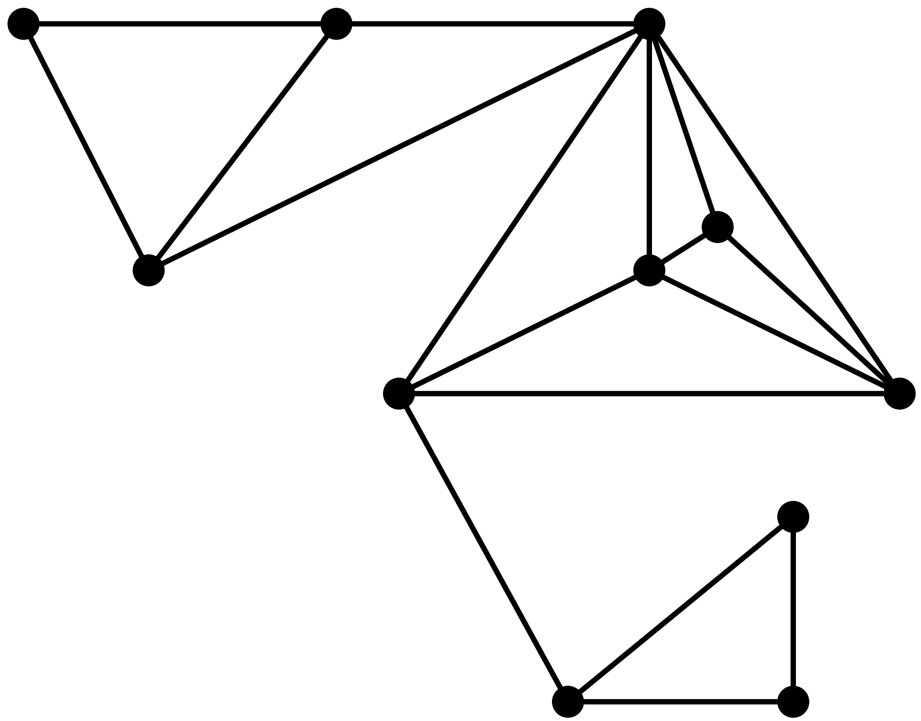
Iteratively add a new vertex connected to the vertices of an existing **clique** (complete subgraph).

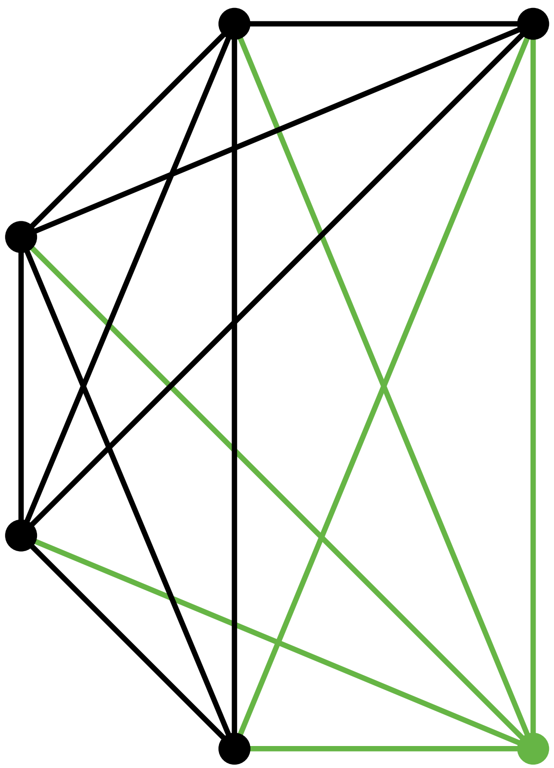




Chordal graphs

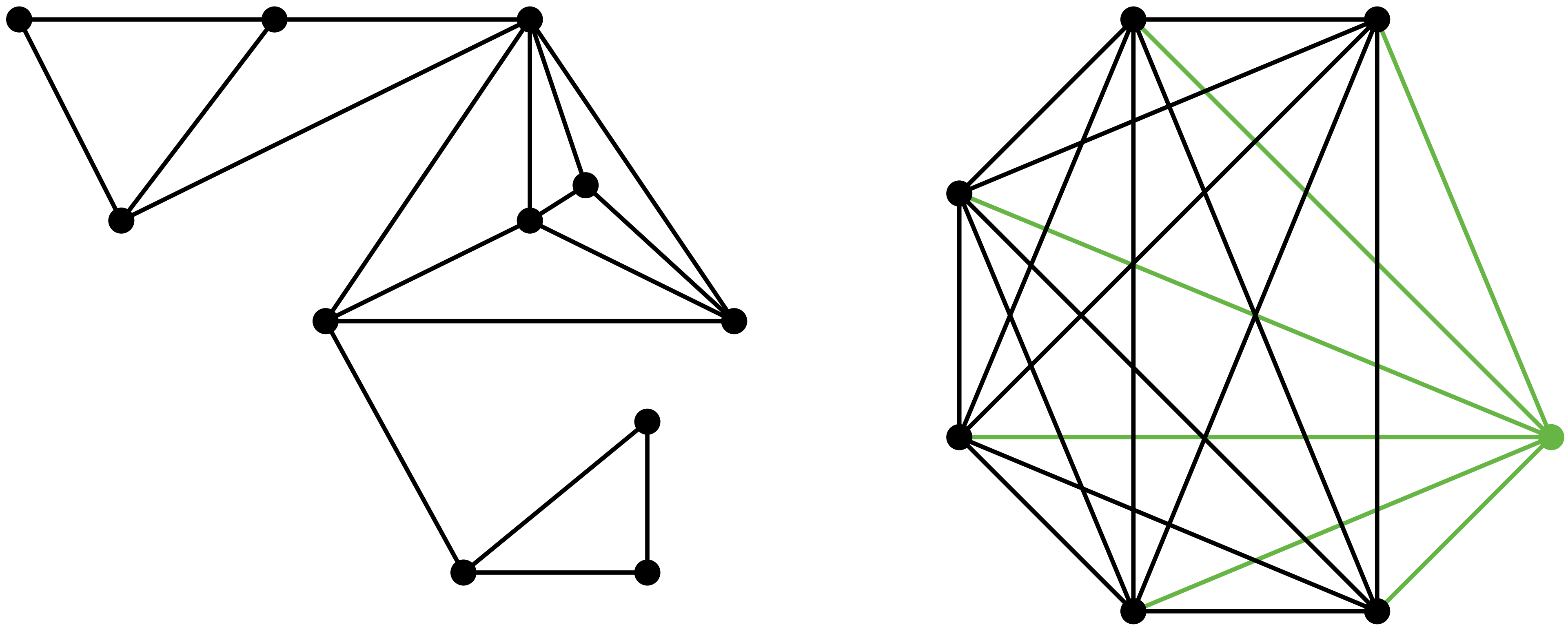
Iteratively add a new vertex connected to the vertices of an existing **clique** (complete subgraph).





