Chopping stuff up…
To decide things fast!

Benjamin Merlin Bumpus - University of Florida 🐊

Based on:

1. arXiv:2302.05575 (w/ Ernst Althaus, James Fairbanks & Daniel Rosiak)
2. arXiv:2207.06091v2 (w/ Jade Master & Zoltan Kocsis)
3. blog posts at bmbumpus.com

This talk

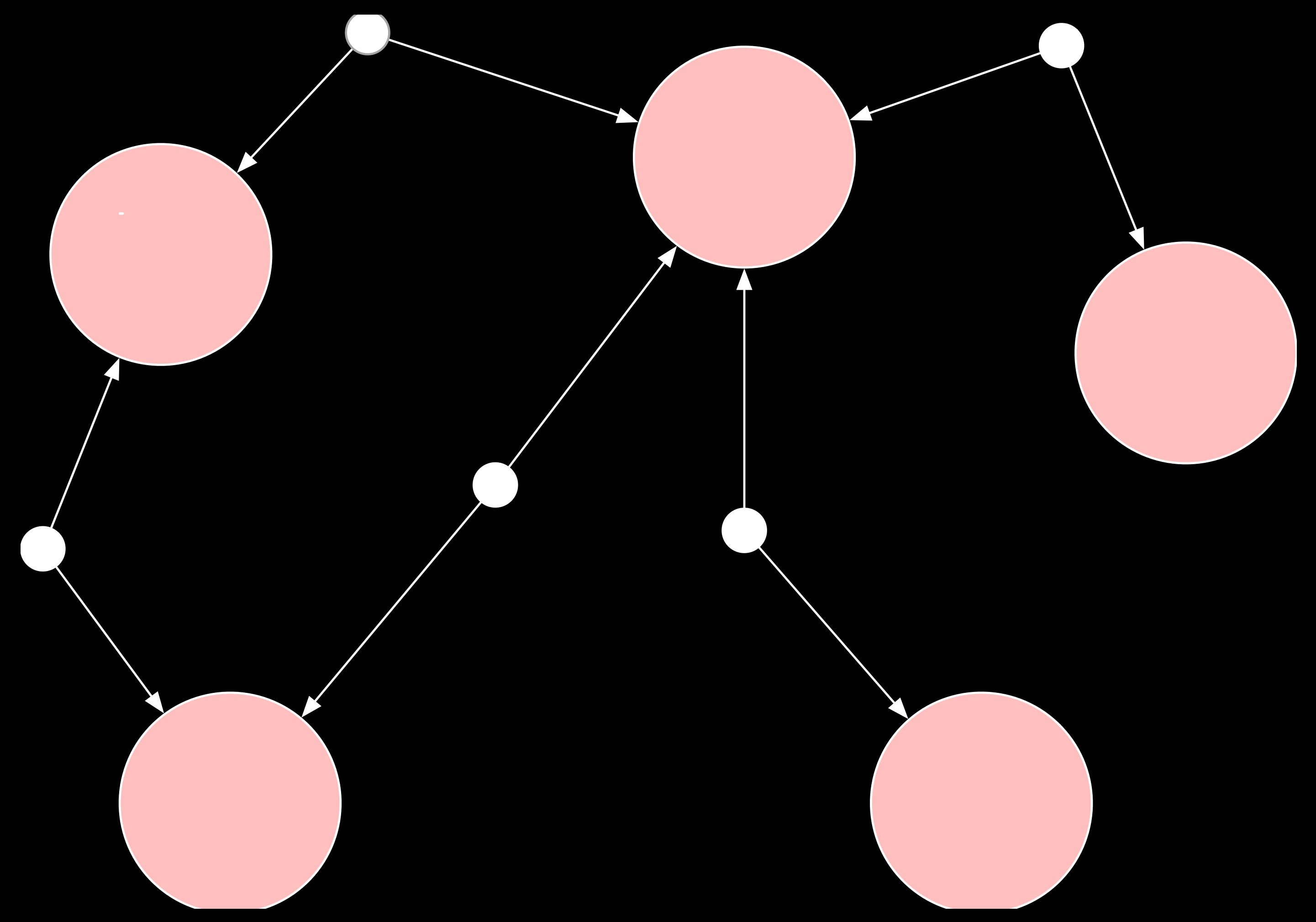
Compositionality

Structural **Representational** **Algorithmic**

This talk

Compositionality

Structural **Representational** **Algorithmic**

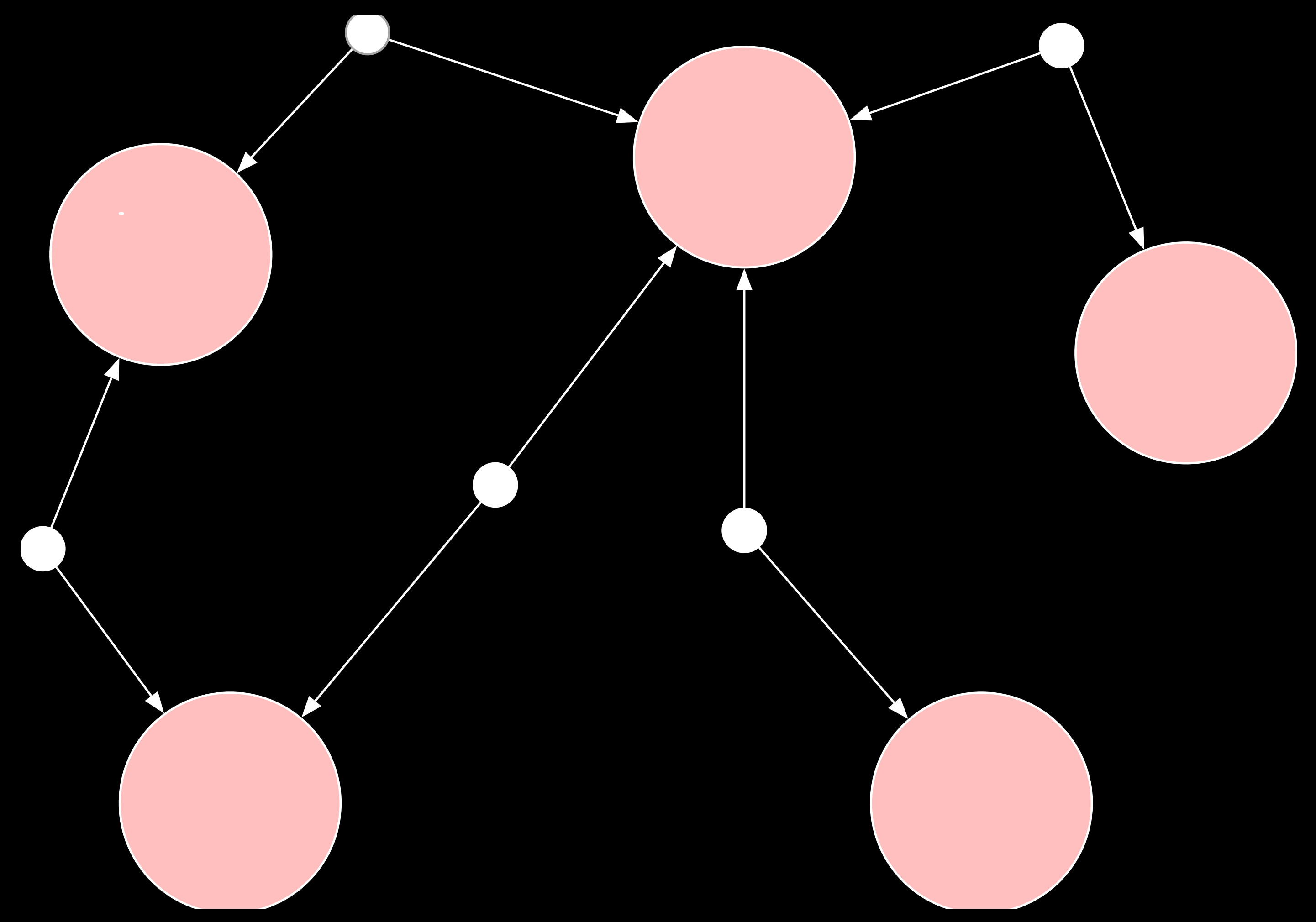


Decompositions

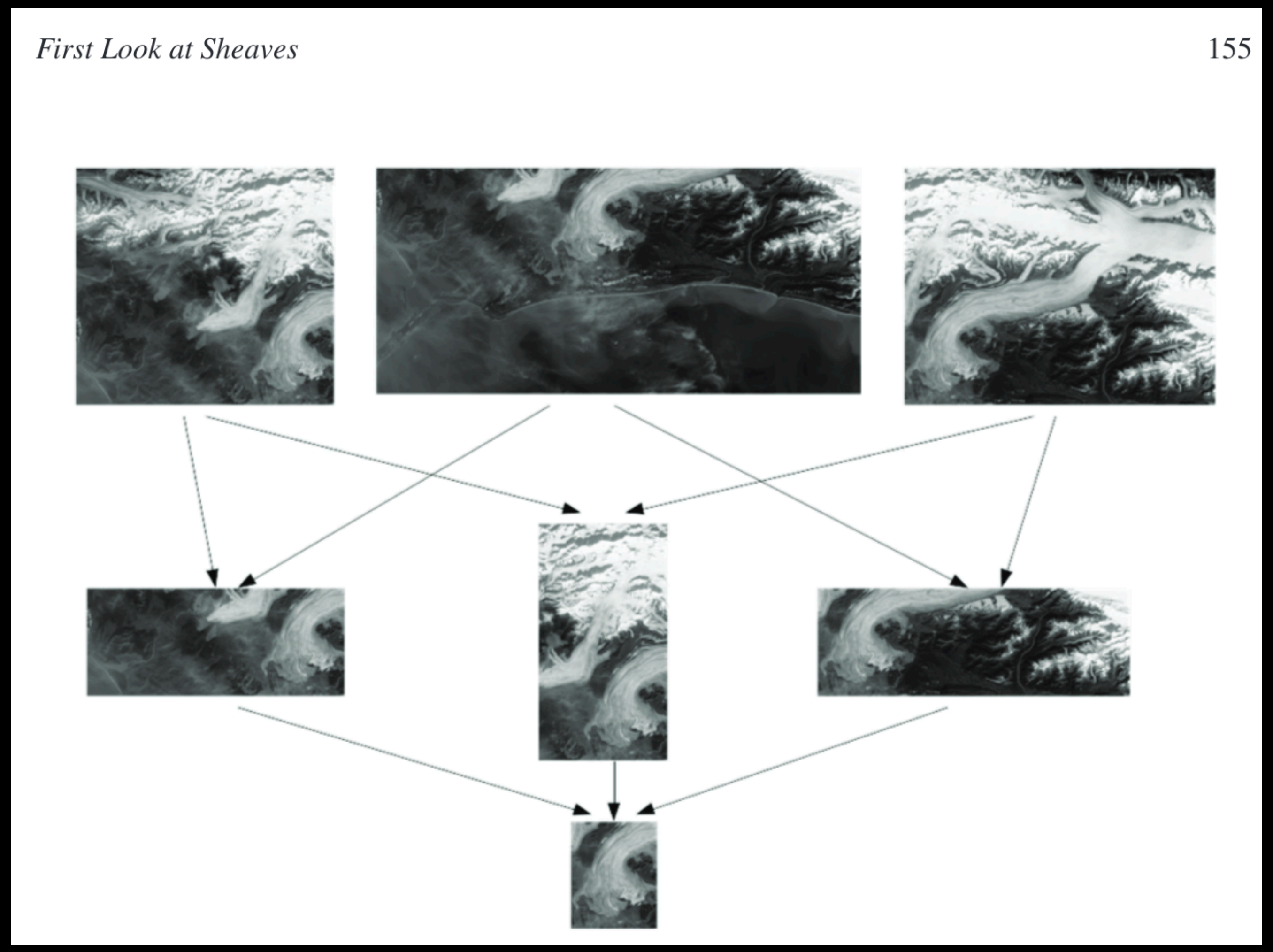
This talk

Compositionality

Structural | **Representational** | **Algorithmic**



Decompositions

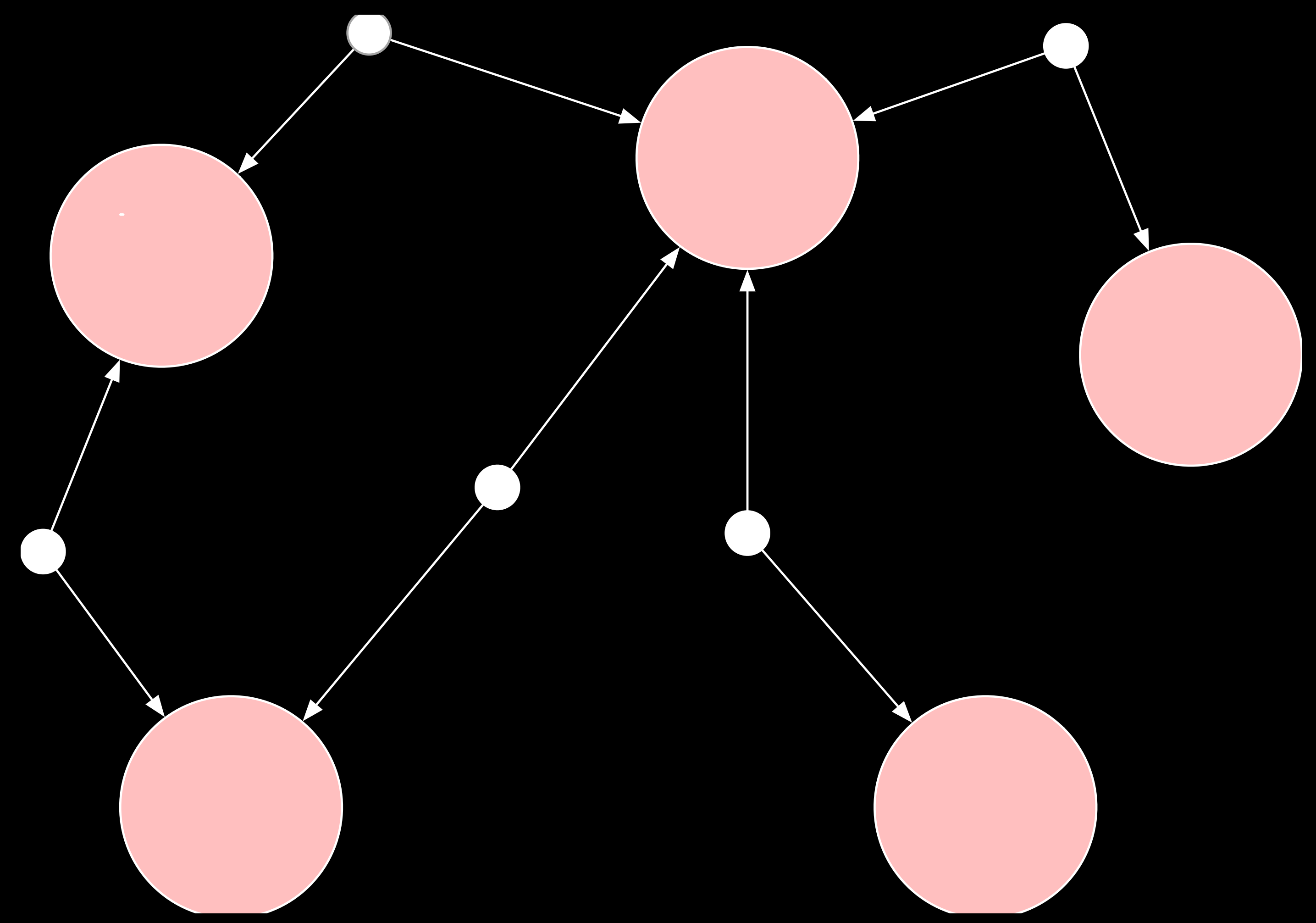


Sheaves

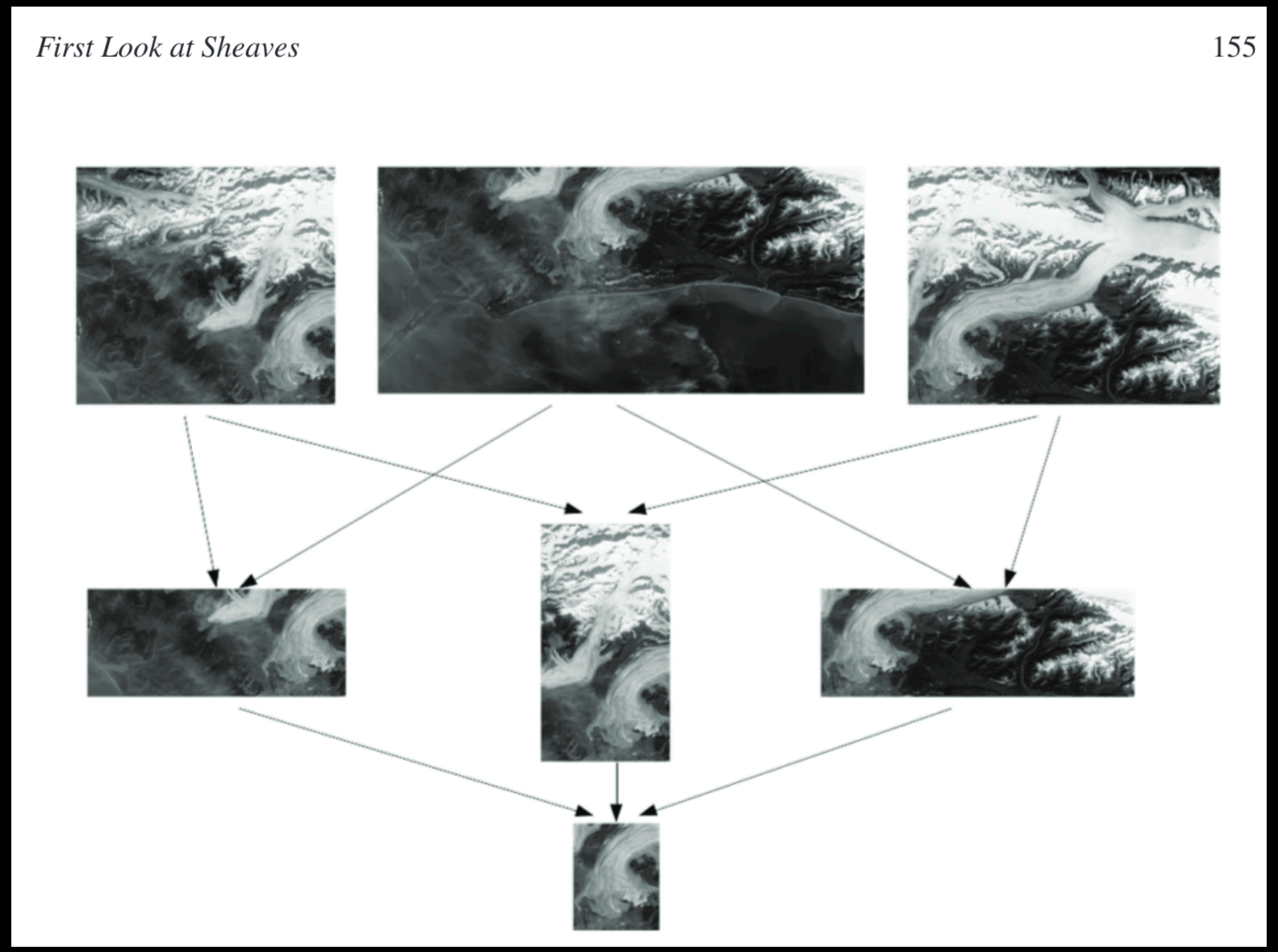
This talk

Compositionality

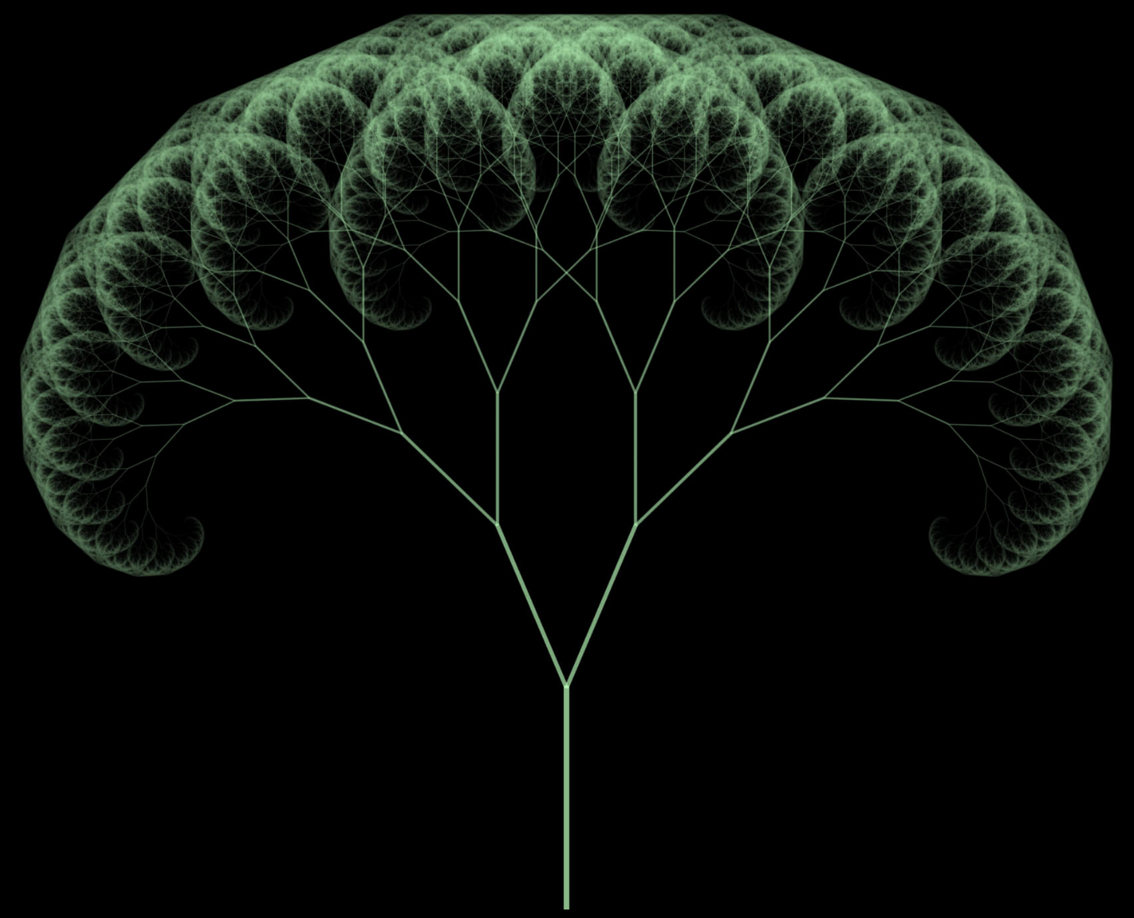
Structural | **Representational** | **Algorithmic**



Decompositions



