A Unifying Framework for Characterizing and Computing Width Measures

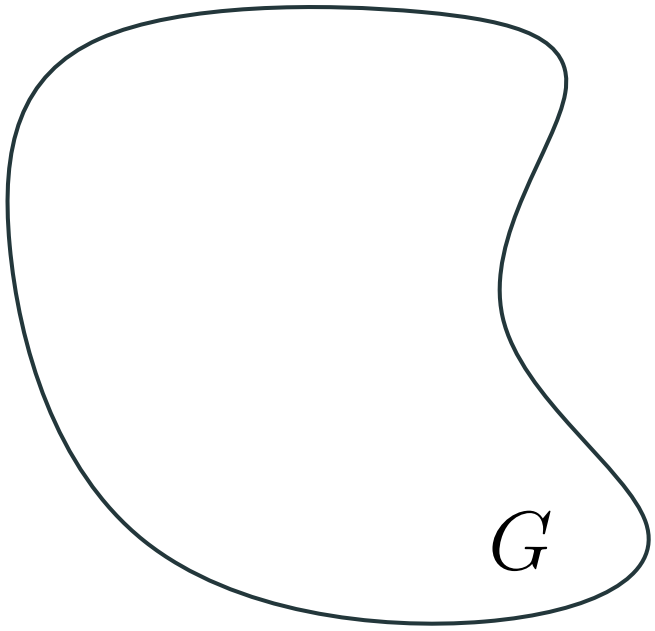
Robert Ganian

GRASTA 2022 (*based on a contribution to* ITCS 2022)

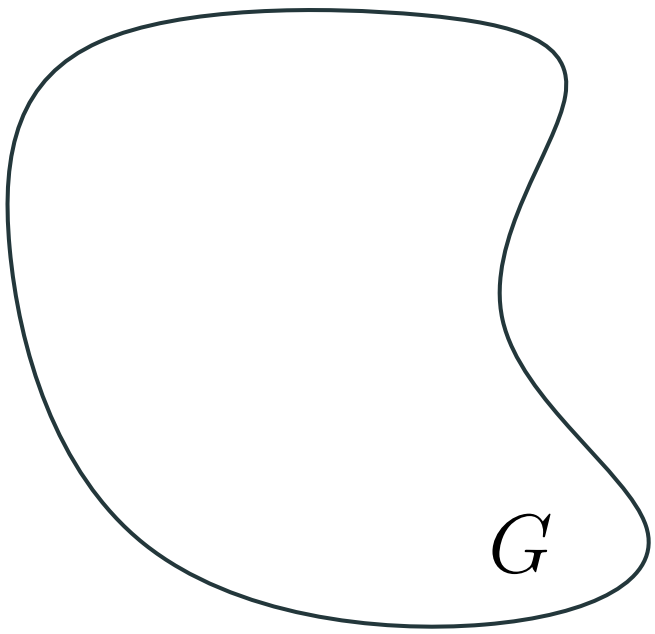
joint work with: Eduard Eiben, Thekla Hamm, *Lars Jaffke*, O-joung Kwon