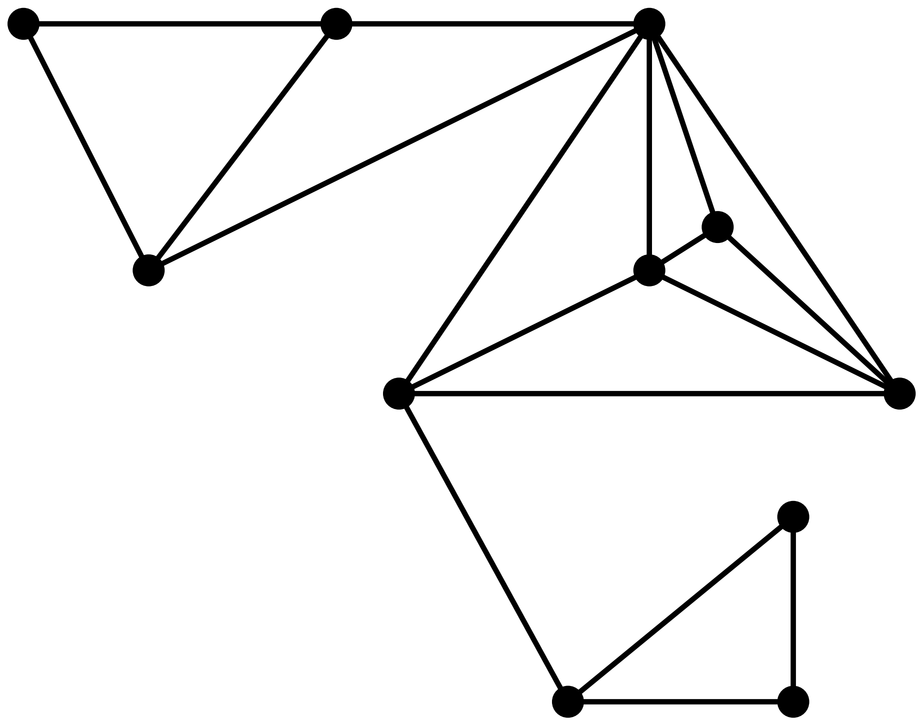
Chordal graphs

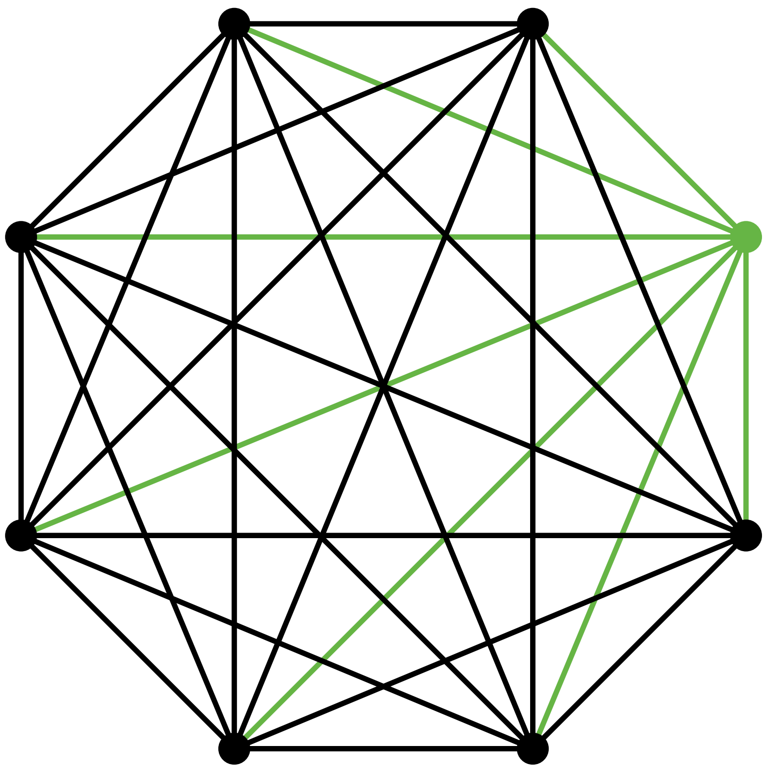
Iterativelyadd a new vertex connected to the vertices of an existing **clique** (complete subgraph).



Chordal graphs

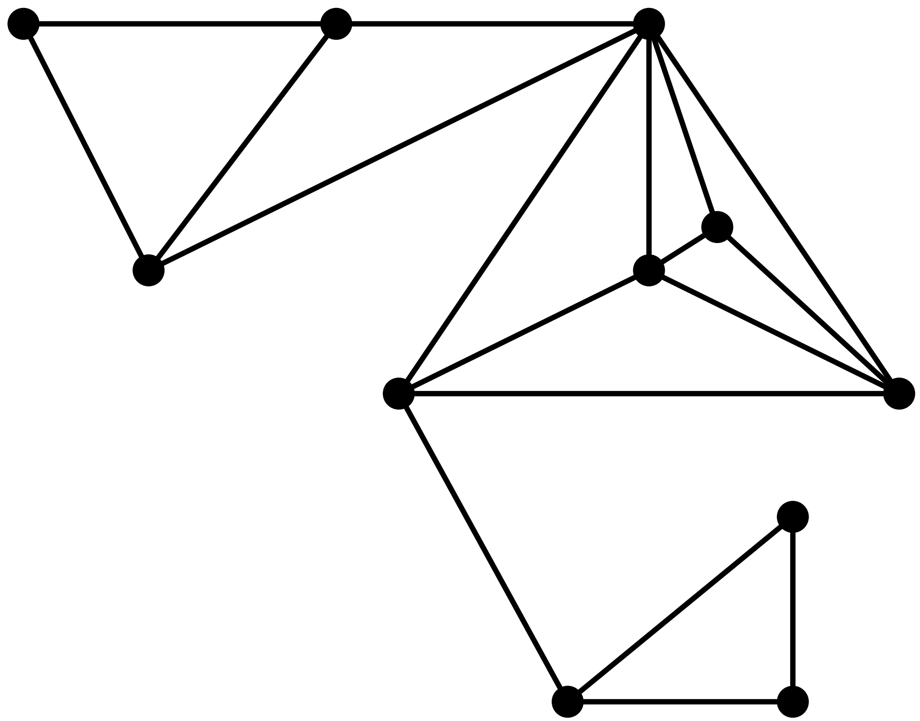
Iterately add a new vertex connected to the vertices of an existing **clique** (complete subgraph).

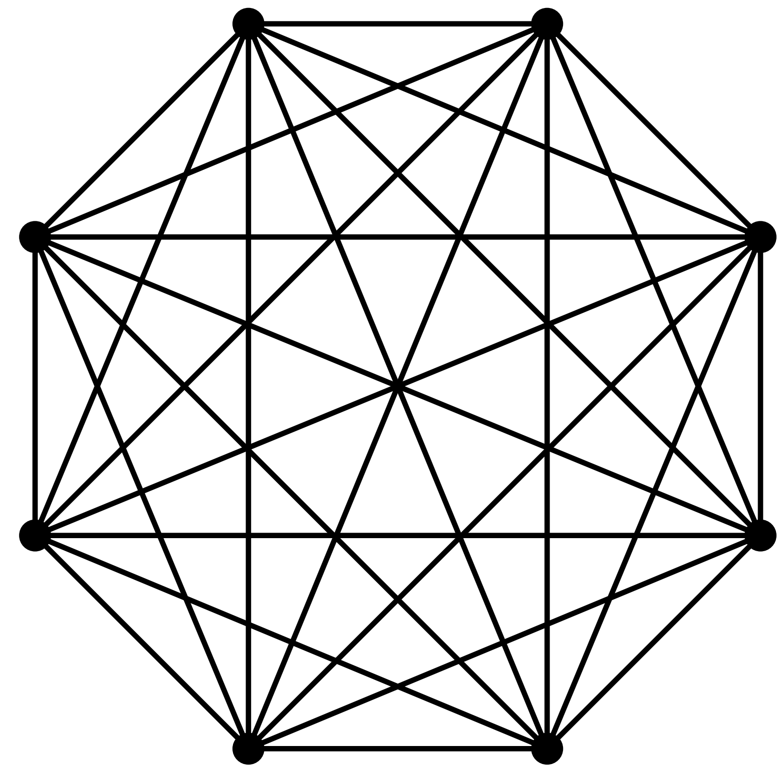




Chordal graphs

Iteratively add a new vertex connected to the vertices of an existing **clique** (complete subgraph).