Sheaves



Dynamic Programming

This talk

Use *category theory* to amalgamate these 3 perspectives

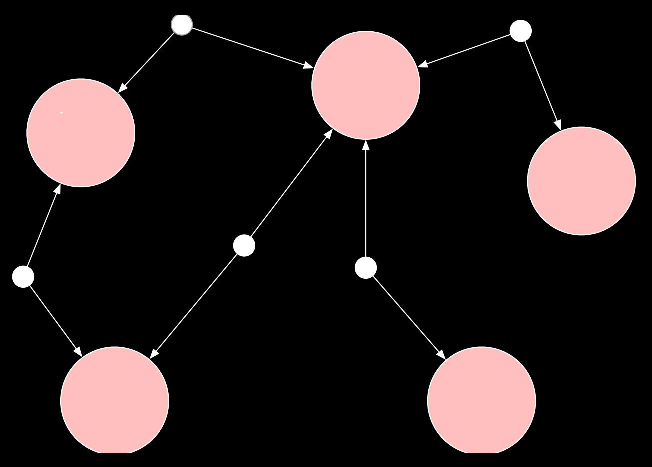
This talk

Use *category theory* to amalgamate these 3 perspectives

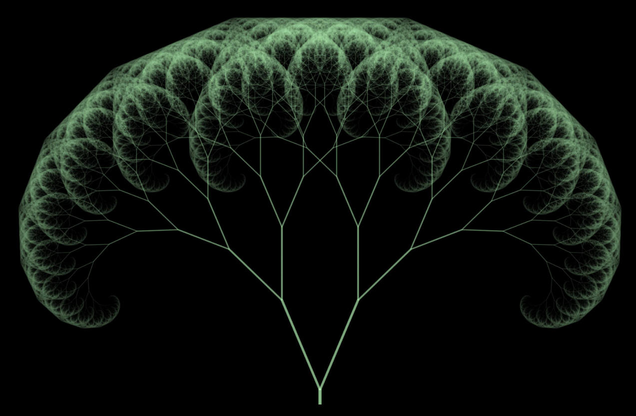
Category Theory

This talk

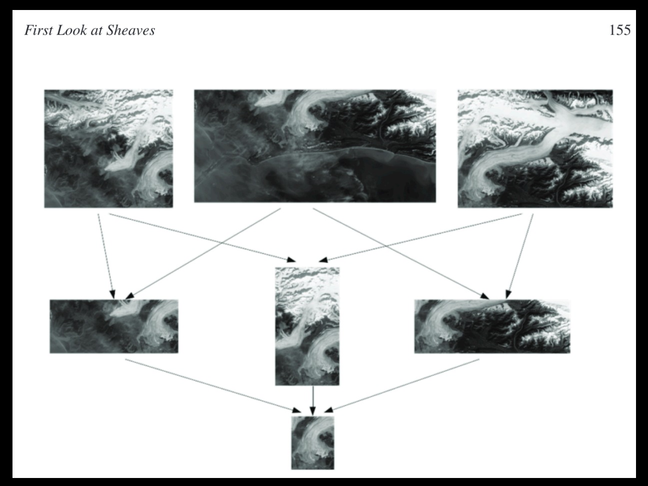
Use *category theory* to amalgamate these 3 perspectives



Decompositions



Dynamic Programming

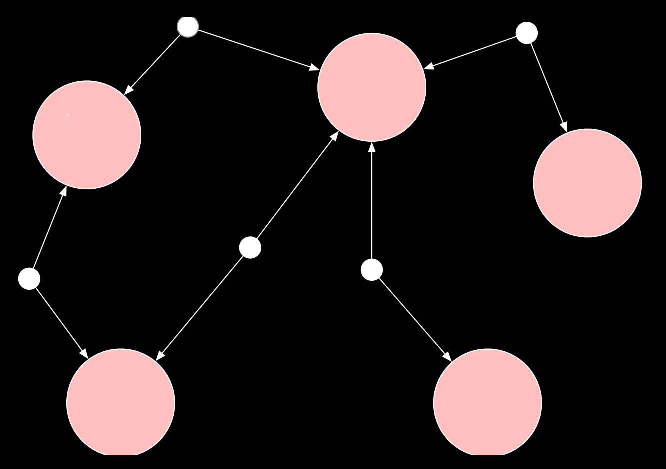


Sheaves

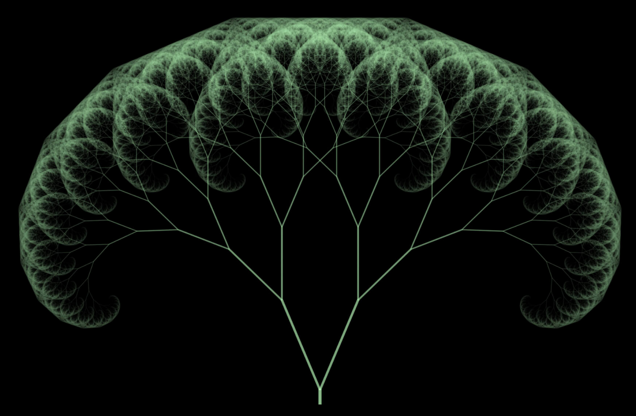
Category Theory

This talk

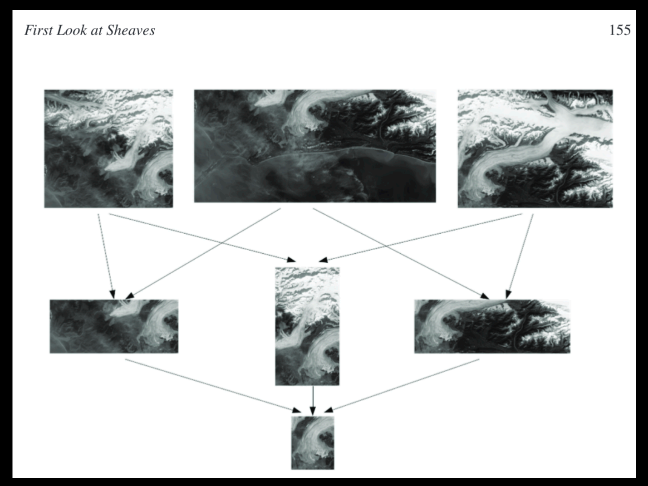
Use *category theory* to amalgamate these 3 perspectives



Decompositions



Dynamic Programming



Sheaves

Category Theory

Obtain Algorithmic meta-theorem:

Deciding Sheaves in linear time

on "nicely" decomposable classes of inputs

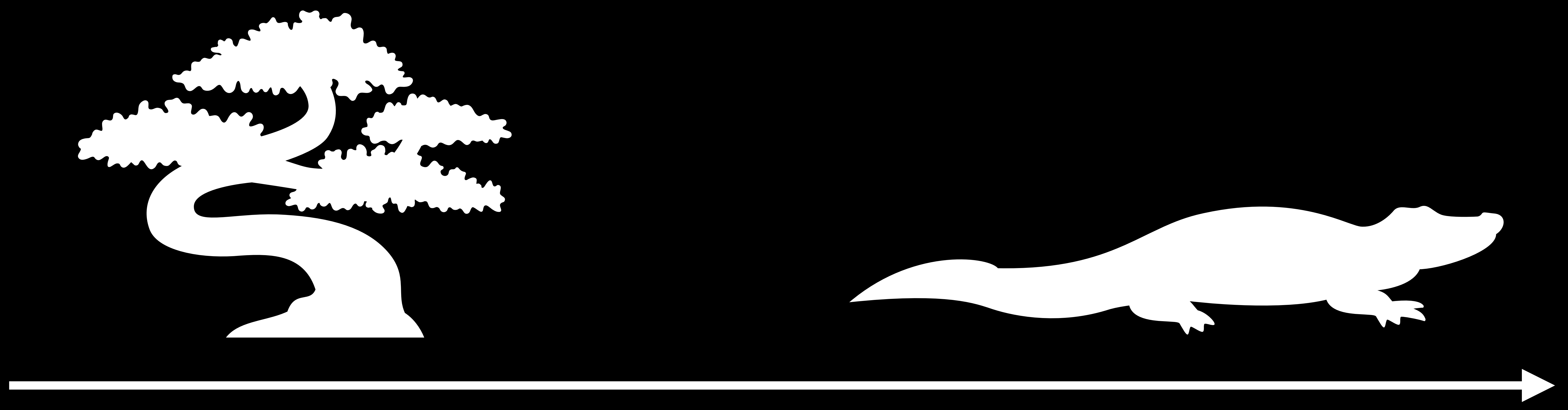
Why am I giving this talk?