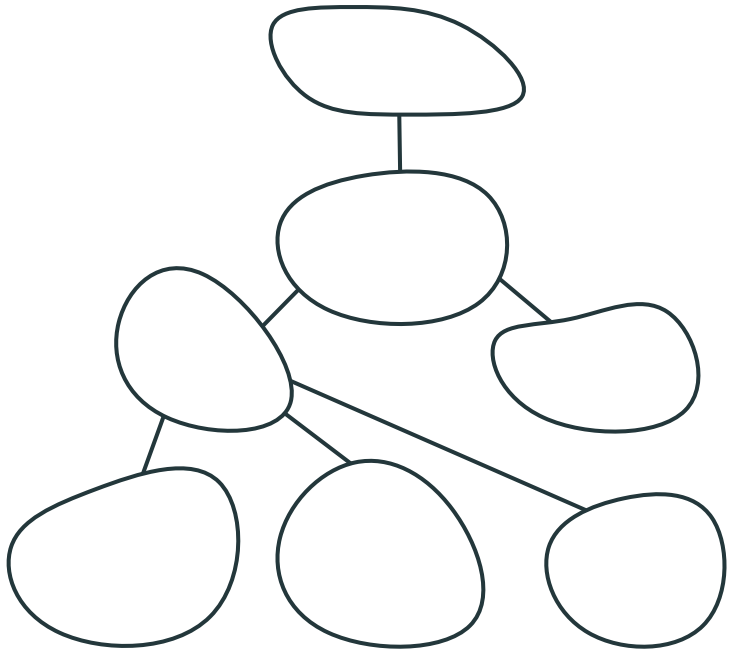
Example 1: Treewidth

Example 1: Treewidth

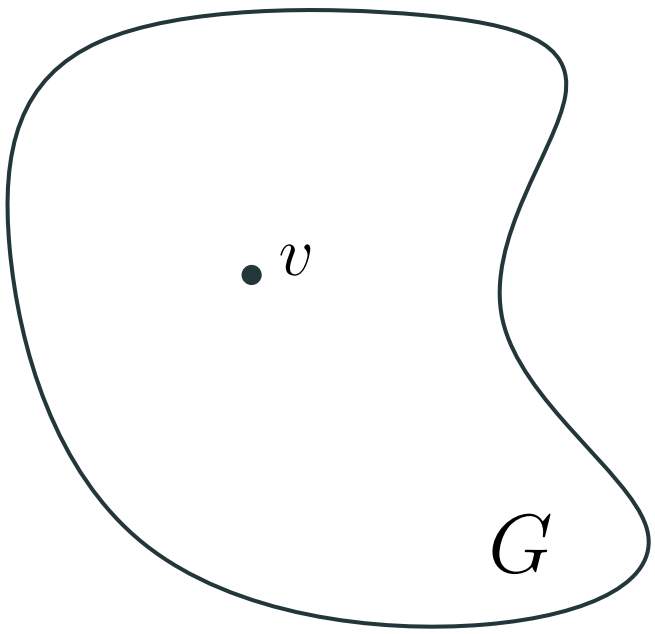


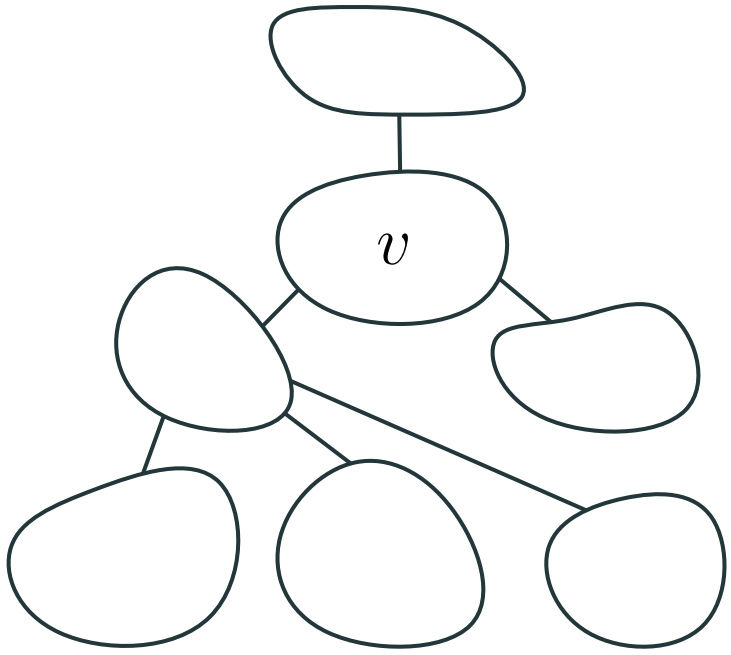