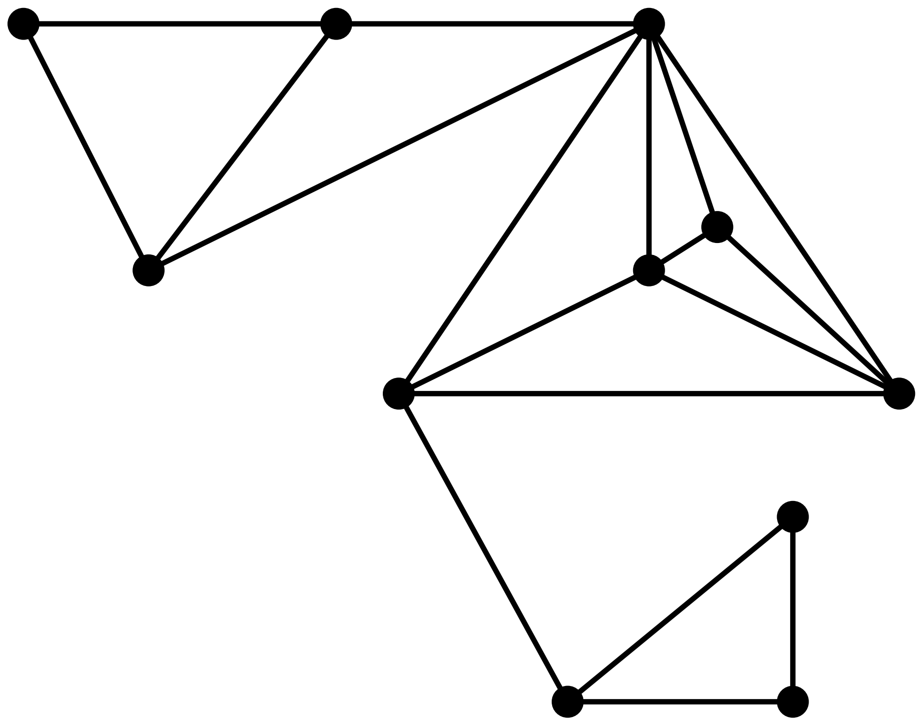


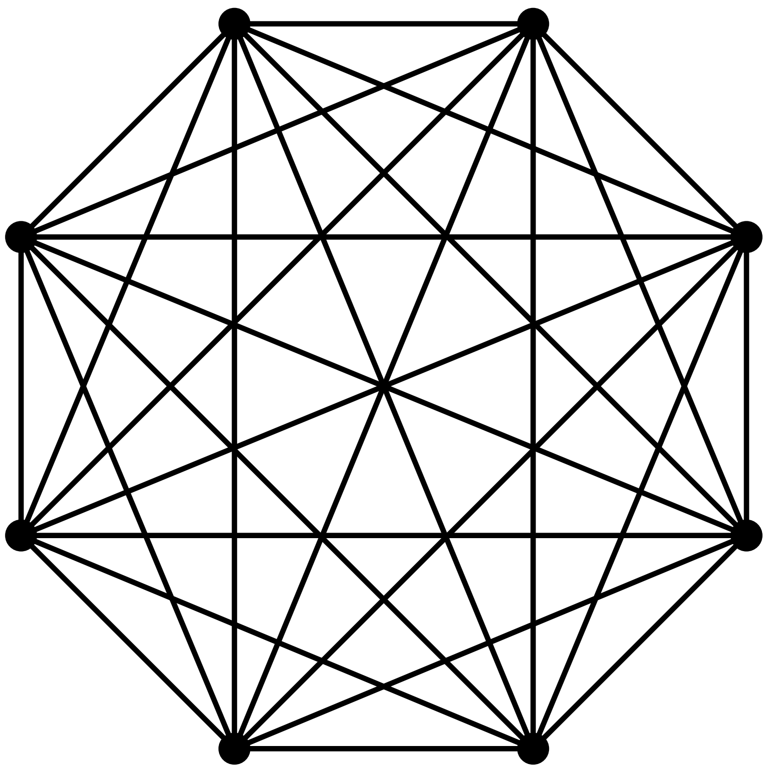


Chordal graphs

Chordal graphs

Iteratively add a new vertex connected to the vertices of an existing **clique** (complete subgraph).





Chordal graphs

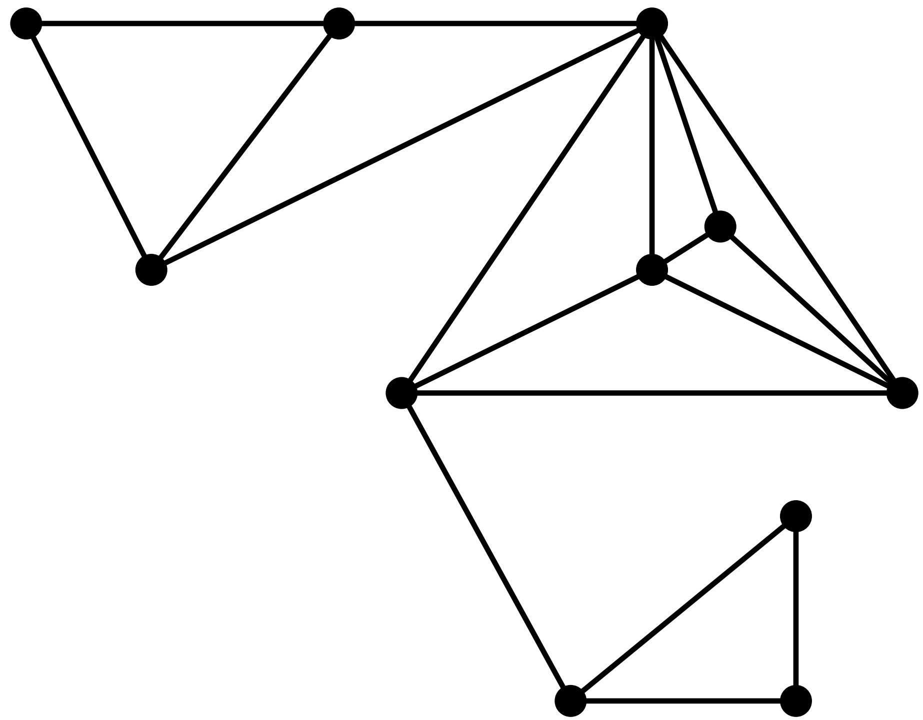
Definition. A graph is **chordal** if it has no induced cycle of lengh greater than 3.

Our class of graphs

Iteratively add a new vertex connected to the vertices of an existing **clique of size at most t**.

Our class of graphs

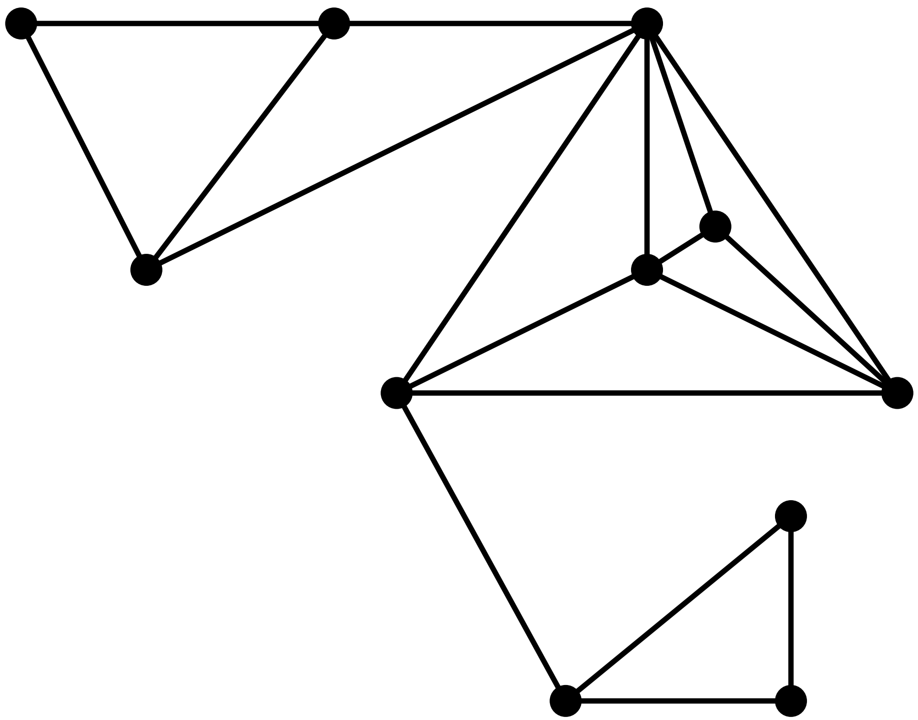
Iteratively add a new vertex connected to the vertices of an existing **clique of size at most t**.



$t = 3$

Our class of graphs

Iteratively add a new vertex connected to the vertices of an existing **clique of size at most t**.

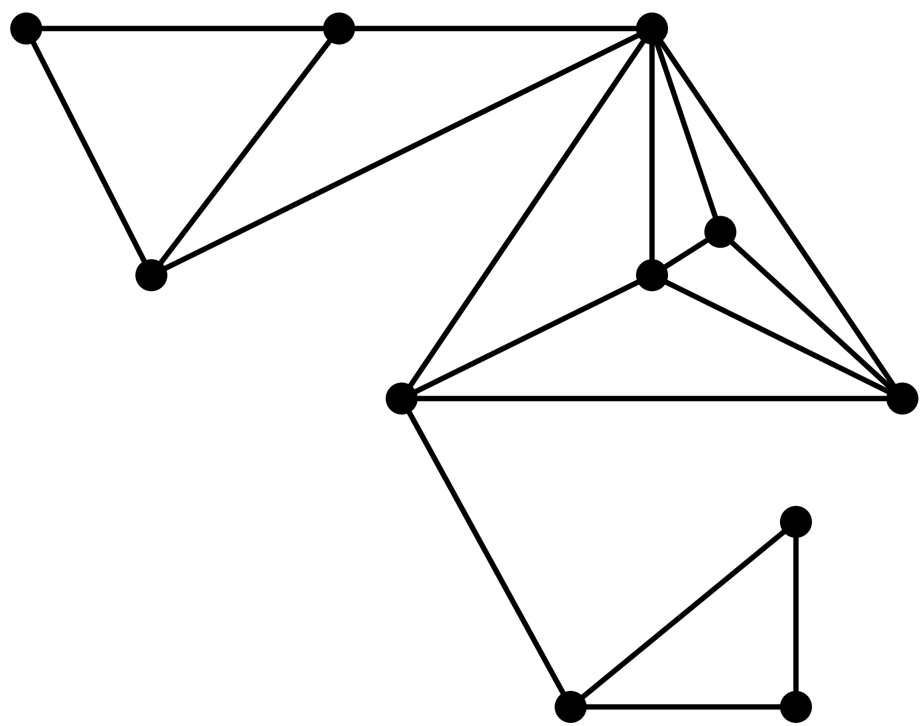


$t = 3$

Chordal graphs with tree-width at most t

Our class of graphs

Iteratively add a new vertex connected to the vertices of an existing **clique of size at most t**.