Why am I giving this talk?

Structural Graph theory &
Complexity theory

(Graph minors, Tree-
Decompositions,
algorithms …)

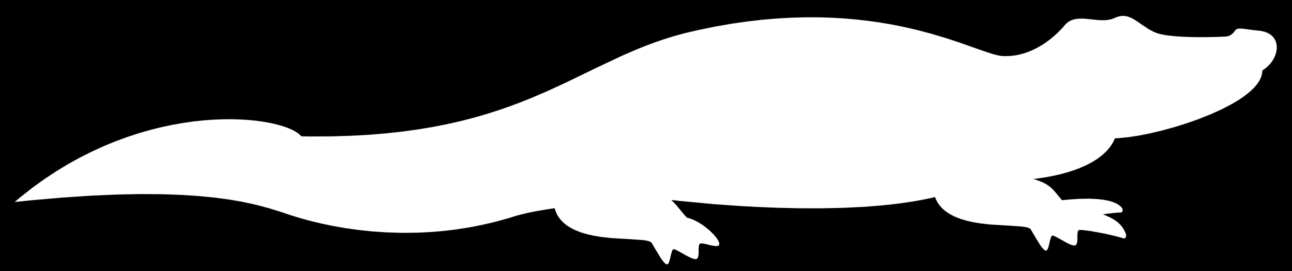
Why am I giving this talk?



Structural Graph theory &
Complexity theory

(Graph minors, Tree-
Decompositions,
algorithms …)

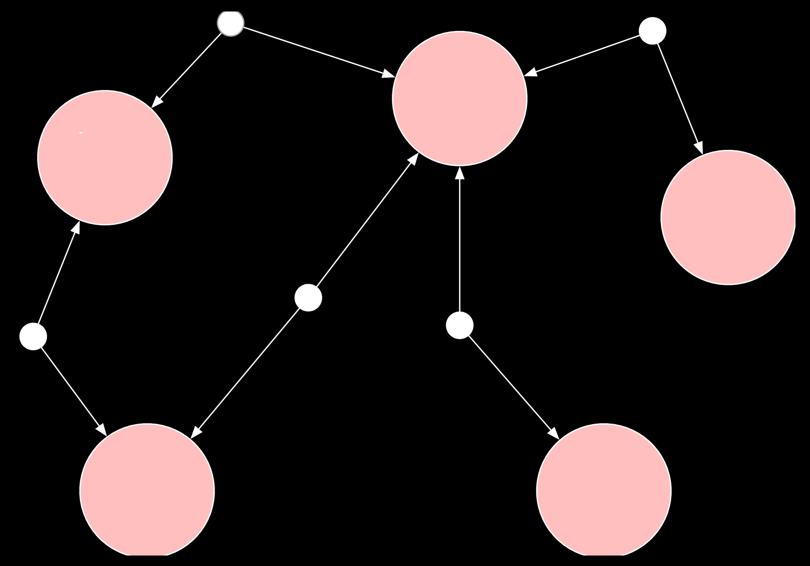
Why am I giving this talk?



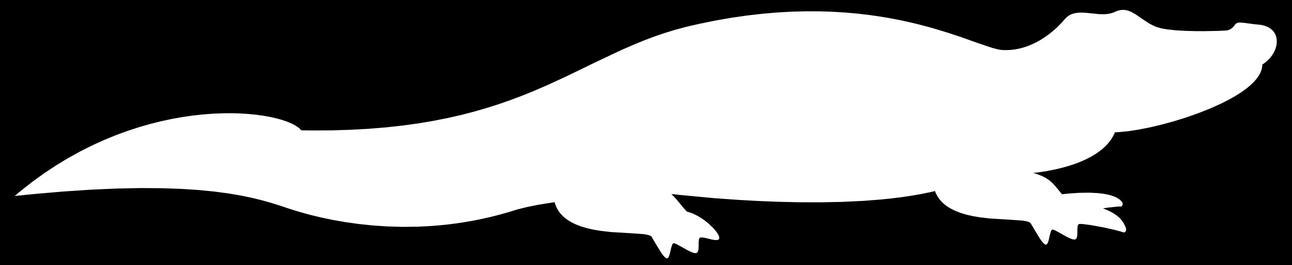
Structural Graph theory &
Complexity theory

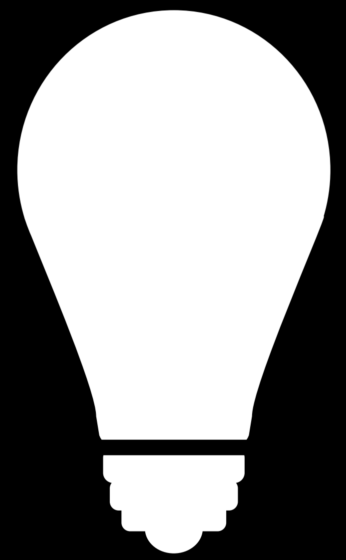
(Graph minors, Tree-Decompositions, algorithms …)

Category Theory



Why am I giving this talk?

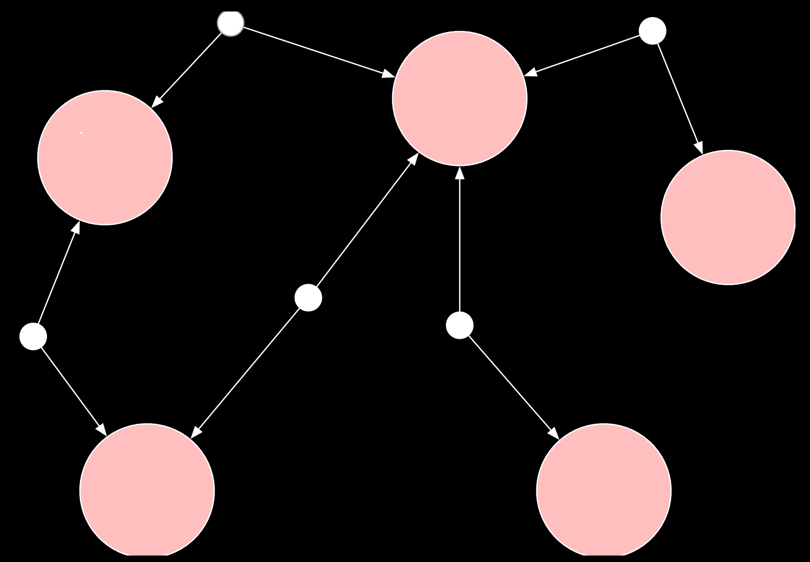




Structural Graph theory & Complexity theory

(Graph minors, Tree-Decompositions, algorithms …)

Category Theory



Lots of deep graph theory is about much more than graphs!

Why am I talking to *you*?

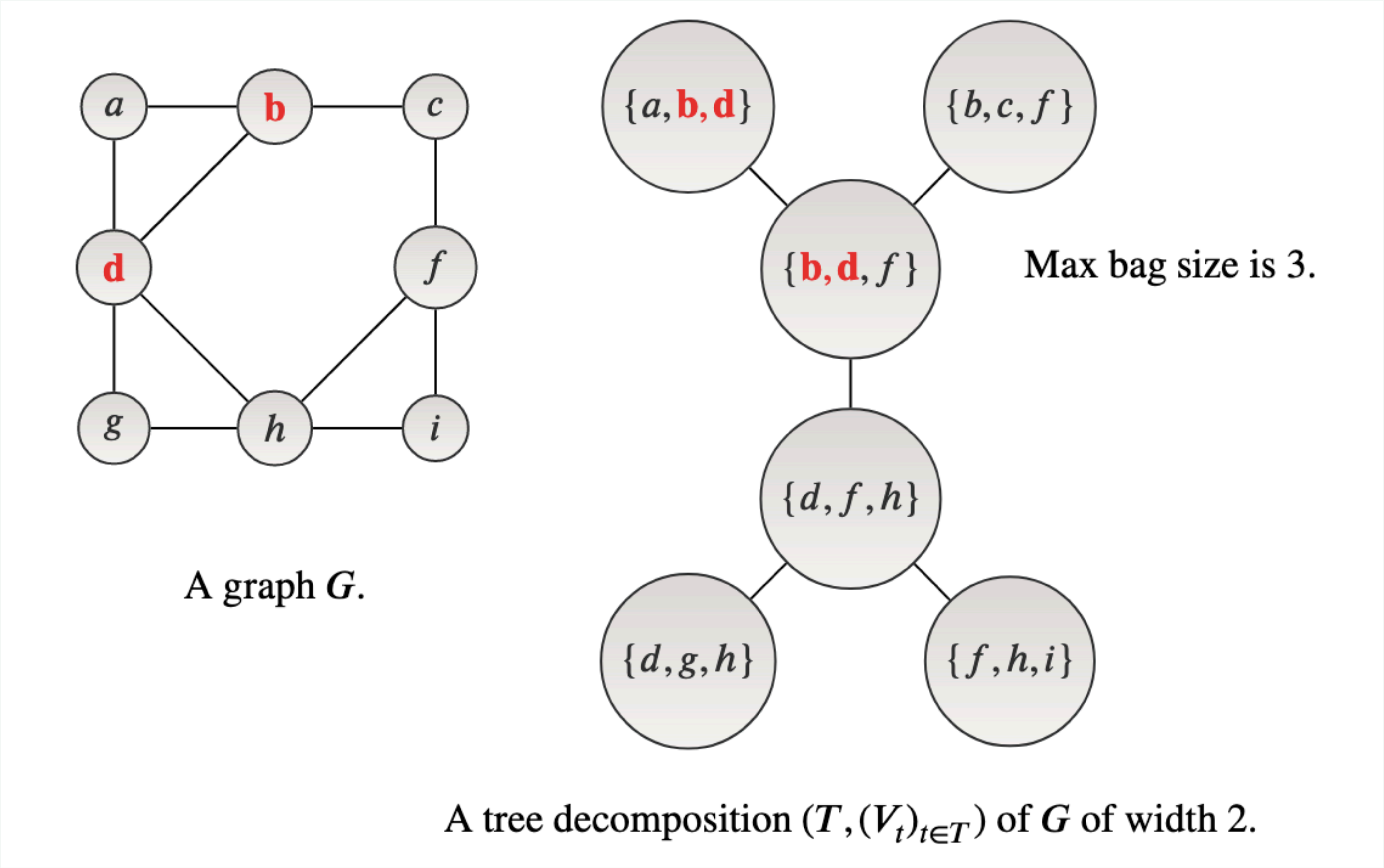
1. To get you excited about decompositions & algorithms
2. To get friendly feedback
3. I'm looking for collaborators!

The Nitty-Gritty PT1: Background

The Nitty-Gritty PT1: Background

Structural Graph theory & Complexity theory

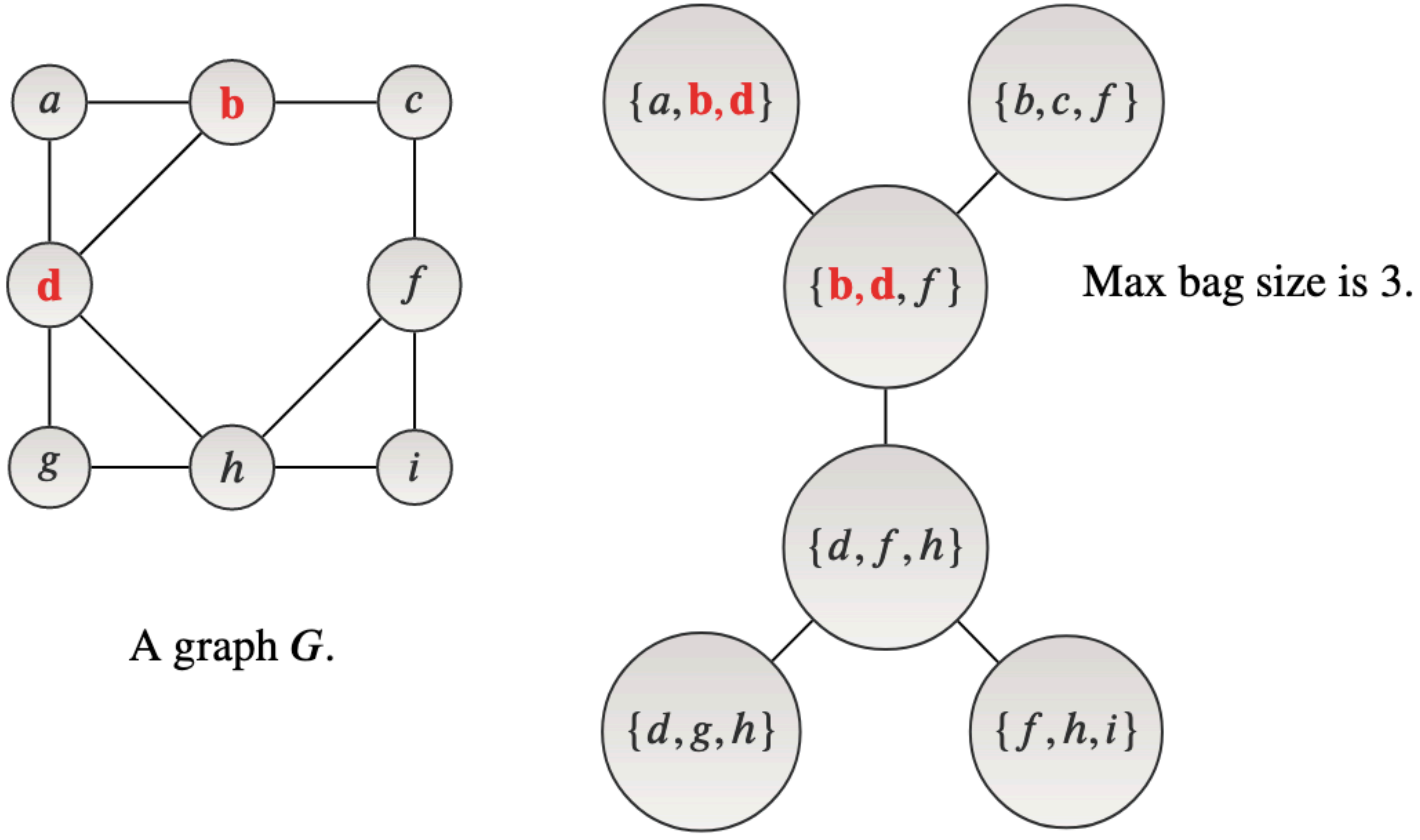
(Graph minors, Tree-Decompositions, algorithms …)



The Nitty-Gritty PT1: Background

3 perspectives on graph decompositions:

1. Completions
2. Decompositions
3. Width measures



A graph G.