Example 1: Treewidth



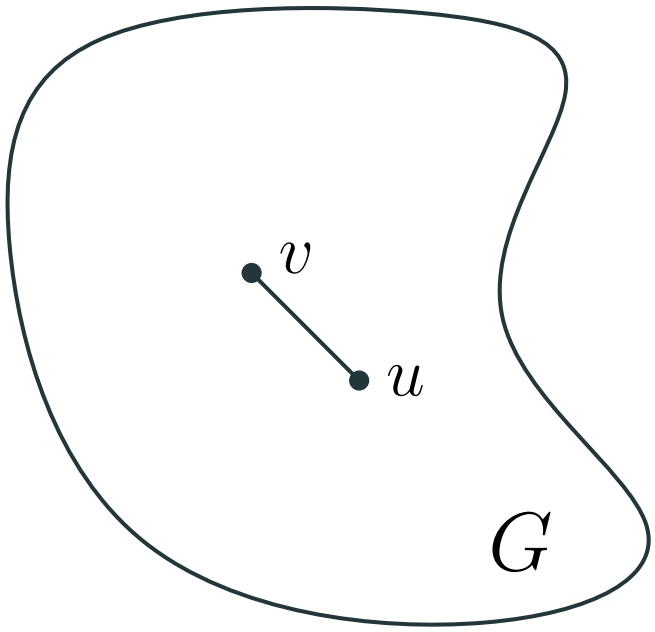


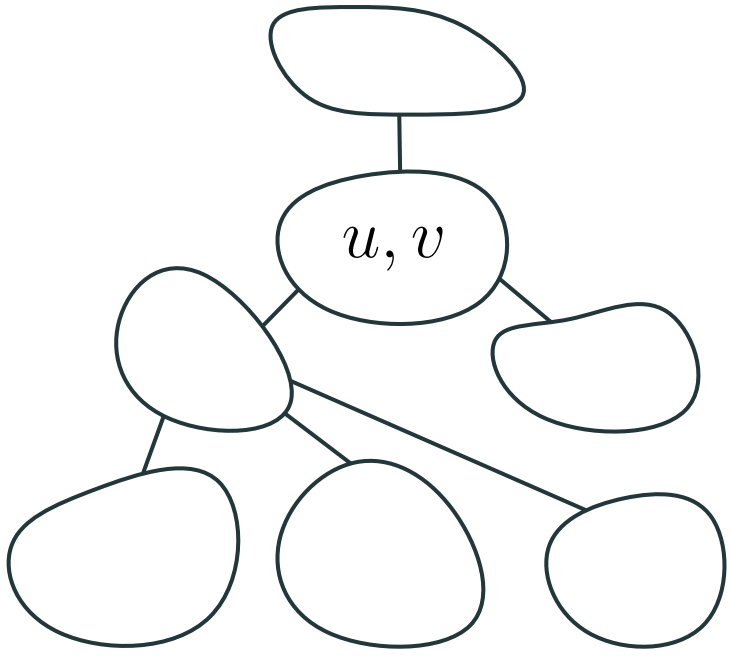
Example 1: Treewidth