$t = 3$

Chordal graphs with tree-width at most t

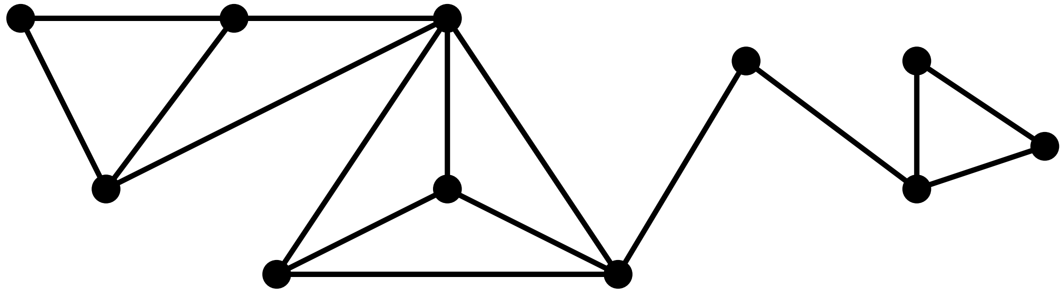
Definition. The **tree-width** of a graph G is the minimum k such that G is the subgraph of a k-tree.

Block decomposition

Definition. The **2-connected components** (or blocks) of a connected graph are its maximal 2-connected subgraphs.

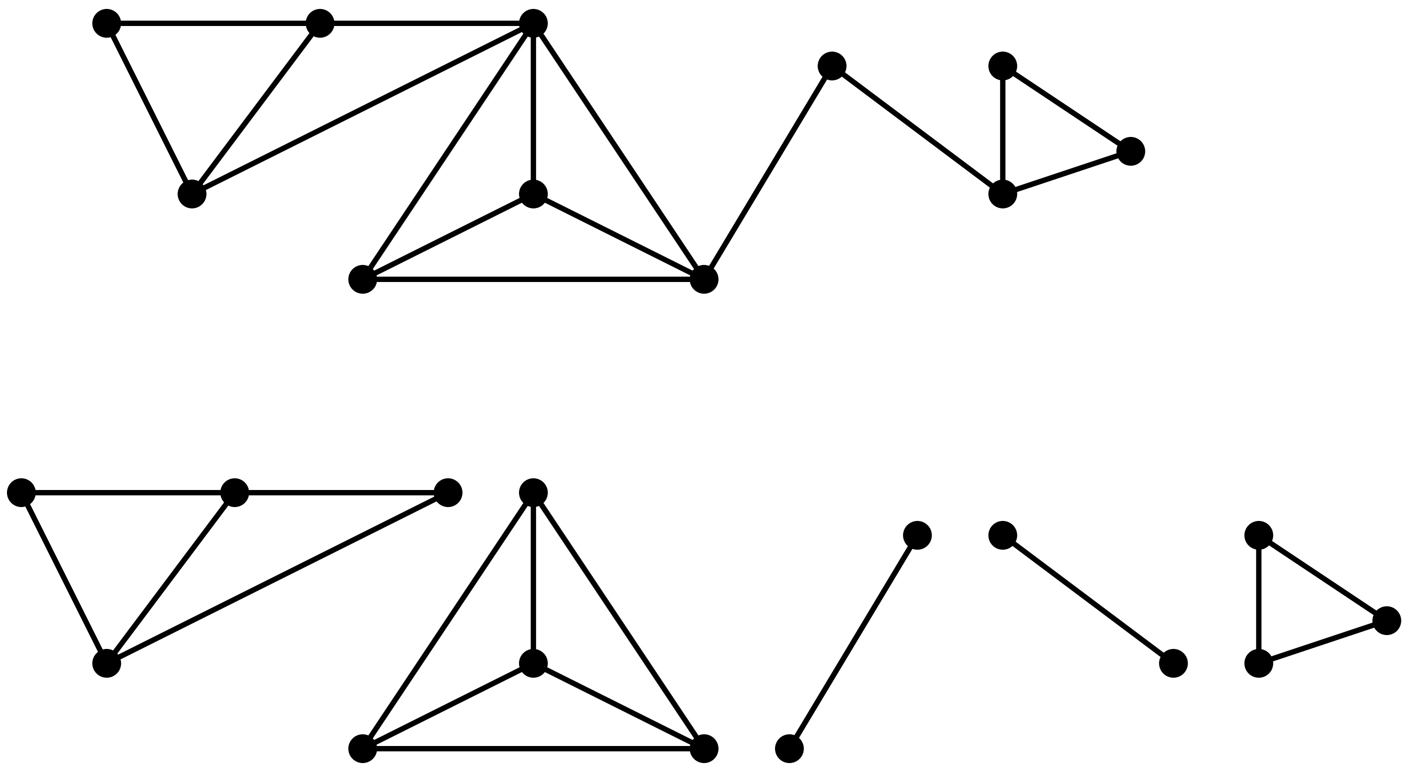
Block decomposition

Definition. The **2-connected components** (or blocks) of a connected graph are its maximal 2-connected subgraphs.



Block decomposition

Definition. The **2-connected components** (or blocks) of a connected graph are its maximal 2-connected subgraphs.



Block decomposition

Let $\mathcal{B} \subset \mathcal{C}$ be the class of the 2-connected members of $\mathcal{G}$. Then,

$$C^\bullet(x) = x\exp(B'(C^\bullet(x))), \quad \text{where } C^\bullet(x) = xC'(x),$$

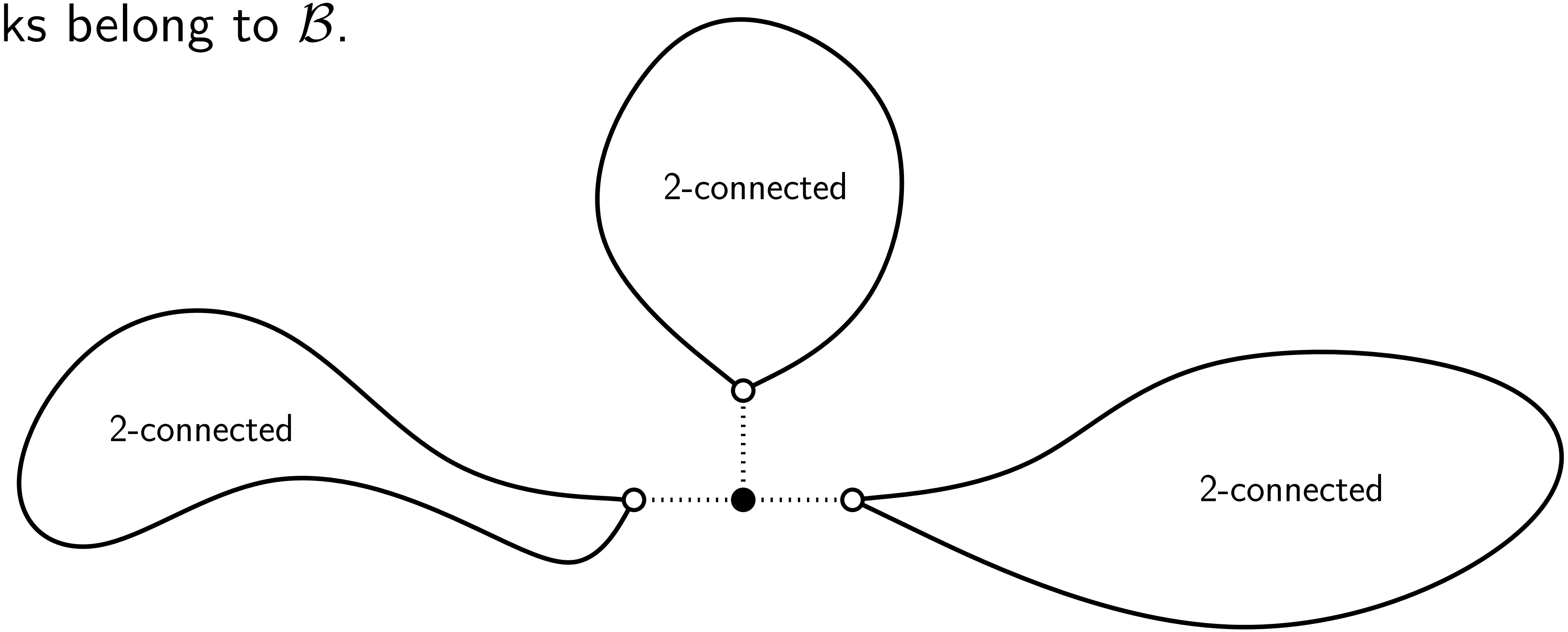
provided that $\mathcal{G}$ is **block-stable**, i.e., that a graph belongs to $\mathcal{C}$ iff its blocks belong to $\mathcal{B}$.