A tree decomposition $(T, (V_t)_{t \in T})$ of G of width 2.

The Nitty-Gritty PT1: Background

1. Completions

Df/ Thm [Dirac]:

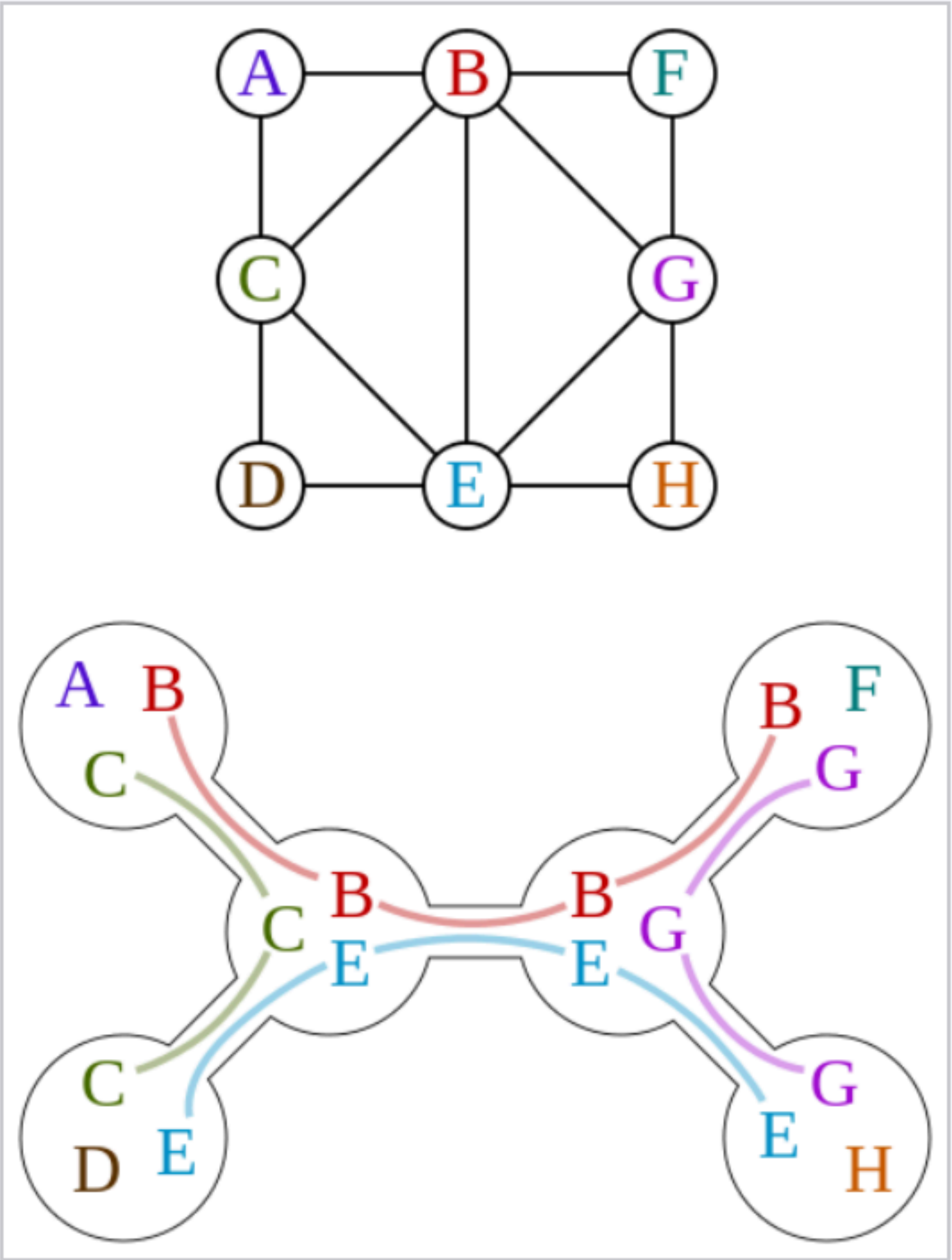
The class of **chordal graphs** is defined recursively as follows:

1. Every complete graph K_n is chordal

2. If H_1 and H_2 are chordal, then the pushout of any span $H_1 \hookleftarrow K_n \hookrightarrow H_2$ is a chordal graph.

The Nitty-Gritty PT1: Background

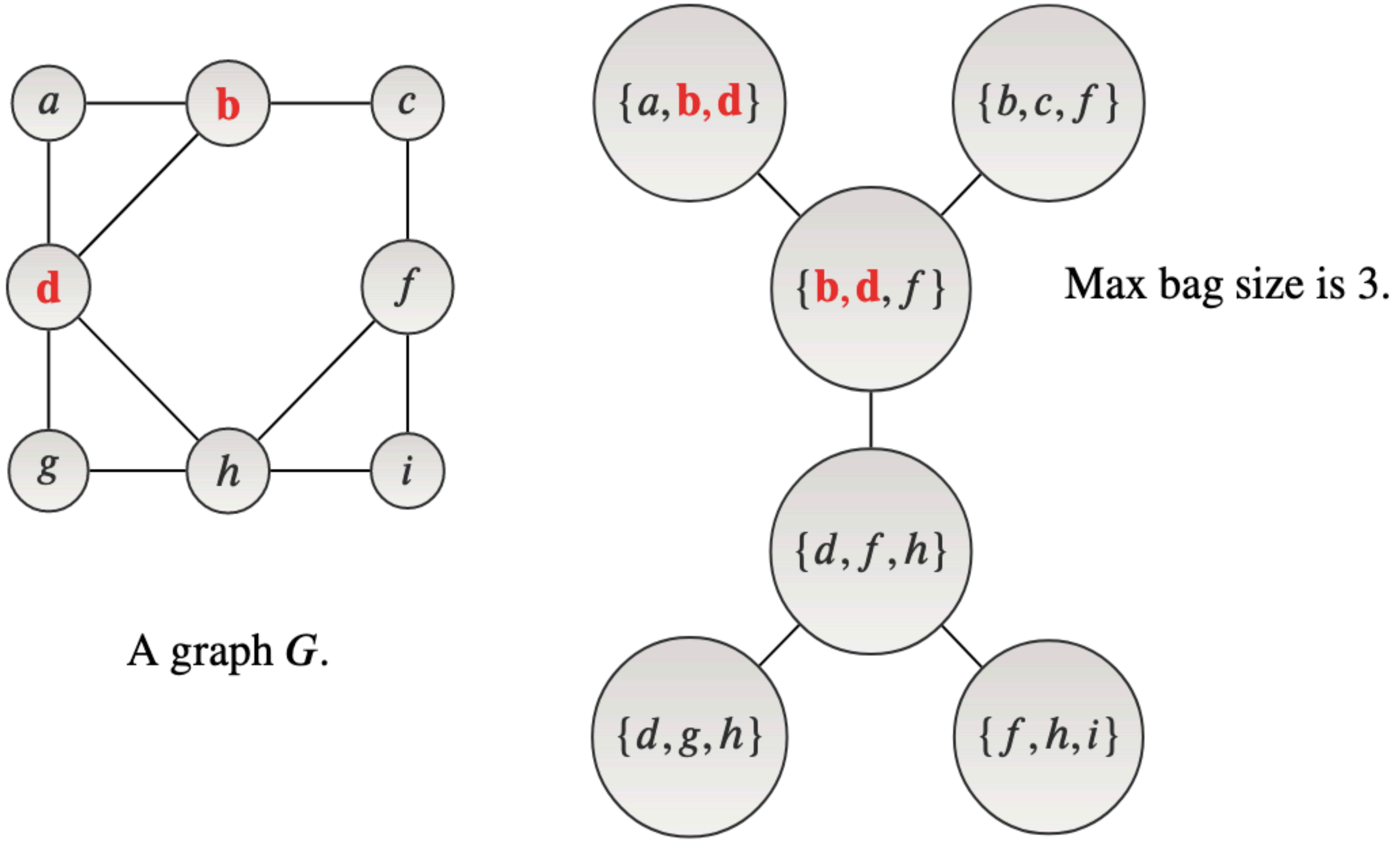
1. Completions



A chordal graph with eight vertices, represented as the intersection graph of eight subtrees of a six-node tree.

The Nitty-Gritty PT1: Background

1. Completions
2. Decompositions

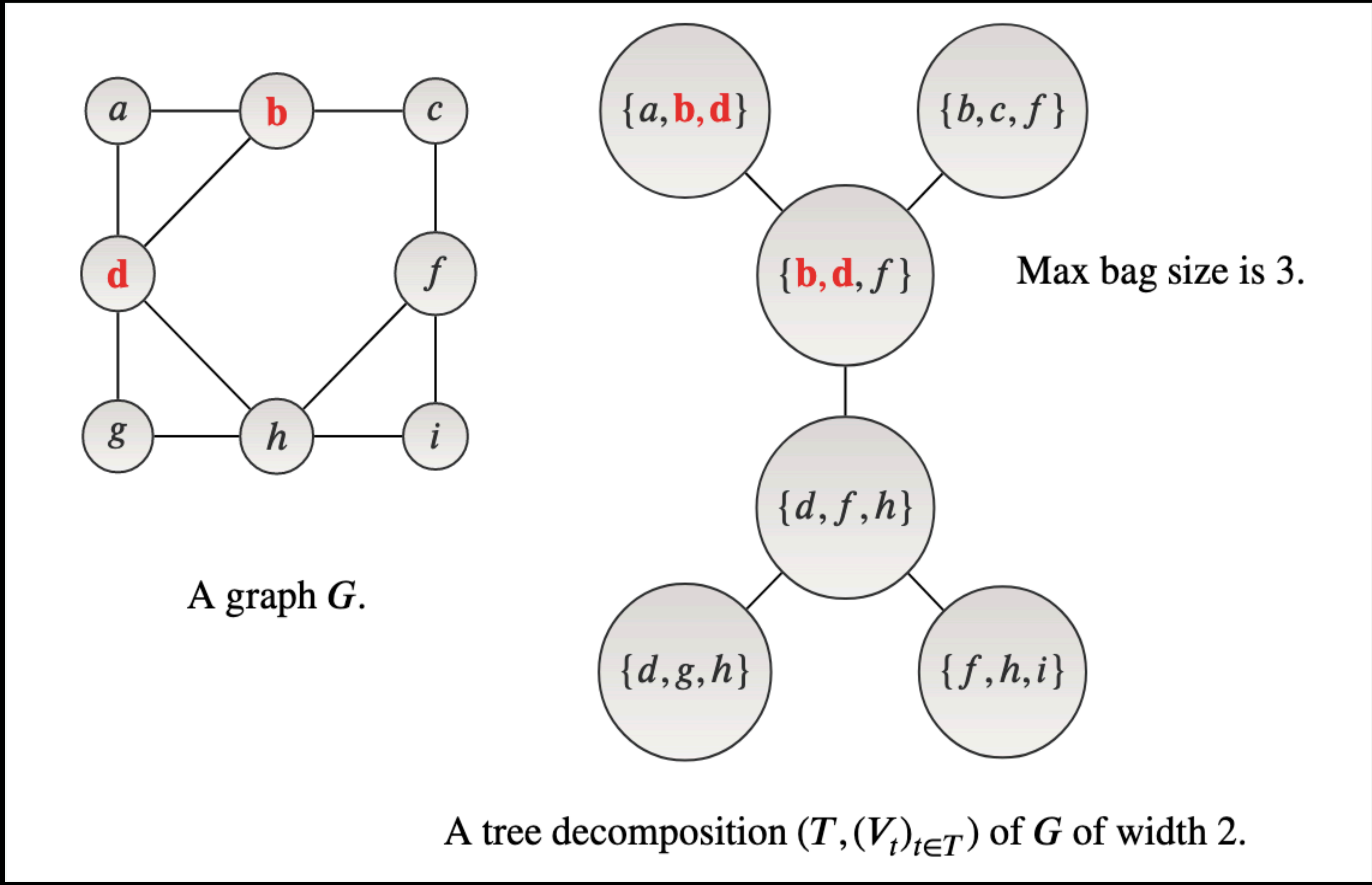


A graph G.

A tree decomposition $(T, (V_t)_{t \in T})$ of G of width 2.