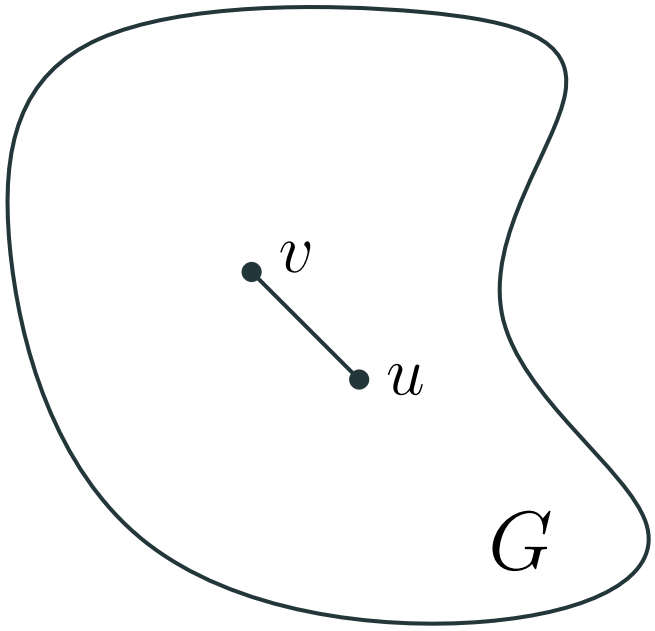


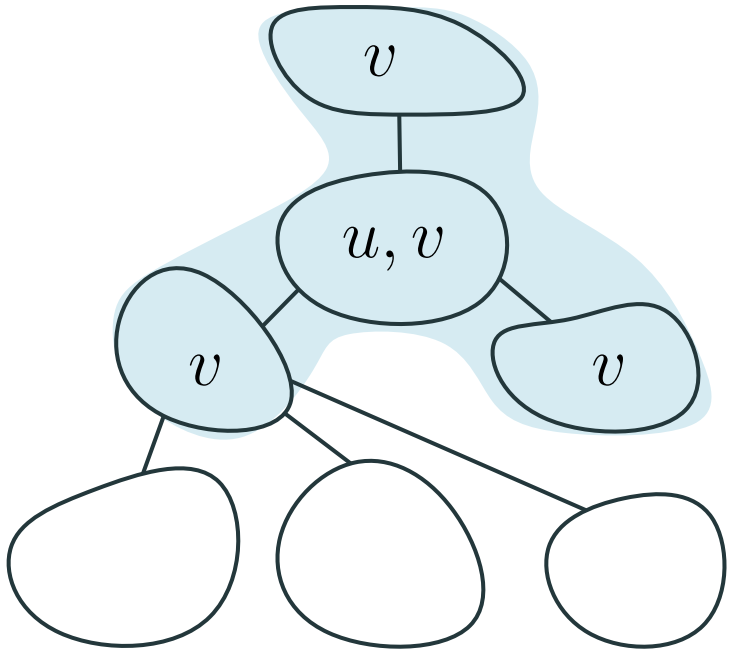
Example 1: Treewidth



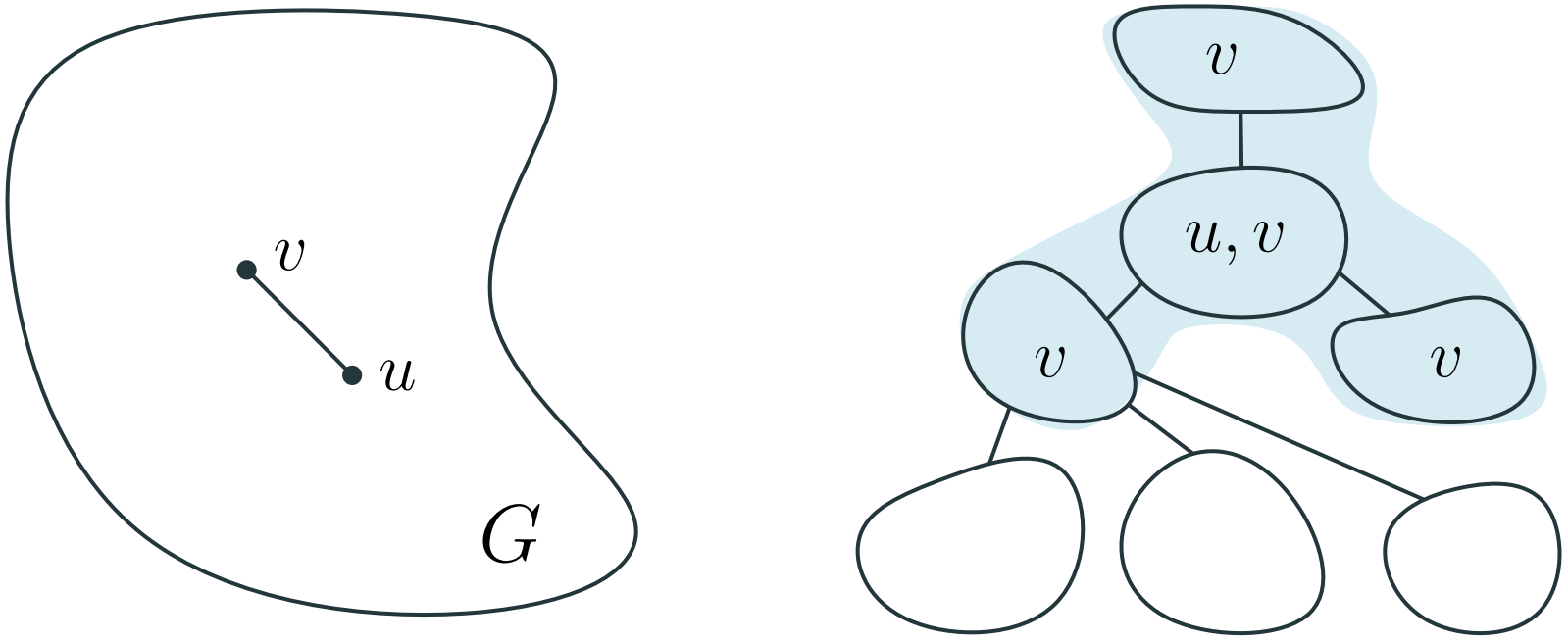


Example 1: Treewidth