Block decomposition

Let $\mathcal{B} \subset \mathcal{C}$ be the class of the 2-connected members of $\mathcal{G}$. Then,

$$C^\bullet(x) = x \exp(B'(C^\bullet(x))), \quad \text{where } C^\bullet(x) = xC'(x),$$

provided that $\mathcal{G}$ is **block-stable**, i.e., that a graph belongs to $\mathcal{C}$ iff its blocks belong to $\mathcal{B}$.

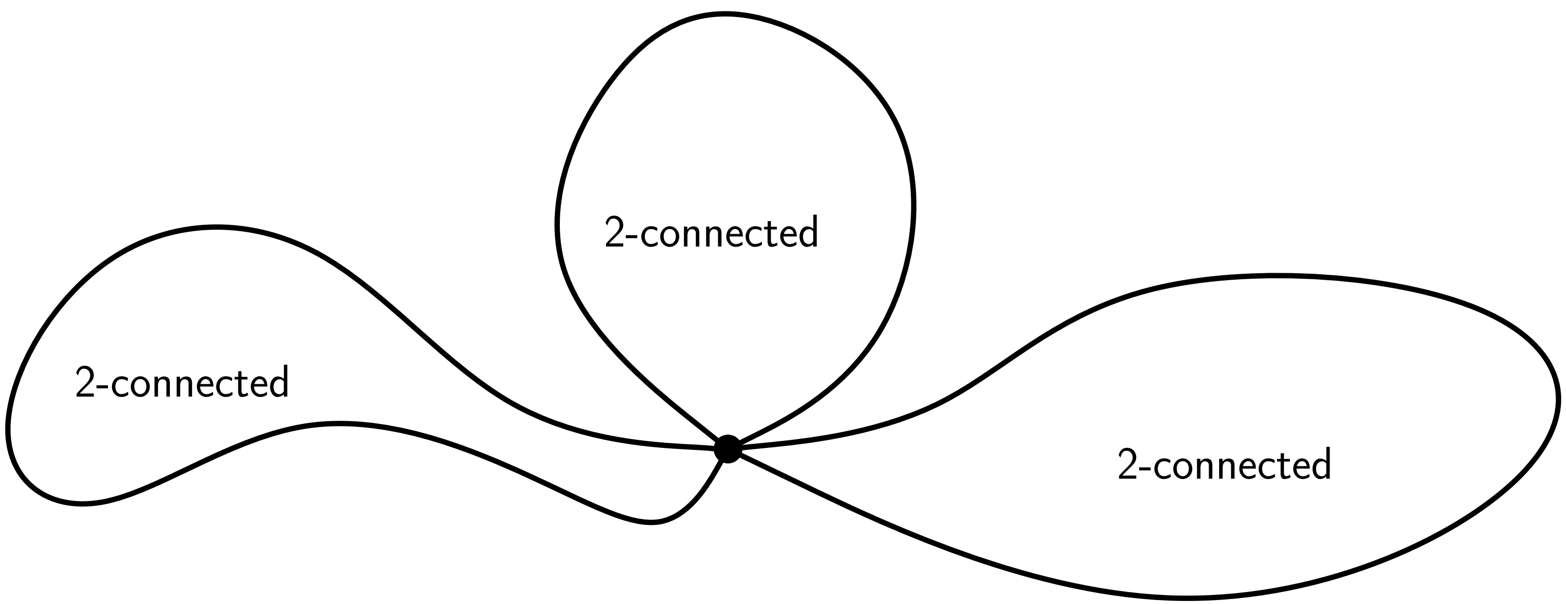


Block decomposition

Let $\mathcal{B} \subset \mathcal{C}$ be the class of the 2-connected members of $\mathcal{G}$. Then,

$$C^\bullet(x) = x\exp(B'(C^\bullet(x))), \quad \text{where } C^\bullet(x) = xC'(x),$$

provided that $\mathcal{G}$ is **block-stable**, i.e., that a graph belongs to $\mathcal{C}$ iff its blocks belong to $\mathcal{B}$.

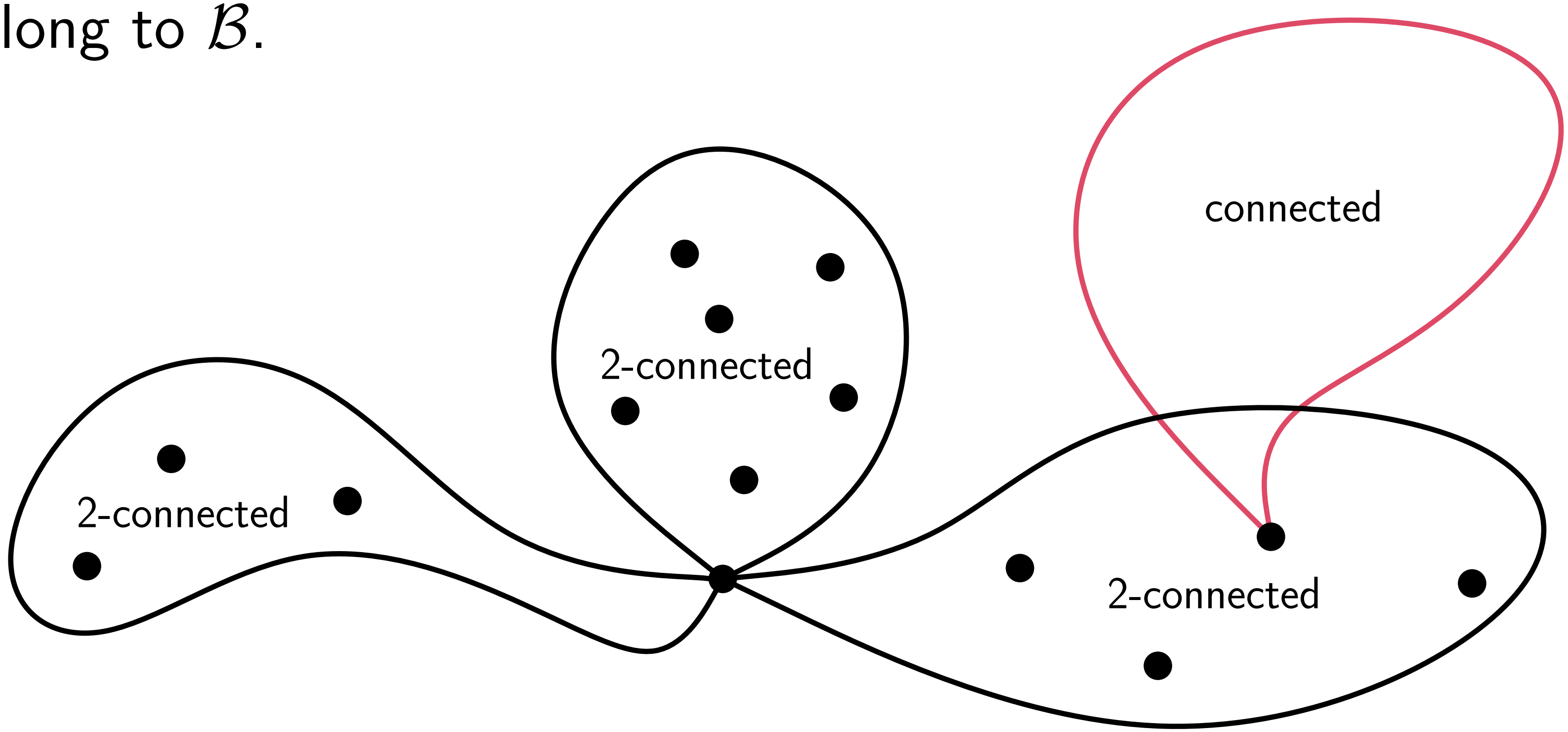


Block decomposition

Let $\mathcal{B} \subset \mathcal{C}$ be the class of the 2-connected members of $\mathcal{G}$. Then,

$$C^\bullet(x) = x\exp(B'(C^\bullet(x))), \quad \text{where } C^\bullet(x) = xC'(x),$$

provided that $\mathcal{G}$ is **block-stable**, i.e., that a graph belongs to $\mathcal{C}$ iff its blocks belong to $\mathcal{B}$.