The Nitty-Gritty PT1: Background

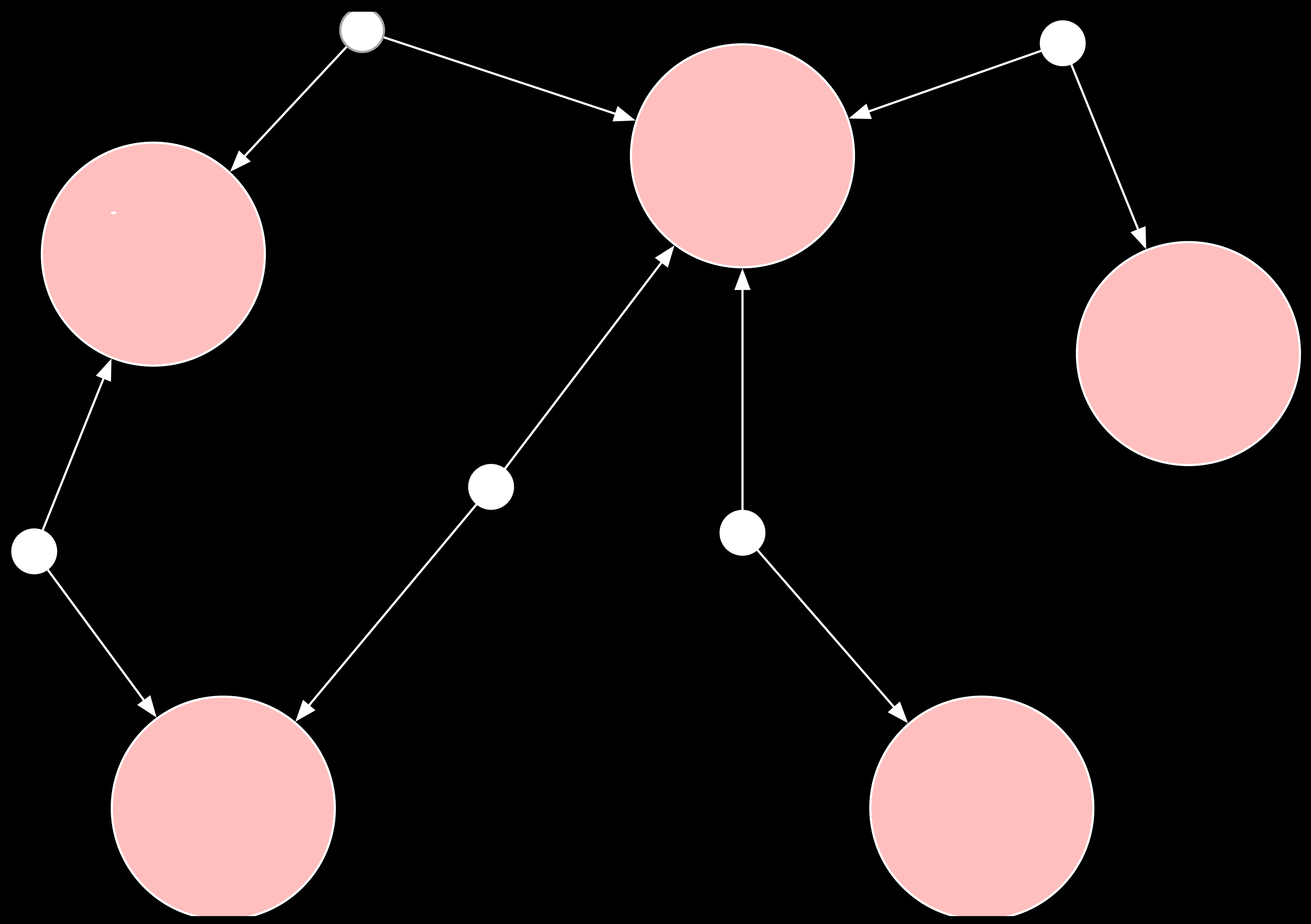
1. Completions
2. Decompositions
3. Width



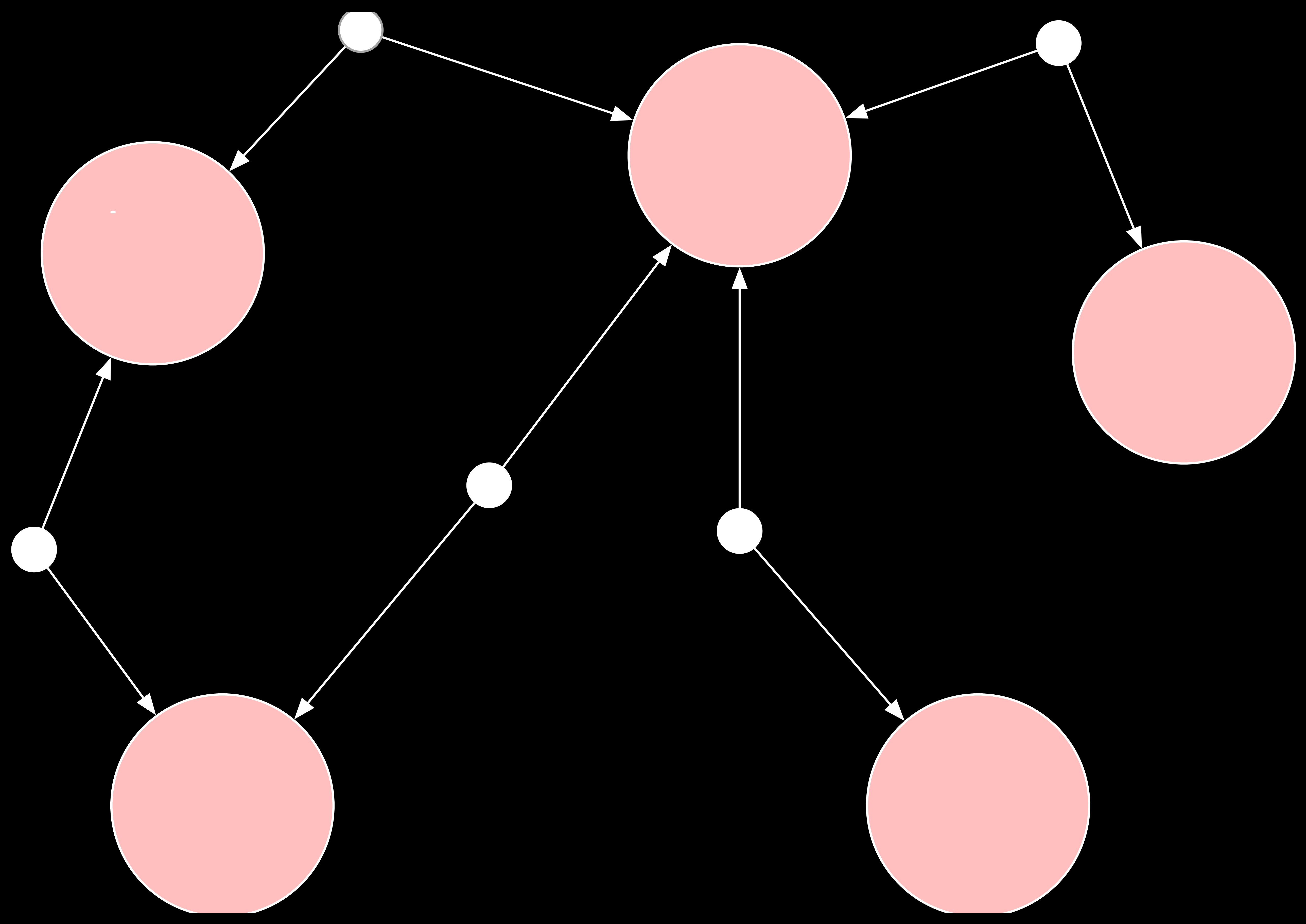
The Nitty-Gritty PT2: compositional data.

Structured decompositions

Structured decompositions



Structured decompositions



Structured decompositions

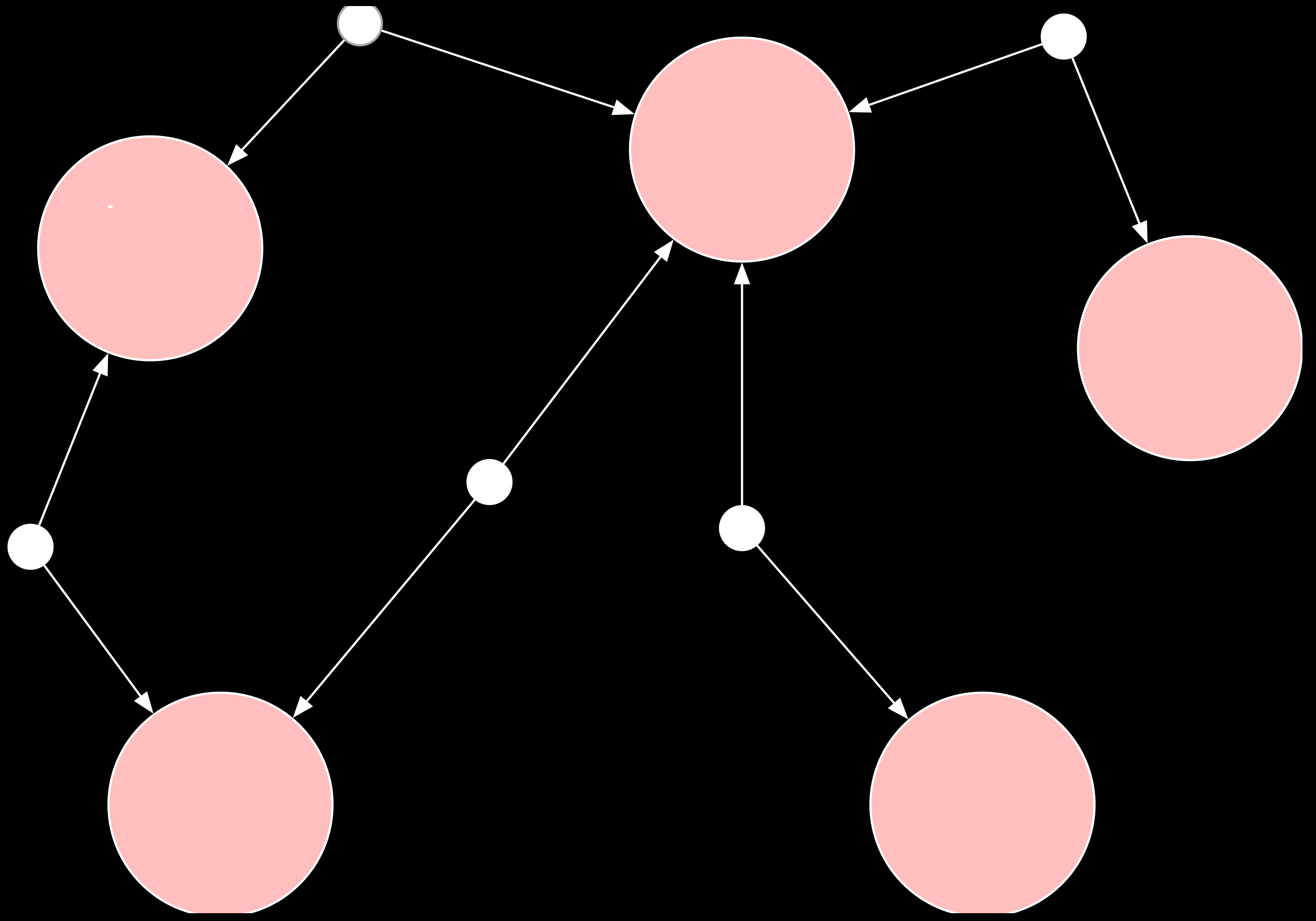
For any category C, a C**-valued structured decomposition of shape** G is a diagram

$$d\colon \int G \to C$$

where G is a graph

$$G\colon \mathrm{GrSch} \to \mathrm{FinSet}$$

$$\mathrm{GrSch} := s, t\colon E \to V$$



Structured decompositions

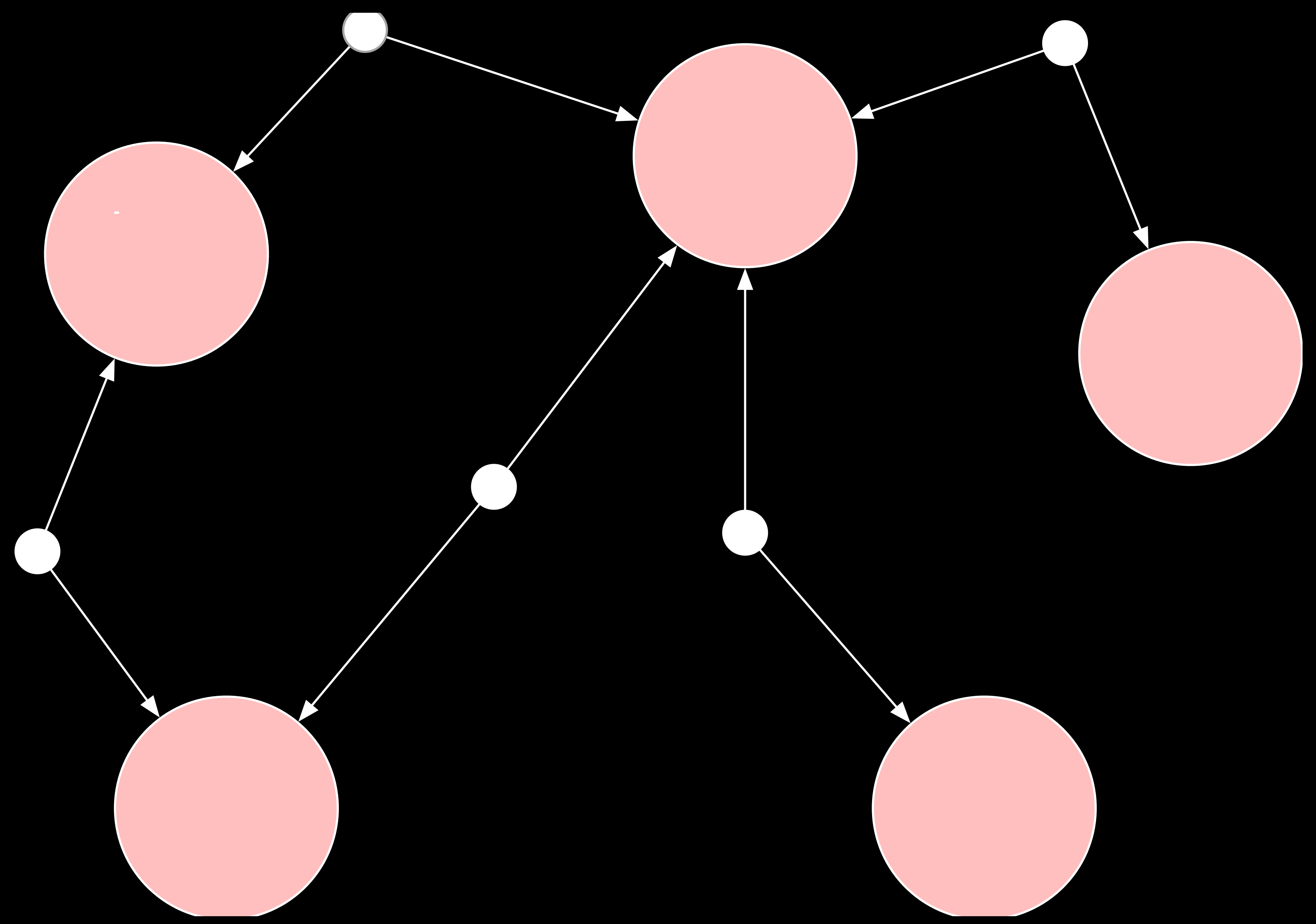
For any category C, a C**-valued structured decomposition of shape** G is a diagram

$$d\colon \int G \to C$$

where G is a graph

$$G\colon \mathrm{GrSch} \to \mathrm{FinSet}$$

$$\mathrm{GrSch} := s, t\colon E \to V$$



Structured decompositions

Structured decompositions

For any graph class $\mathcal{G}$, there is a **category**

$$\mathfrak{D}_{\mathcal{G}}\mathsf{C} \hookrightarrow \mathsf{Diag}\,\mathsf{C}$$

of C-valued structured decompositions with shapes in $\mathcal{G}$.