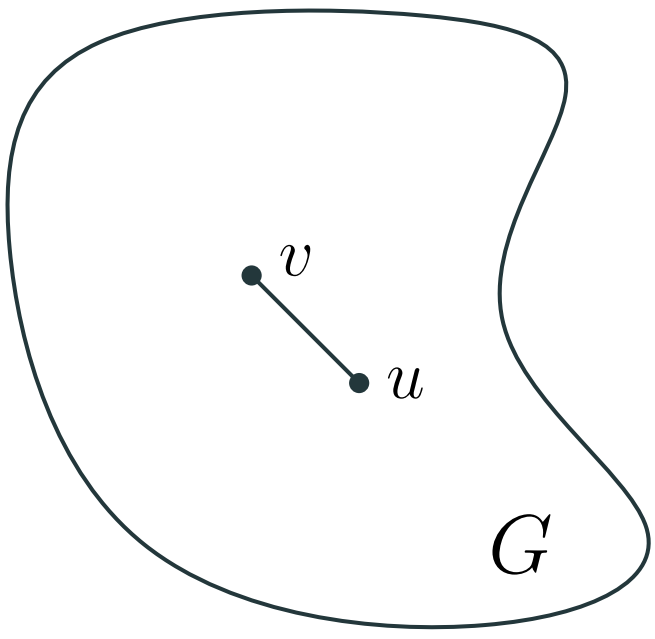


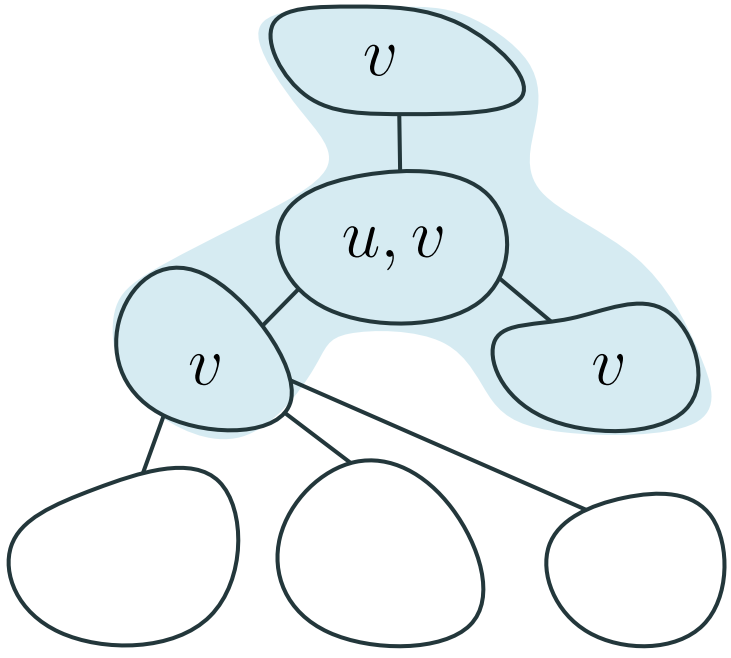
Example 1: Treewidth



Width of tree decomposition: max bag size -1.