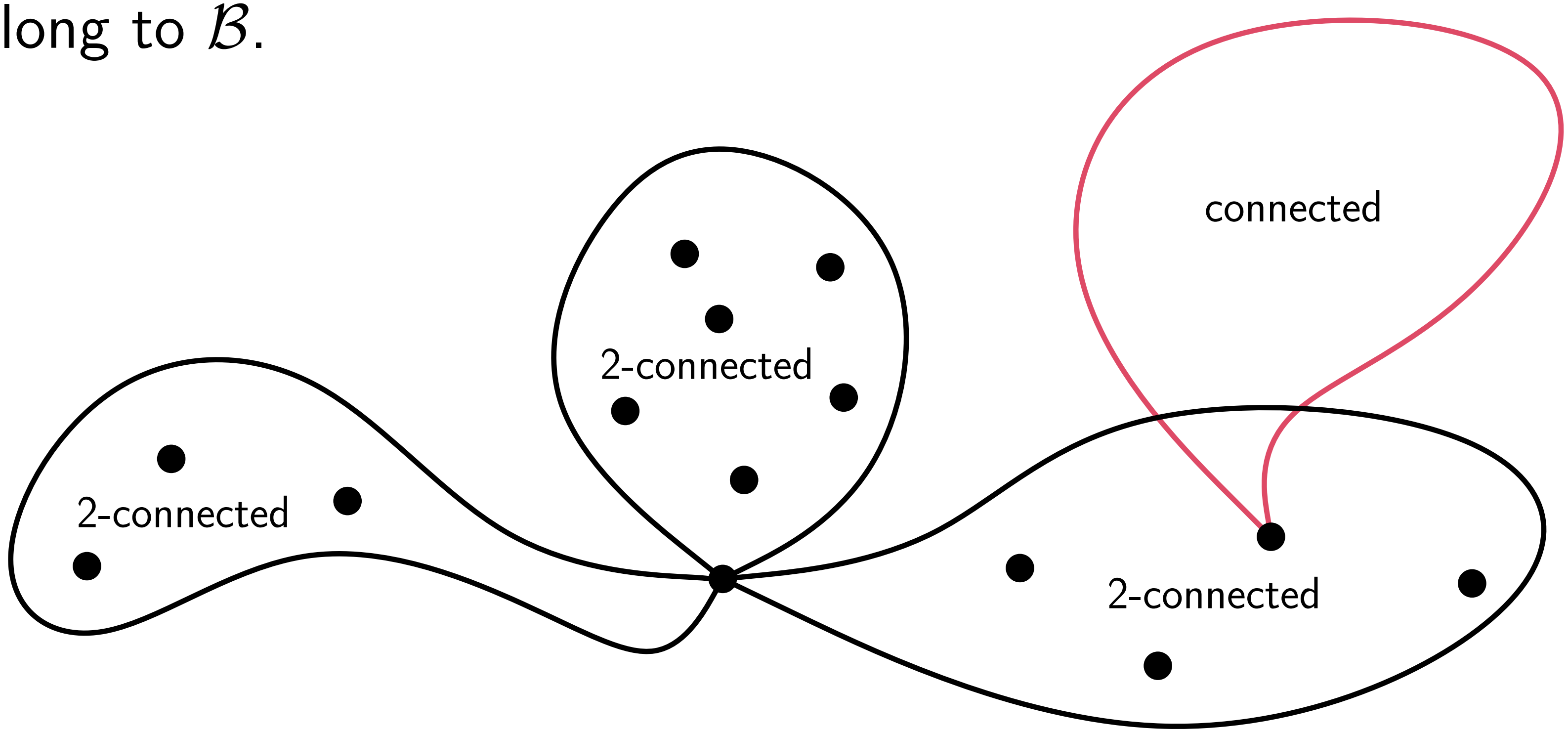


Block decomposition

Let $\mathcal{B} \subset \mathcal{C}$ be the class of the 2-connected members of $\mathcal{G}$. Then,

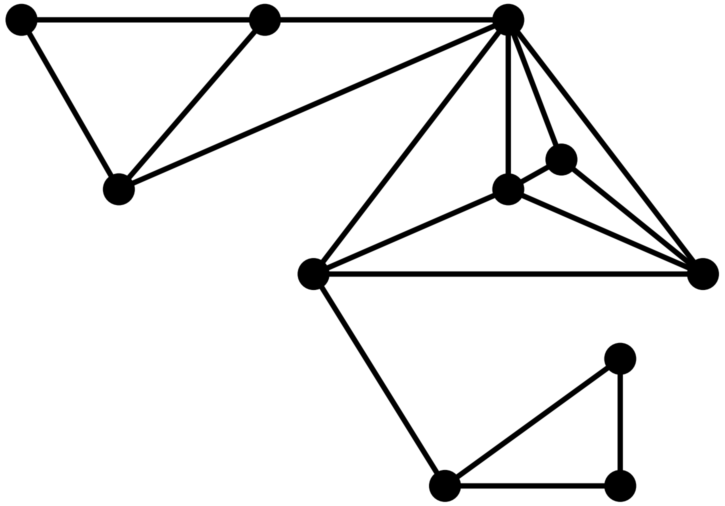
$$C^\bullet(x) = x \exp(B'(C^\bullet(x))), \quad \text{where } C^\bullet(x) = xC'(x),$$

provided that $\mathcal{G}$ is **block-stable**, i.e., that a graph belongs to $\mathcal{C}$ iff its blocks belong to $\mathcal{B}$.



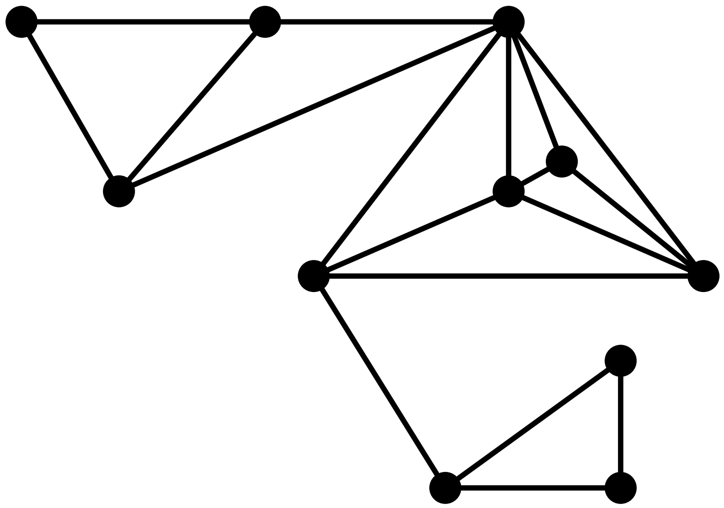
The 3-connected components of a 2-connected graph can also be defined, but the details are a little more involved. However, 4-connected components cannot be defined in general.