Structured decompositions

For any graph class $\mathscr{G}$, there is a **category**

$$\mathfrak{D}_{\mathscr{G}}\mathsf{C} \hookrightarrow \mathsf{Diag}\ \mathsf{C}$$

of C-valued structured decompositions with shapes in $\mathscr{G}$.

Objects: $d \colon \int G \to C$ with $G \in \mathscr{G}$

Structured decompositions

For any graph class $\mathscr{G}$, there is a **category**

$$\mathfrak{D}_{\mathscr{G}}\mathsf{C} \hookrightarrow \mathsf{Diag}\ \mathsf{C}$$

of C-valued structured decompositions with shapes in $\mathscr{G}$.

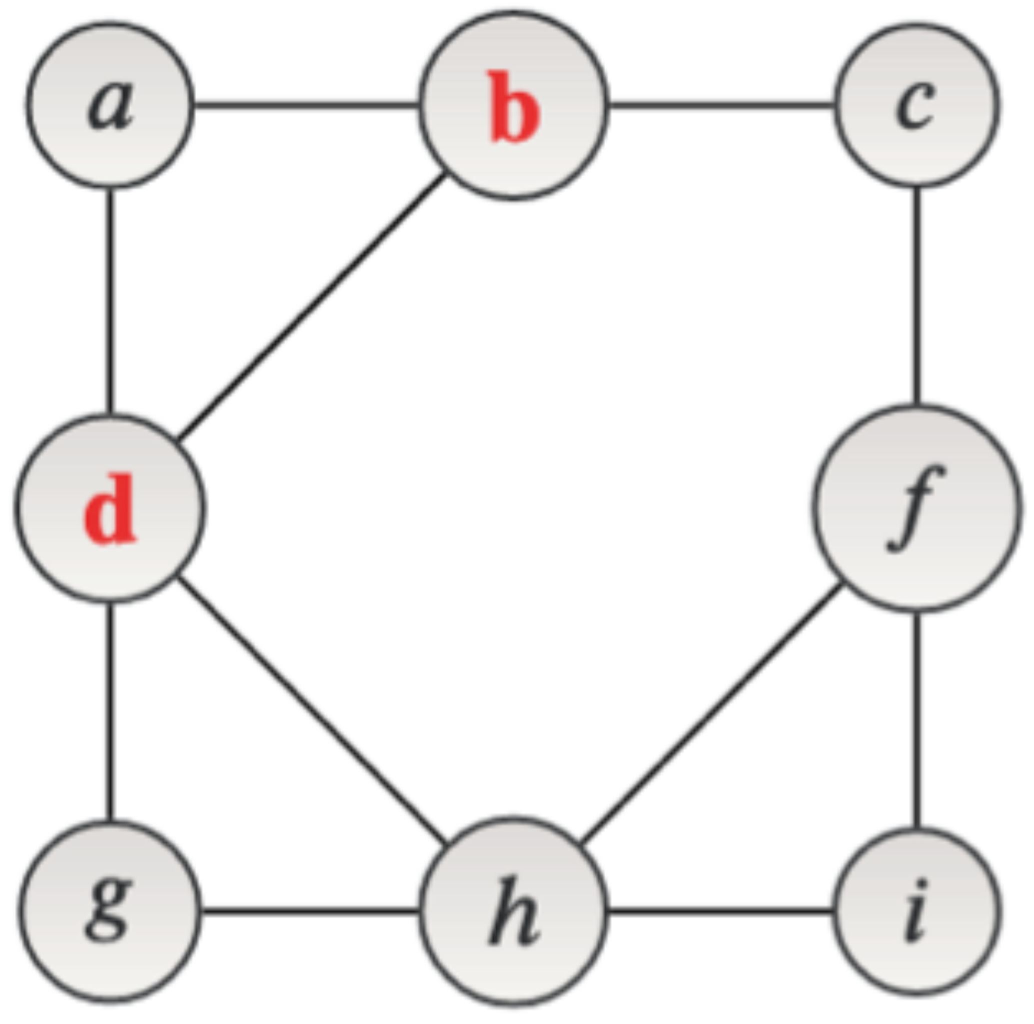
Objects: $d\colon \int G \to C$ with $G \in \mathscr{G}$

Morphisms: $(F, \eta)\colon \left(d\colon \int G \to C\right) \to \left(d'\colon \int G' \to C\right)$

where $F\colon \int G \to \int G'$ and $\eta\colon d \Rightarrow d' \circ F$

Structured decompositions (of graphs)

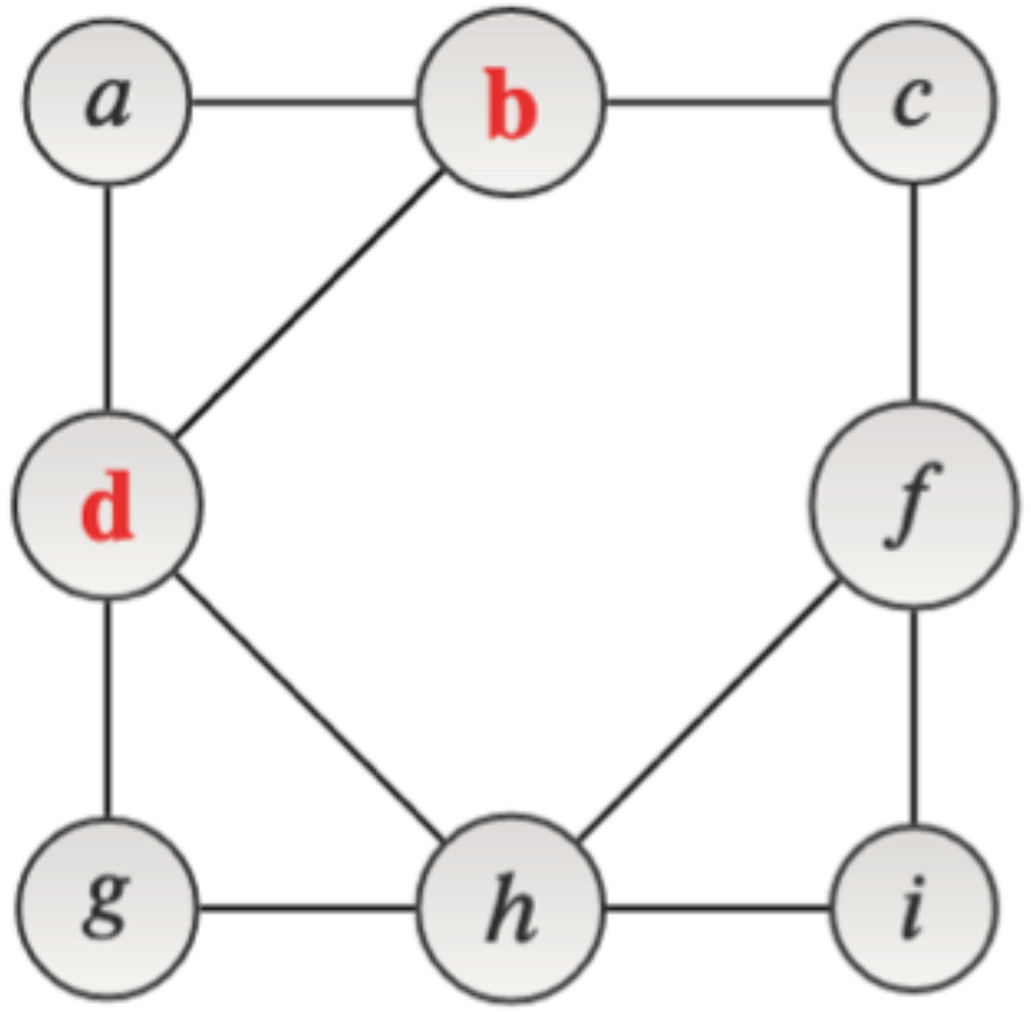
$$d \colon \int T \to \mathsf{Gr}$$



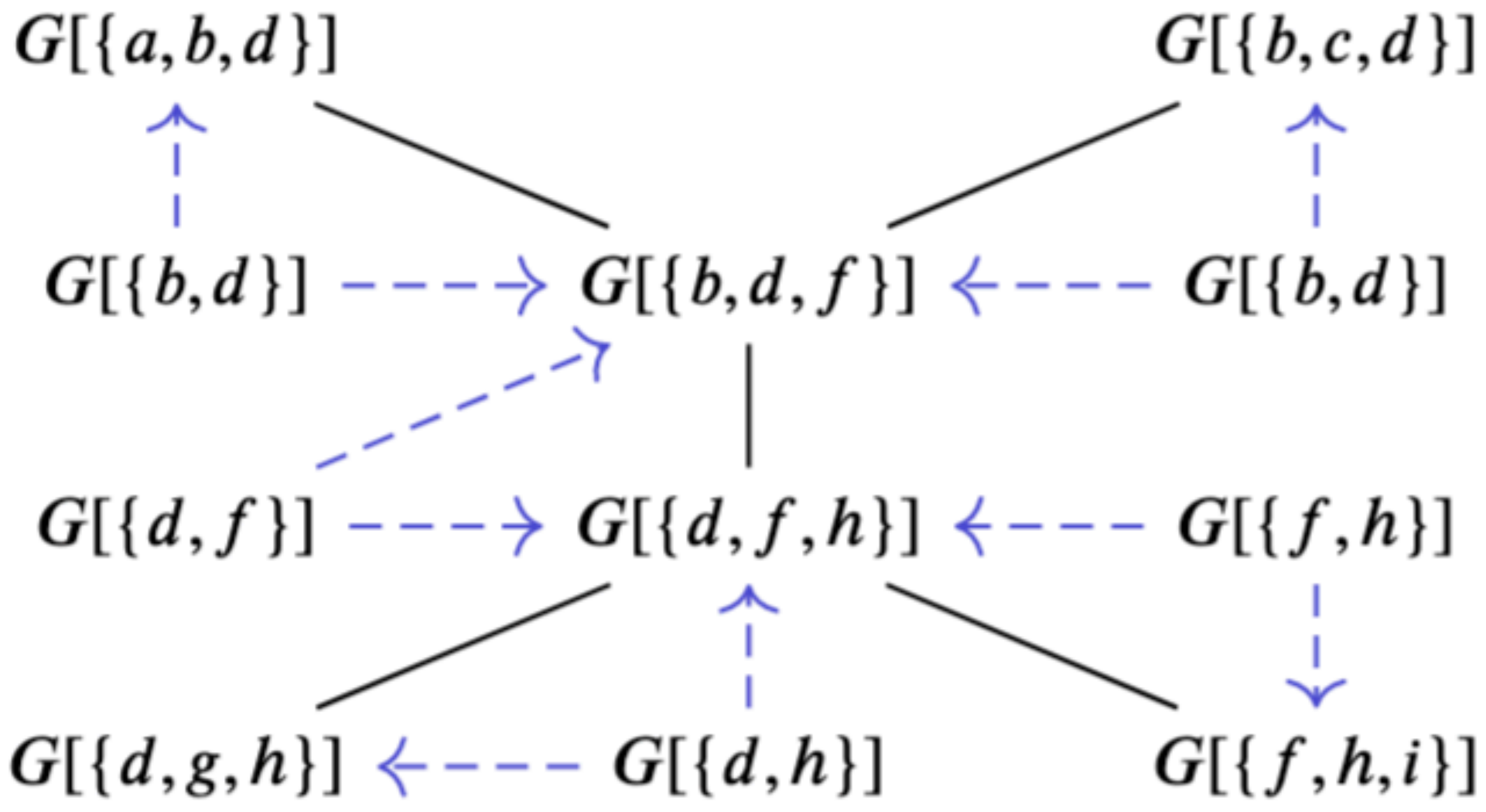
A graph G.

Structured decompositions (of graphs)

$$d \colon \int T \to \mathsf{Gr}$$



A graph G.



Defining Width Measures

Defining Width Measures

Aim: use decompositions to define a measure of *structural complexity*

Recall “completions vs decompositions”

Defining Width Measures

Aim: use decompositions to define a measure of *structural complexity*

Recall “completions vs decompositions”

monos into *chordal graphs* — *“recipes for constructing a graph”*

Defining Width Measures

Idea: pick out some "atomic building blocks" from C

$$\Omega : \mathrm{K} \to \mathrm{C}$$

Defining Width Measures

Idea: pick out some "atomic building blocks" from C

$$\Omega\colon \mathsf{K} \to \mathsf{C}$$

Construct "maximally dense" compositional structures

$$\mathfrak{D}\mathsf{K} \xrightarrow{\mathfrak{D}\Omega} \mathfrak{D}\mathsf{C} \xrightarrow{\text{colim}} \mathsf{C}$$

Defining Width Measures

Idea: pick out some “atomic building blocks” from C

$$\Omega \colon \mathsf{K} \to \mathsf{C}$$

Construct “maximally dense” compositional structures

$$\mathfrak{D}\mathsf{K} \xrightarrow{\mathfrak{D}\Omega} \mathfrak{D}\mathsf{C} \xrightarrow{\text{colim}} \mathsf{C}$$

Def: a $(\mathscr{G}, \Omega)$**-completion** of some $c \in \mathsf{C}$ is a mono

$$c \hookrightarrow (\text{colim} \circ \mathfrak{D}\Omega)d.$$

Say that $c \in \mathsf{C}$ has $(\mathscr{G}, \Omega)$**-width at most** K if it admits a $(\mathscr{G}, \Omega)$-completion

Defining Width Measures

Defining Width Measures

Aim: use decompositions to define a measure of *structural complexity*

Recall “completions vs decompositions”

Defining Width Measures

Aim: use decompositions to define a measure of *structural complexity*

Recall “completions ✓ vs decompositions”

Defining Width Measures

Aim: use decompositions to define a measure of *structural complexity*

Recall “completions ✓ vs decompositions”

Lemma 5.10. *Let* $\mathbf{C}$ *be an adhesive category,* $\mathbf{J}$ *be a non-empty category with finitely-many objects and whose arrows are monic,* $D : \mathbf{J} \to \mathbf{C}$ *be a diagram in* $\mathbf{C}$ *which preserves monomorphisms,* Λ *be a colimit conone over* D *and* $\delta : X \to \operatorname{colim} D$ *be an arrow. Then* X *is the colimit of the diagram* $D_X : \mathbf{J} \to \mathbf{C}$ *obtained by pullback of* Λ *along* δ *as in the following diagram.*

$$\begin{tikzcd}
X \cong \operatorname{colim} D_x \arrow[r, "\delta"] & \operatorname{colim} D \\
D_x J \arrow[u, Rightarrow] \arrow[r, Rightarrow] \arrow[ur, phantom, very near start, "\urcorner"] & DJ \arrow[u, Rightarrow, "\Lambda"']
\end{tikzcd}$$

[Bumpus, Kocsis, Master]

Defining Width Measures

Summary:

- Pick out “atomic” objects to be viewed as “complex”
- This, together with completions —> define structural complexity
- From any completion we can recover a genuine decomposition

The Nitty-Gritty PT2.