Example 1: Treewidth





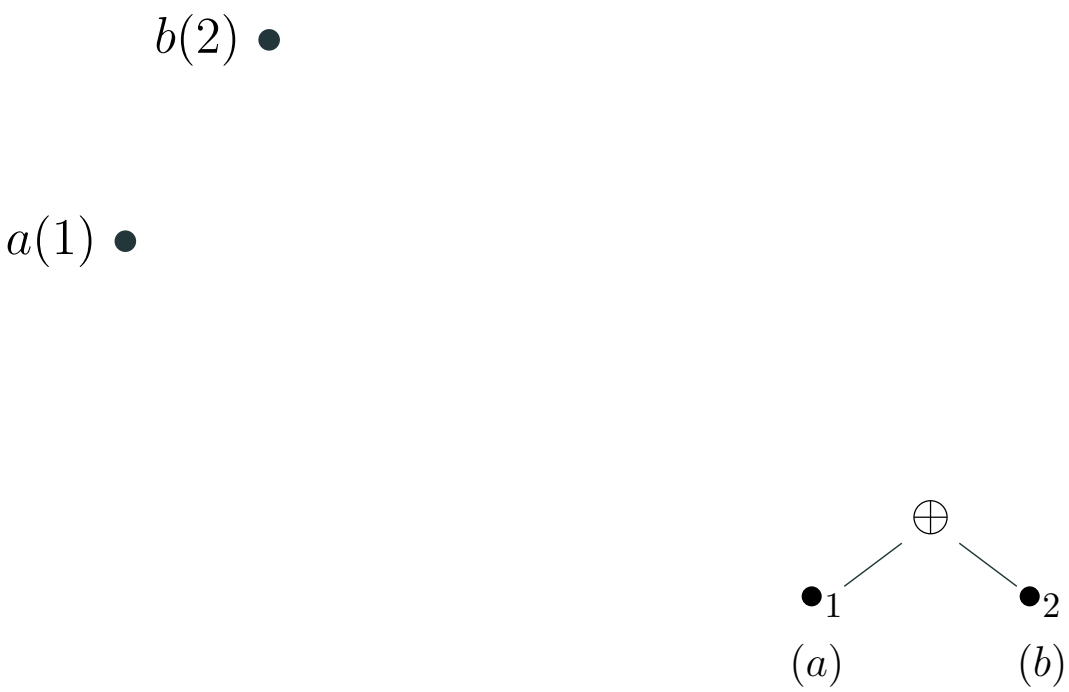
Width of tree decomposition: max bag size -1.

Treewidth of G: min width of any of its tree decompositions.