Our class of graphs



Chordal graphs with tree-width at most t

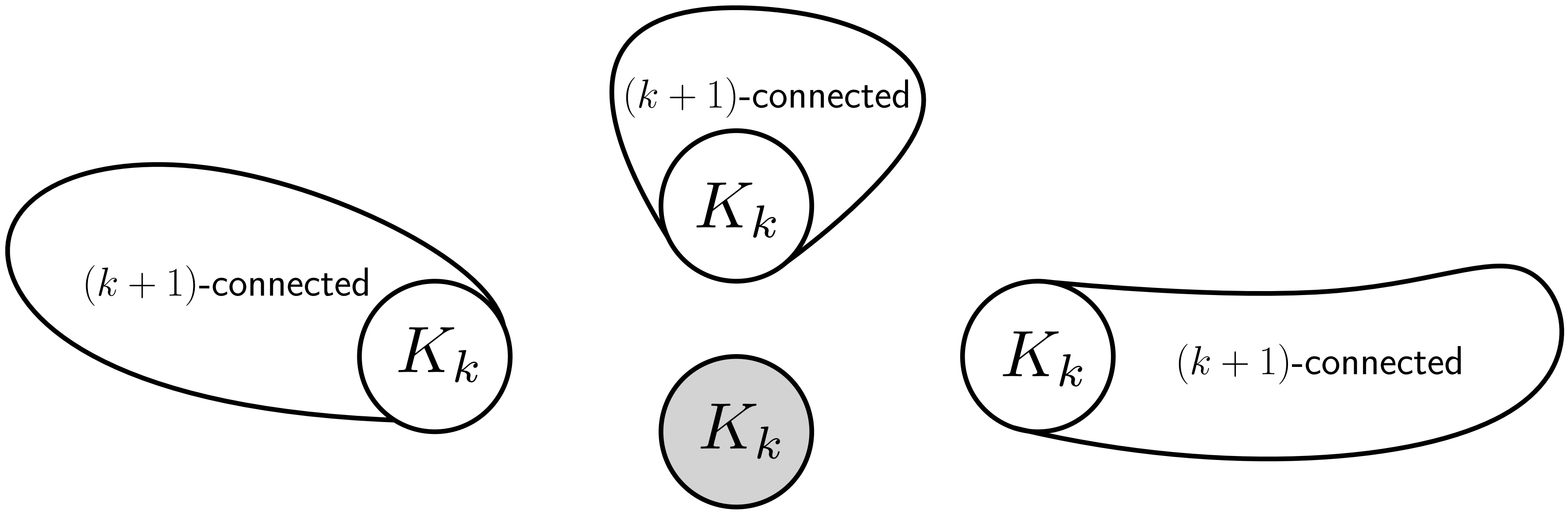
Our class of graphs



Chordal graphs with tree-width at most t

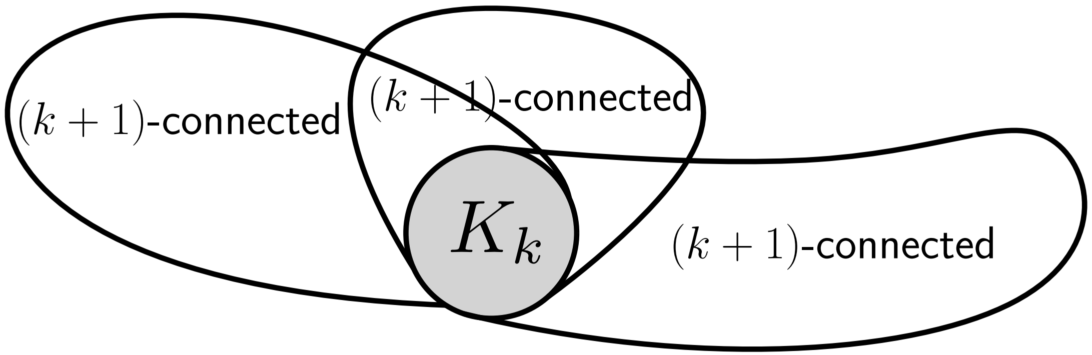
[Wormald (1985)]: they admit a decomposition into k-connected components.

Our class of graphs



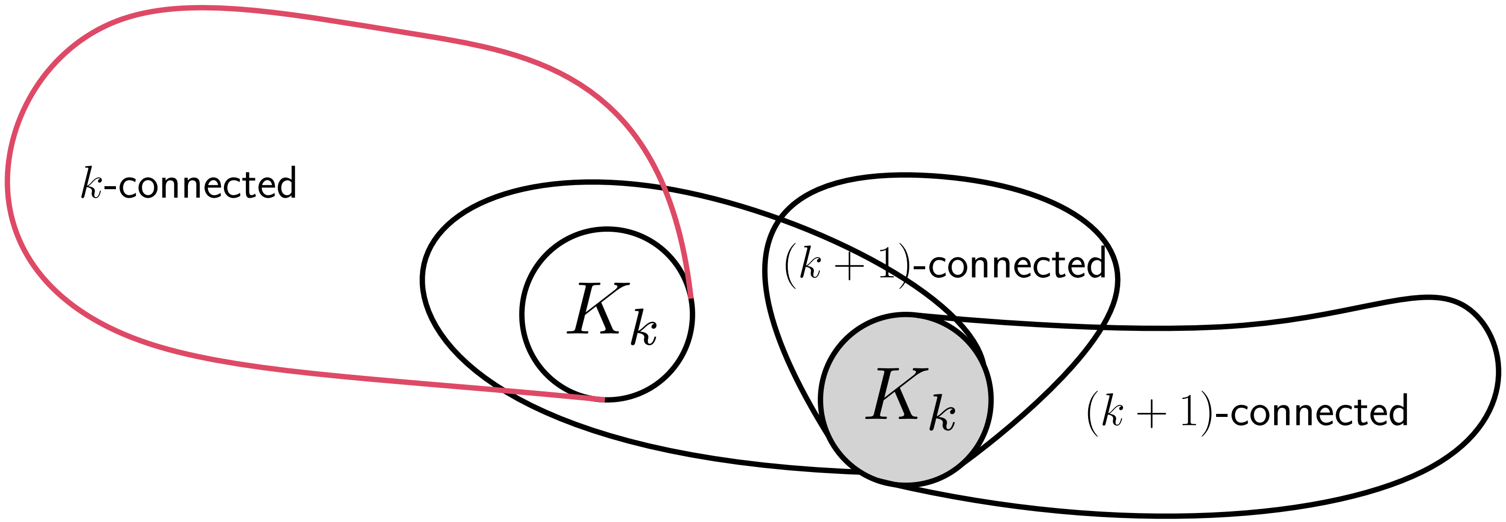
[Wormald (1985)]: they admit a decomposition into k-connected components.

Our class of graphs



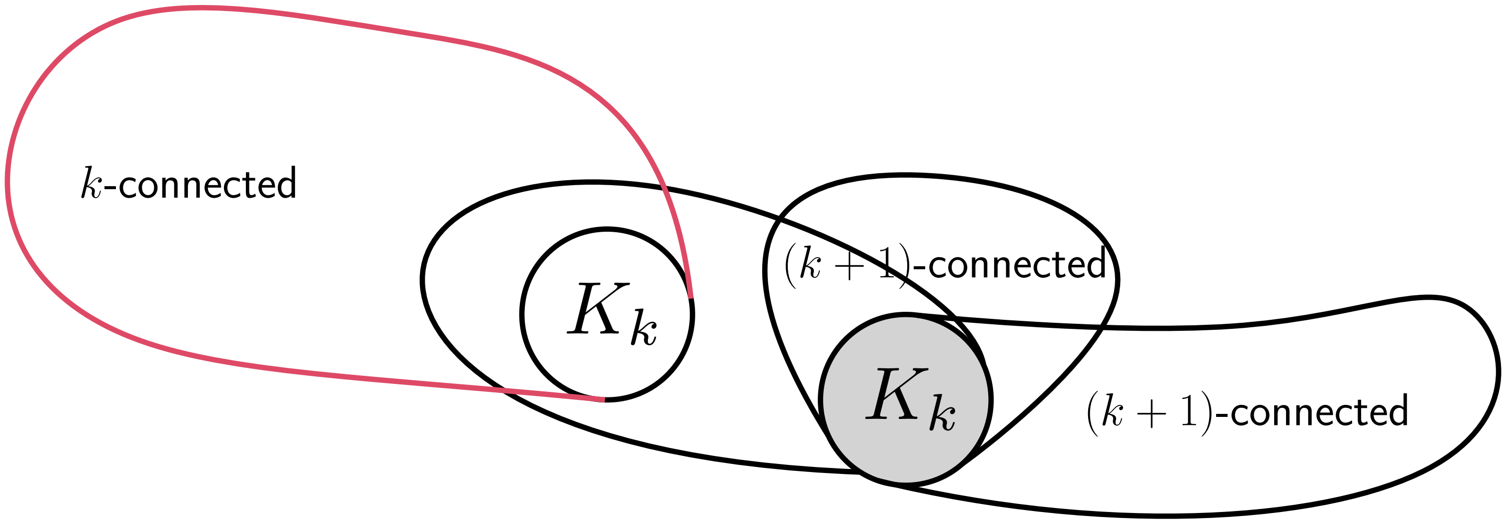
[Wormald (1985)]: they admit a decomposition into k-connected components.

Our class of graphs



[Wormald (1985)]: they admit a decomposition into k-connected components.

Our class of graphs



[Wormald (1985)]: they admit a decomposition into k-connected components.

[C., Drmota, Noy & Requilé (2023)]: assyptotic enumeration of the labeled class.

$$|\mathcal{G}_{t,n}| \sim c_t \cdot n^{-5/2} \cdot \gamma_t^n \cdot n! \qquad \text{as } n \to \infty,$$

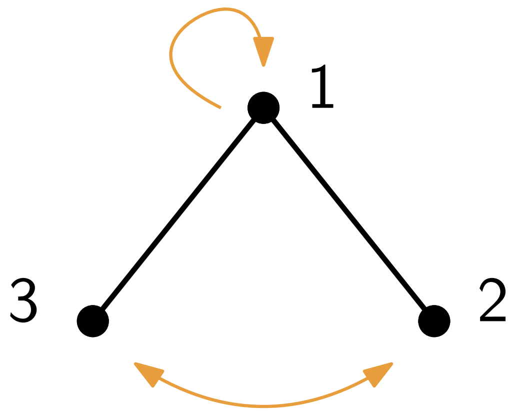
for some $c_t > 0$ and $\gamma_t > 1$

An extension of Pólya theory

We need to take into account **cycles of cliques**, not just vertices.

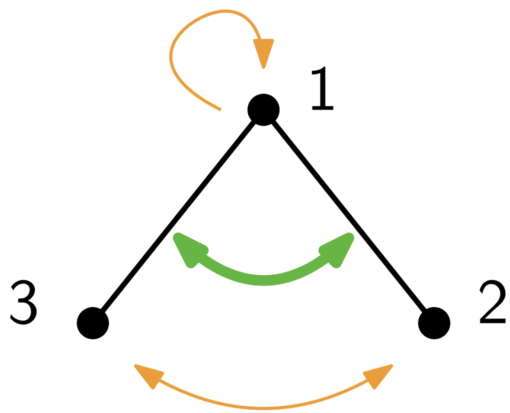
An extension of Pólya theory

We need to take into account **cycles of cliques**, not just vertices.