Compositional algorithms: deciding sheaves on presheaves

Algorithms

$$\mathsf{C} \xrightarrow{\quad \mathcal{F} \quad} \mathsf{finset}^{op} \xrightarrow{\quad \mathrm{dec}^{op} \quad} \mathbf{2}^{op}$$

Algorithms

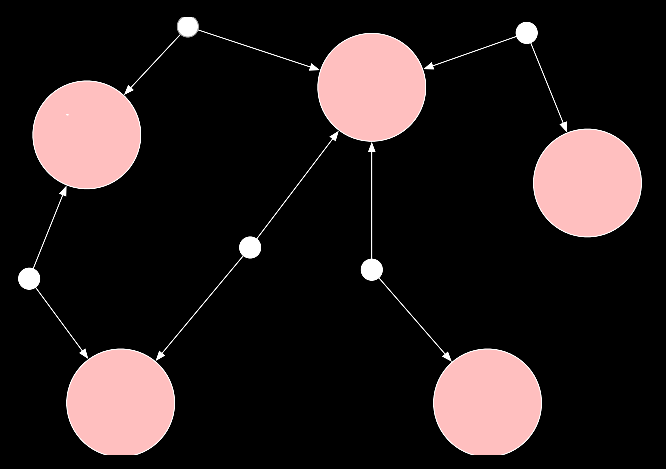
$$\mathsf{C} \xrightarrow{\quad\mathcal{F}\quad} \mathsf{finset}^{op} \xrightarrow{\quad\mathrm{dec}^{op}\quad} \mathbf{2}^{op}$$

Our aim:

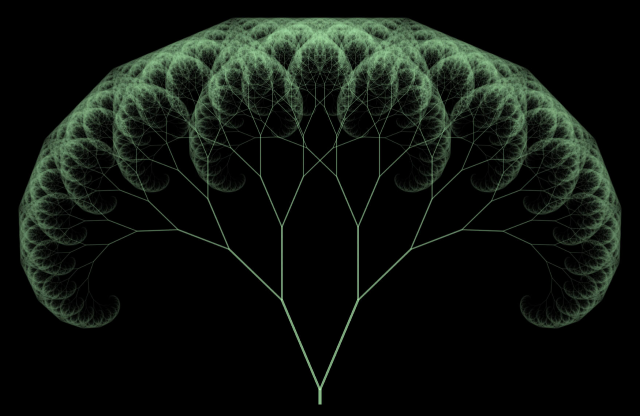
Our aim:

Category Theory

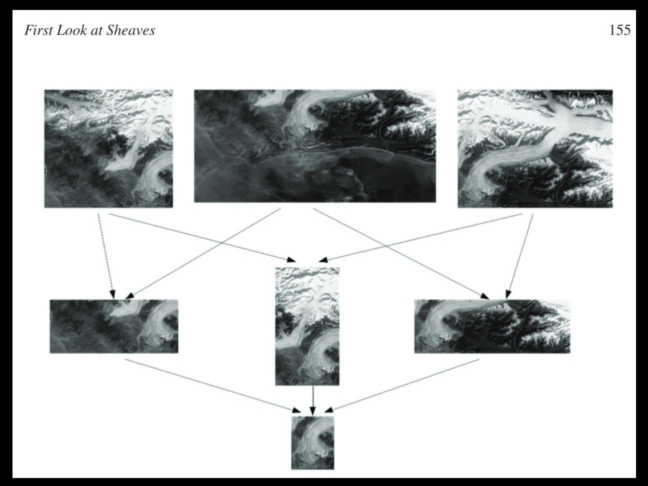
Our aim:



Decompositions



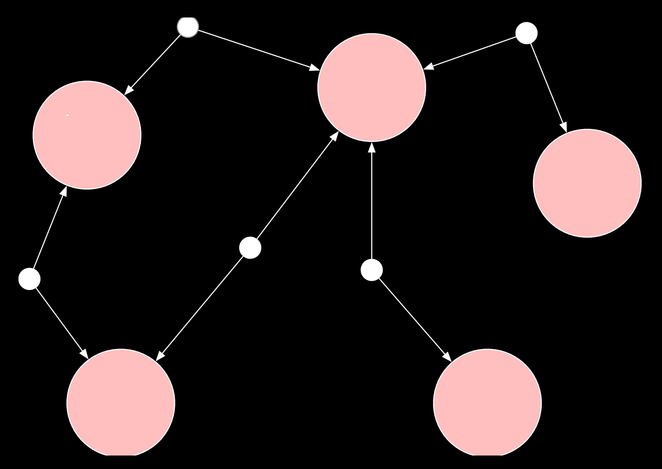
Dynamic
Programming



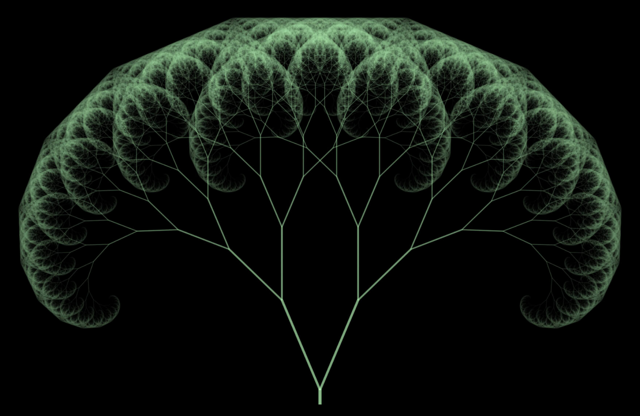
Sheaves

Category Theory

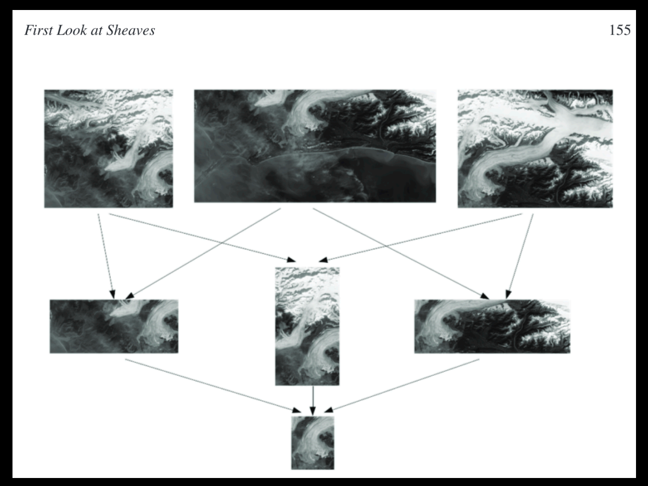
Our aim:



Decompositions



Dynamic Programming



Sheaves

Category Theory

Obtain Algorithmic meta-theorem:

Deciding Sheaves in linear time

on “nicely” decomposable classes of inputs

Algorithms

Structured decomps. yield **Grothendieck topologies** on *adhesive categories*

w/ sufficiently many colimits…

Algorithms

Structured decomps. yield **Grothendieck topologies** on *adhesive categories*

w/ sufficiently many colimits…

Theorem 3.5. *If* C *is a small adhesively cocomplete category, then the functor* Dcmp *(defined below) is a subfunctor of* subMon.

$$\mathsf{Dcmp} : \mathsf{C}^{op} \to \mathbf{Set}$$

$$\mathsf{Dcmp} : c \mapsto \{d \mid \operatorname{colim} d = c \text{ and } d \in \mathfrak{D}\mathsf{C}\}$$

$$\mathsf{Dcmp} : (f : a \to b) \mapsto \big(f^* : (d \in \mathsf{subMon}\, b) \mapsto (d_f \in \mathsf{subMon}\, a)\big).$$

Furthermore, for any such C*, we have that* $(\mathsf{C}, \mathsf{Dcmp})$ *and* $\left(\mathsf{C}_{mono}, \mathsf{Dcmp}|_{\mathsf{C}_{mono}}\right)$ *are sites.*

[Althaus, Bumpus, Fairbanks, Rosiak]

Algorithms

Structured decomps. yield **Grothendieck topologies** on *adhesive categories*

w/ sufficiently many colimits…

Algorithms

Structured decomps. yield **Grothendieck topologies** on *adhesive categories*

w/ sufficiently many colimits…

… we can speak of sheaves with respect to the decomposition topology

Algorithms

$$
\begin{tikzcd}
 & \mathsf{C} \arrow[r, "\mathcal{F}"] & \mathsf{FinSet}^{op} \arrow[r, "\mathrm{dec}^{op}"] & \mathbf{2}^{op} \\
\int \mathsf{Dcmp} \arrow[ur, "\mathsf{fst}"] \arrow[dr, "\mathsf{snd}"'] & \textit{comm.} & & \\
 & \mathfrak{D}\mathsf{C} \arrow[r, "\mathfrak{D}\mathcal{F}"] & \mathfrak{D}\mathsf{FinSet}^{op} \arrow[uu, "\mathsf{Lan}_{\mathfrak{D}}\, \mathsf{id}_{\mathsf{FinSet}^{op}}"'] &
\end{tikzcd}
$$

Algorithms

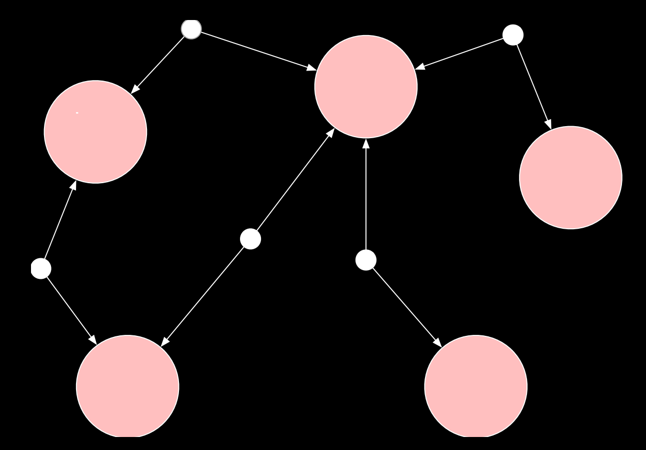
$$
\begin{tikzcd}
 & \mathsf{C} \arrow[r, "\mathcal{F}"] & \mathsf{FinSet}^{op} \arrow[rr, "\mathrm{dec}^{op}"] & & \mathbf{2}^{op} \\
\int \mathsf{Dcmp} \arrow[ur, "\mathsf{fst}"] \arrow[dr, "\mathsf{snd}"'] & \arrow[r, phantom, "\textit{comm.}"] & {} & \arrow[r, phantom, "\textit{comm.}\text{ (by Thm. 4.1)}"] & {} \\
 & \mathfrak{D}\mathsf{C} \arrow[r, "\mathfrak{D}\mathcal{F}"'] & \mathfrak{D}\mathsf{FinSet}^{op} \arrow[uu, "\mathsf{Lan}_{\mathfrak{D}}\, \mathsf{id}_{\mathsf{FinSet}^{op}}"'] \arrow[r, "\mathcal{A}"'] & \mathfrak{D}\mathsf{FinSet}^{op} \arrow[r, "\mathfrak{D}\mathrm{dec}^{op}"'] & \mathfrak{D}\mathbf{2}^{op} \arrow[uu, "\wedge"']
\end{tikzcd}
\tag{4}
$$

Recap

Recap

Structured decompositions represent compositional data.

$$d\colon \int G \to \mathsf{C}$$

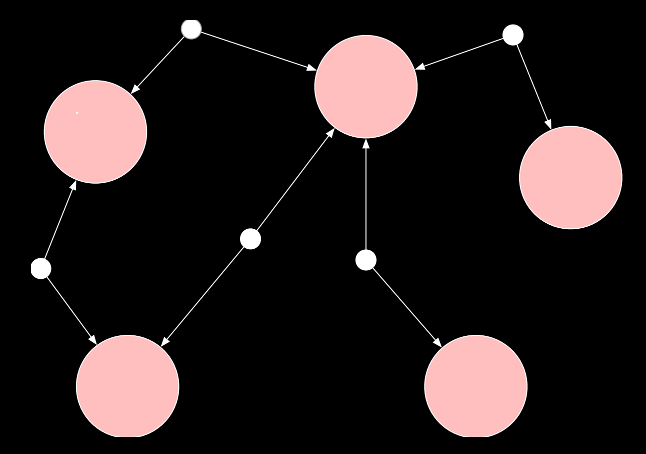


Recap

Structured decompositions represent compositional data.

& yield Grothendieck topologies

$$d\colon \int G \to \mathsf{C}$$

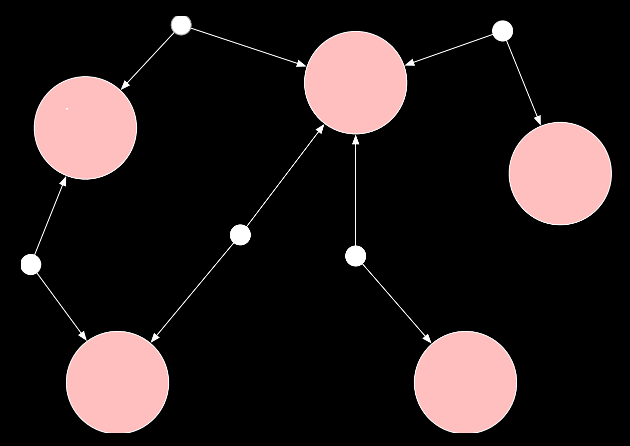


Recap

Structured decompositions represent compositional data.