An extension of Pólya theory

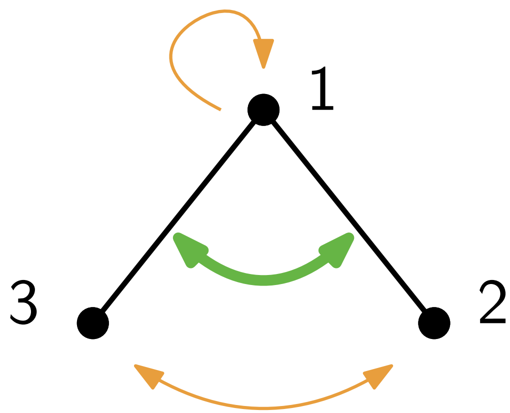
We need to take into account **cycles of cliques**, not just vertices.



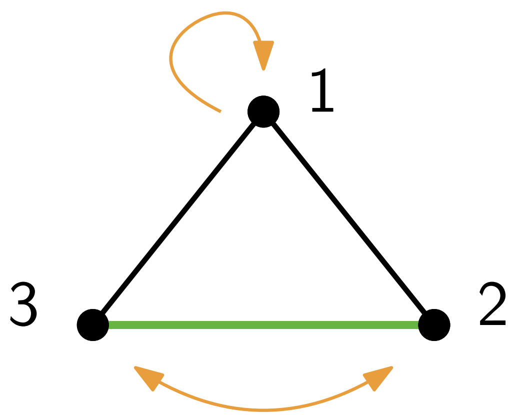
The **edges** are in a cycle of length 2.

An extension of Pólya theory

We need to take into account **cycles of cliques**, not just vertices.



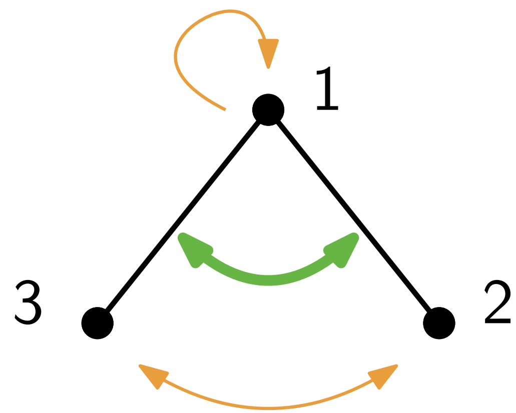
The **edges** are in a cycle of length 2.



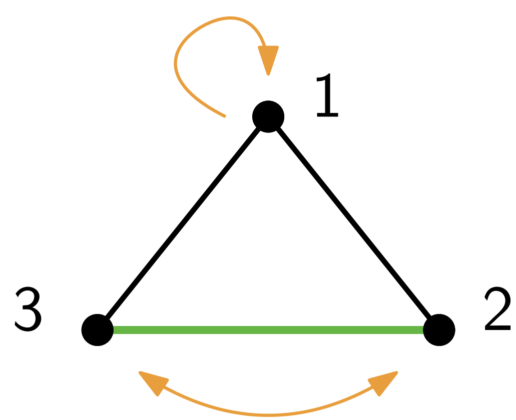
The new edge is in a cycle of length 1 but different type: it flips itself.

An extension of Pólya theory

We need to take into account **cycles of cliques**, not just vertices.



The **edges** are in a cycle of length 2.



The new edge is in a cycle of length 1 but different type: it flips itself.

What we do:

- Refinement of cycle index sums to encode cycles of cliques.
- Extend cycle-pointing to cycles of cliques.

Results

- [Harary & Palmer (1968)], [Fowler, Gessel, Labelle & Leroux (2002)]: asymptotic enumeration of unlabeled (rooted and unrooted) **2-trees**

Results

- [Harary & Palmer (1968)], [Fowler, Gessel, Labelle & Leroux (2002)]: asymptotic enumeration of unlabeled (rooted and unrooted) **2-trees**
- [Gainer-Dewar (2012)], [Gainer-Dewar & Gessel (2014)]: system of equations to compute the OGF of unlabeled k-**trees** with n vertices.

Results

- [Harary & Palmer (1968)], [Fowler, Gessel, Labelle & Leroux (2002)]: asymptotic enumeration of unlabeled (rooted and unrooted) **2-trees**

- [Gainer-Dewar (2012)], [Gainer-Dewar & Gessel (2014)]: system of equations to compute the OGF of unlabeled k-**trees** with n vertices.

- [Drmota & Yu Jin (2014)]: asymptotic enumeration of unlabeled k-**trees**.

Results

- [Harary & Palmer (1968)], [Fowler, Gessel, Labelle & Leroux (2002)]: asymptotic enumeration of unlabeled (rooted and unrooted) **2-trees**

- [Gainer-Dewar (2012)], [Gainer-Dewar & Gessel (2014)]: system of equations to compute the OGF of unlabeled k-**trees** with n vertices.

- [Drmota & Yu Jin (2014)]: asymptotic enumeration of unlabeled k-**trees**.

This talk: generalisation of previous results.

- [C. & Requilé (2024+)]: system of equations to compute the OGF of unlabeled **chordal graphs with tree-width** $\leq t$ and also unlabeled **chordal planar graphs**.

Results

- [Harary & Palmer (1968)], [Fowler, Gessel, Labelle & Leroux (2002)]: asymptotic enumeration of unlabeled (rooted and unrooted) **2-trees**

- [Gainer-Dewar (2012)], [Gainer-Dewar & Gessel (2014)]: system of equations to compute the OGF of unlabeled k-**trees** with n vertices.

- [Drmota & Yu Jin (2014)]: asymptotic enumeration of unlabeled k-**trees**.

This talk: generalisation of previous results.

- [C. & Requilé (2024+)]: system of equations to compute the OGF of unlabeled **chordal graphs with tree-width** $\leq t$ and also unlabeled **chordal planar graphs**.

Future:

- [C.,Drmota & Requilé (soon?)]: asymptotic enumeration of unlabeled chordal graphs with bounded tree-width and unlabeled chordal planar graphs.

The system

$$\begin{cases} X^{\lambda}_{\mathcal{G}^{(k)}_{t,k+1}} = \frac{k!}{\alpha(\lambda)\kappa(\lambda)} \frac{\partial}{\partial s_{\lambda,1}} X_{\mathcal{G}_{t,k+1}}, \\ X^{\lambda}_{\mathcal{G}^{(k)}_{t,k}} = Z_{\mathrm{SET}}(s_j \to (X^{\lambda^j}_{\mathcal{G}^{(k)}_{t,k+1} \circ_k \mathcal{G}^{(k)}_{t,k}})^{[j]})_{j\geq 1}, \\ X^{\lambda}_{\mathcal{G}^{(k)}_{t,k+1} \circ_k \mathcal{G}^{(k)}_{t,k}} = X^{\lambda}_{\mathcal{G}^{(k)}_{t,k+1}} (s_{\mu,j} \to (X^{\mu}_{\mathcal{G}^{(k)}_{t,k}})^{[j]})_{\mu\vdash k, j\geq 1}, \\ X_{\mathcal{G}^{\bullet_k}_{t,k}} = \sum_{\mu\vdash k} \frac{\alpha(\mu)\kappa(\mu)}{k!} t_{\mu,1} X^{\mu}_{\mathcal{G}^{(k)}_{t,k}} + X_{(\mathcal{G}_{t,k})^{\bullet_k}_{\geq 2}}, \\ X_{(\mathcal{G}_{t,k})^{\bullet_k}_{\geq 2}} = X_{(\mathcal{G}_{t,k+1})^{\bullet_k}_{\geq 2}} (s_{\mu,j} \to (X^{\mu}_{\mathcal{G}^{(k)}_{t,k}})^{[j]}, t_{\mu,j} \to (X^{\mu}_{(\mathcal{G}^{(k)}_{t,k})^{\bullet_k}})^{[j]})_{\mu\vdash k, j\geq 1} \\ \qquad + \sum_{\mu\vdash k} \frac{\alpha(\mu)\kappa(\mu)}{k!} s_{\mu,1} Z_{\mathrm{SET}^{\bullet}_{\geq 2}}(s_j \to (X^{\mu^j}_{\mathcal{G}^{(k)}_{t,k+1} \circ_k \mathcal{G}^{(k)}_{t,k}})^{[j]}, \\ \qquad\qquad t_j \to (X^{\mu^j}_{(\mathcal{G}^{(k)}_{t,k+1} \circ_k \mathcal{G}^{(k)}_{t,k})^{\bullet_k}})^{[j]})_{j\geq 1}, \\ X_{\mathcal{G}_{t,k}} = \sum_{\lambda\vdash k} \sum_{1\leq j\leq \binom{t+1}{k}} \int \frac{1}{j t_{\lambda,j}} X_{\mathcal{G}^{\bullet_k}_{t,k}} (S(\lambda, j) \to 0, T(\lambda, j) \to 0) ds_{\lambda,j} \end{cases}$$