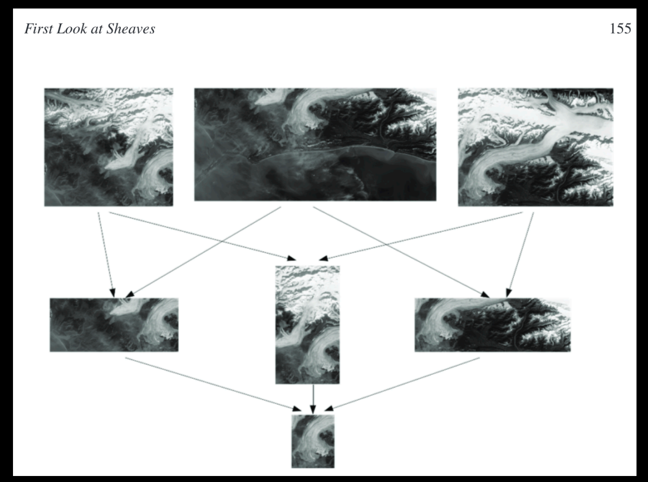
& yield Grothendieck topologies

$$d\colon \int G \to \mathsf{C}$$



Shaves represent compositional problems

$$\mathsf{C} \xrightarrow{\mathscr{F}} \mathsf{FinSet}^{op} \xrightarrow{\mathsf{dec}^{op}} \mathbf{2}^{op}$$

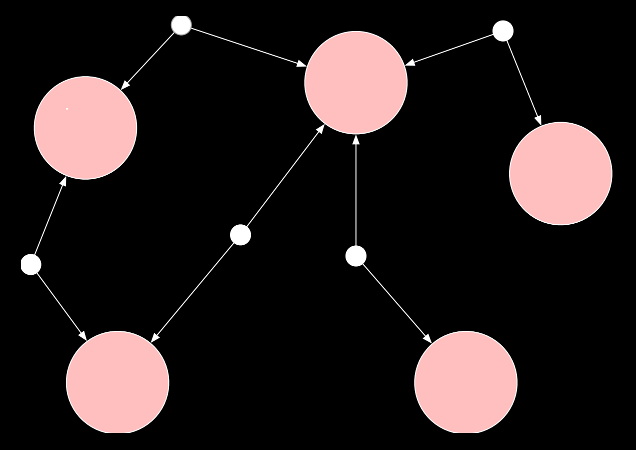


Recap

Structured decompositions represent compositional data.

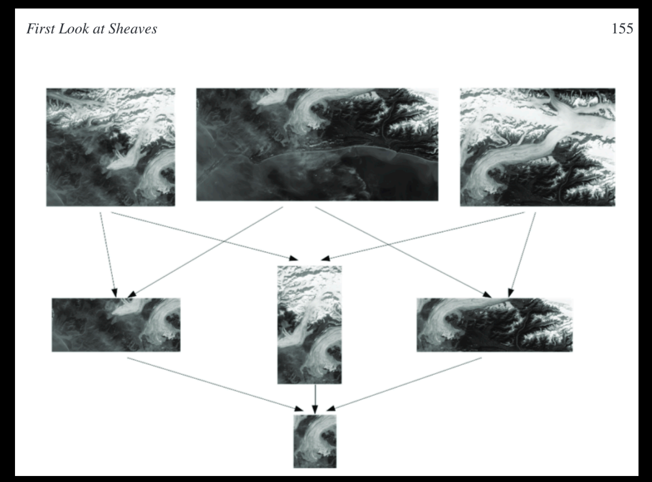
& yield Grothendieck topologies

$$d\colon \int G \to \mathsf{C}$$

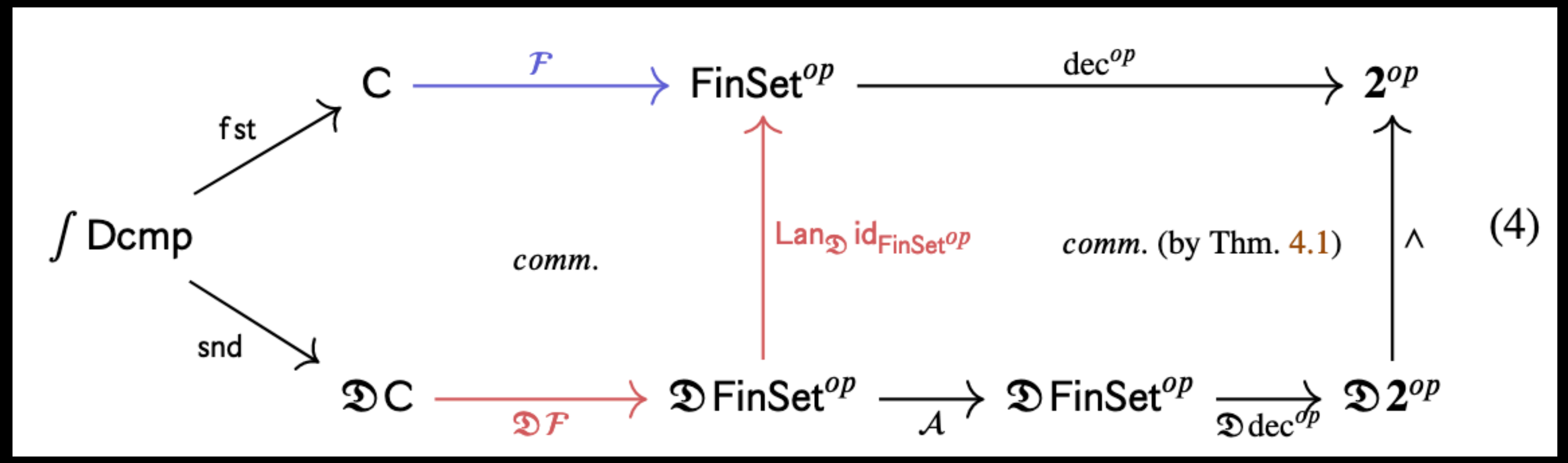


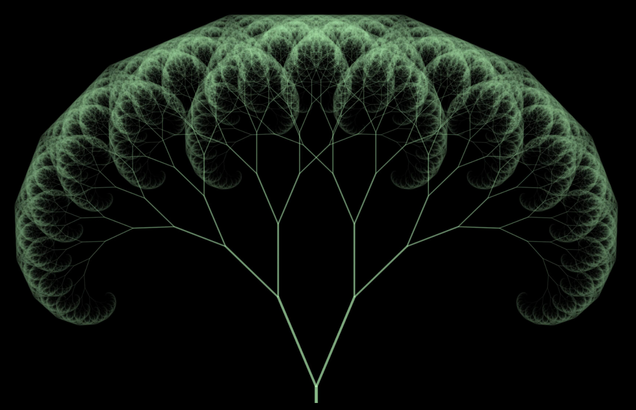
Shaves represent compositional problems

$$\mathsf{C} \xrightarrow{\mathscr{F}} \mathsf{FinSet}^{op} \xrightarrow{\mathsf{dec}^{op}} \mathbf{2}^{op}$$



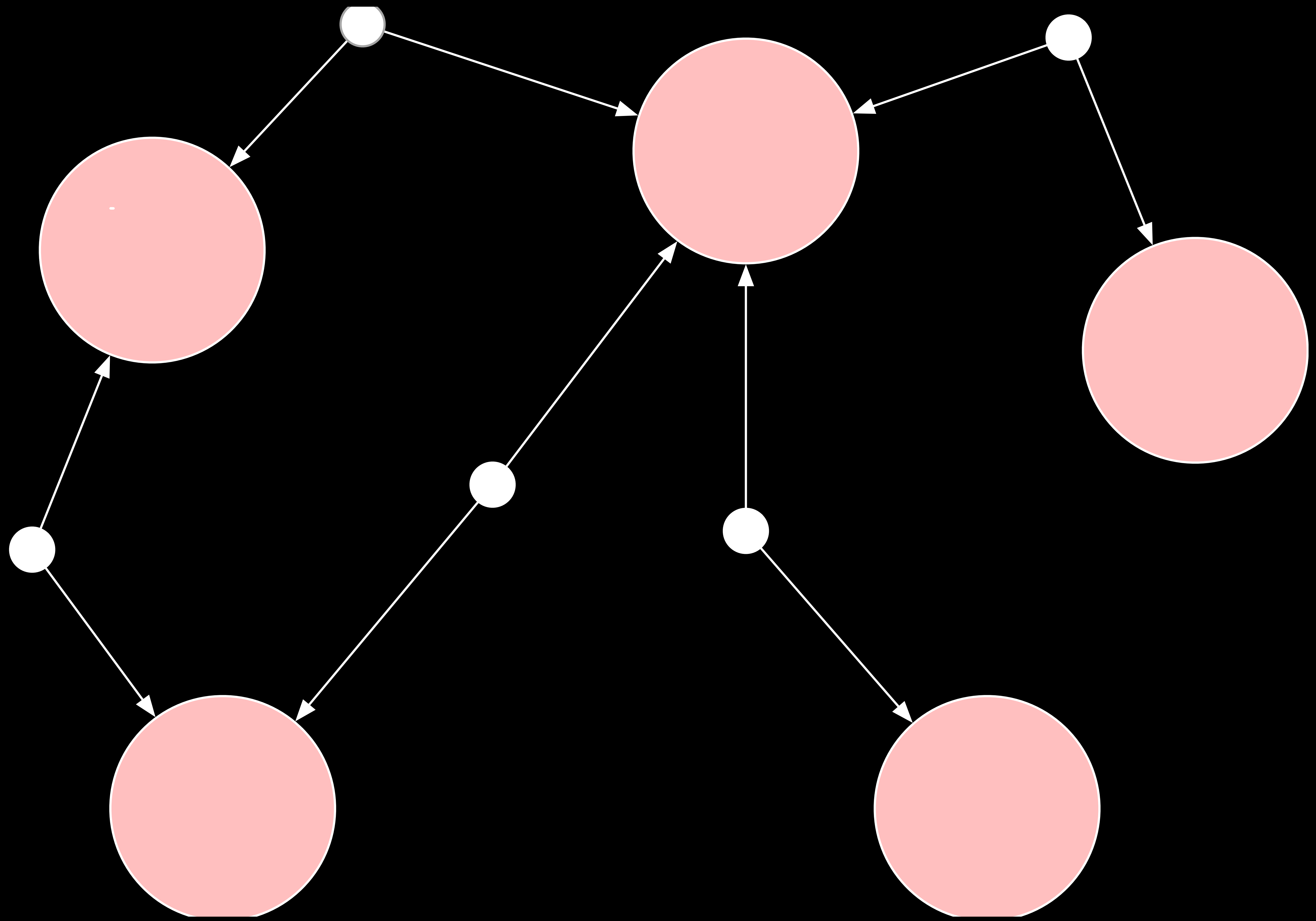
We obtain a fast dynamic programming algorithm for problems encoded as sheaves w.r.t. decomposition topology





What next?

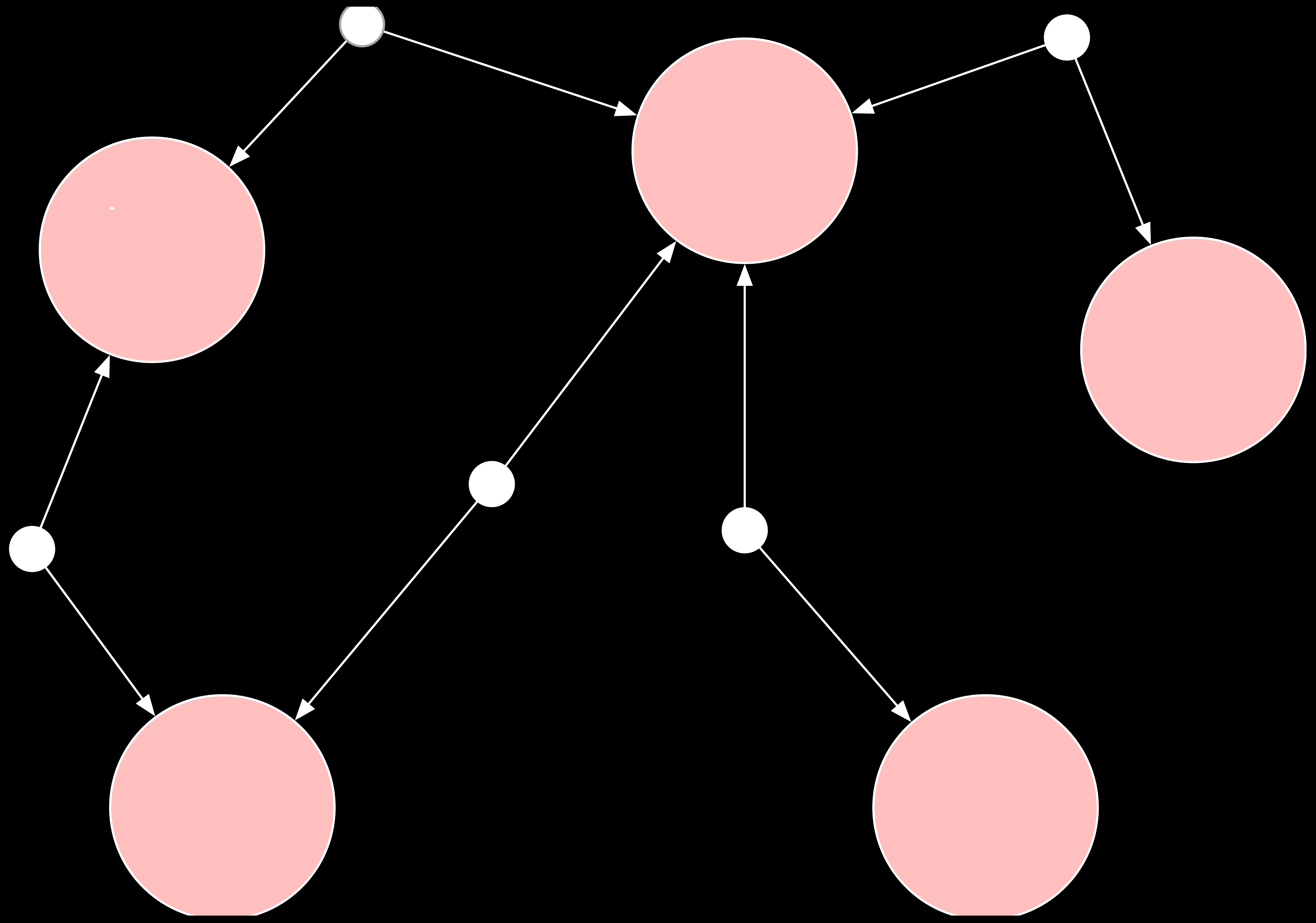
1. Structured decomps $d\colon \int G \to C$, but with G another kind of C-set?
2. Obstructions to categorial decompositions (tangles, brambles, etc.)
3. Applications of decompositions to other areas of math / CS?
4. How does this relate to your work?
5. Get in touch to get involved!



Thanks!

Blog: bmbumpus.com

Twitter: @BenMBumpus



Thanks!

Blog: bmbumpus.com

Twitter: @BenMBumpus