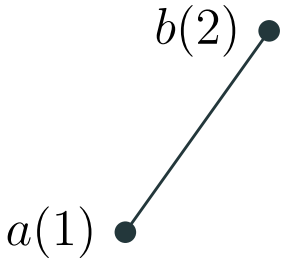
Example 2: Clique-width

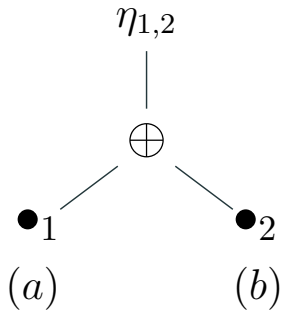
Example 2: Clique-width

Example 2: Clique-width

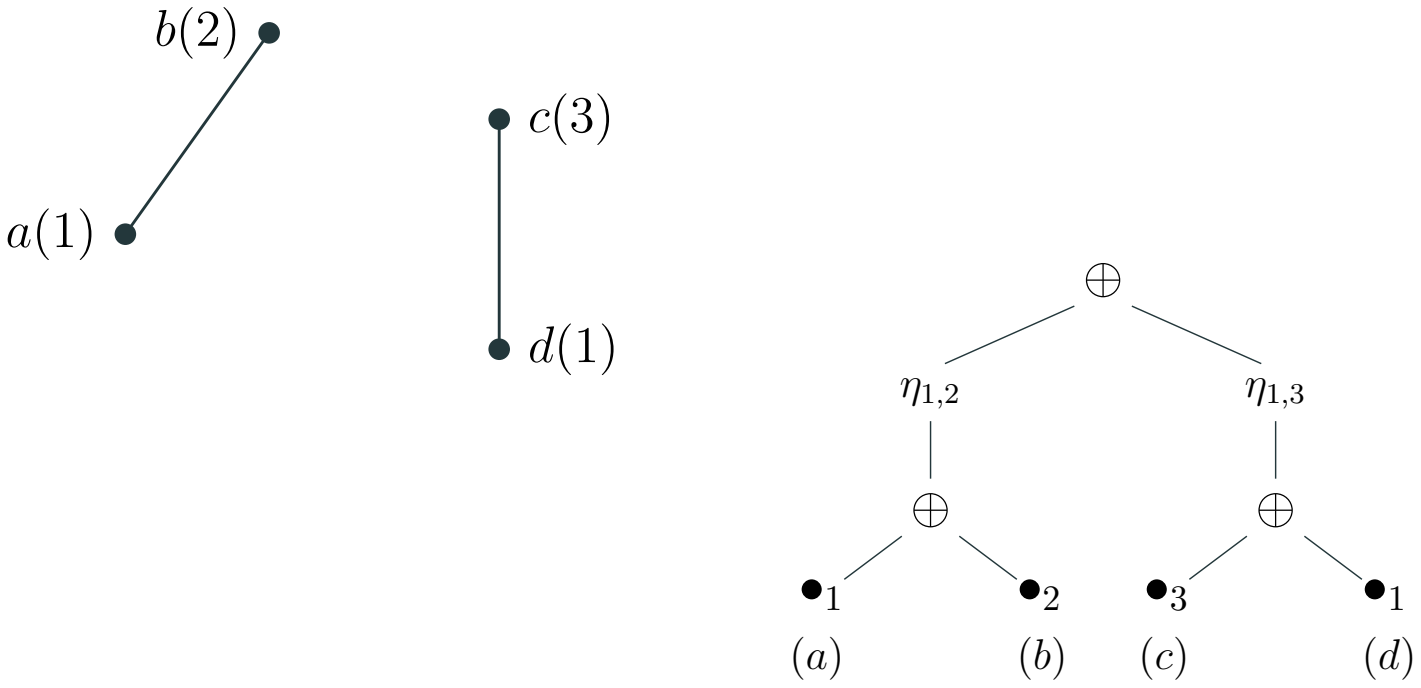


Example 2: Clique-width

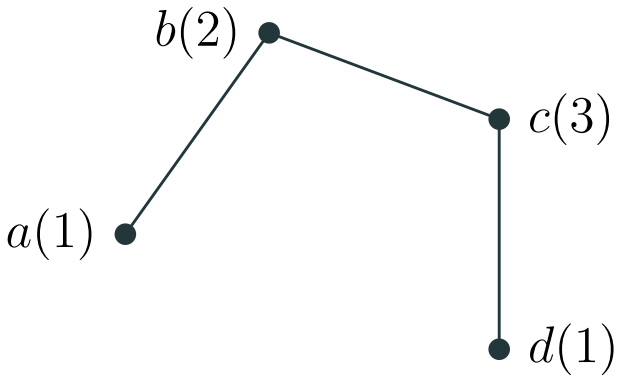


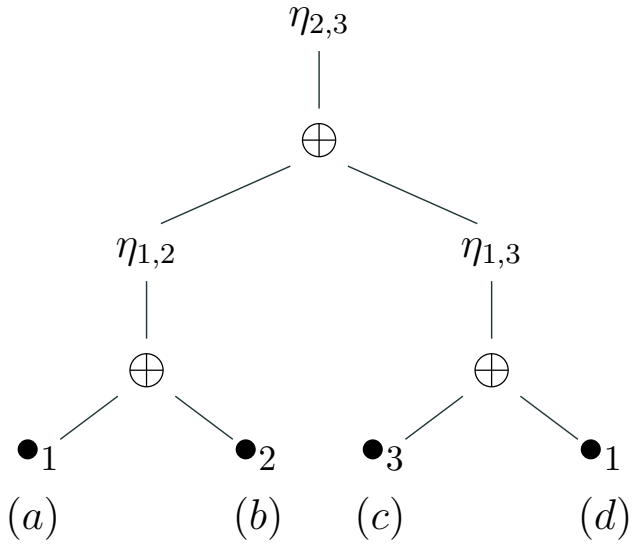


Example 2: Clique-width

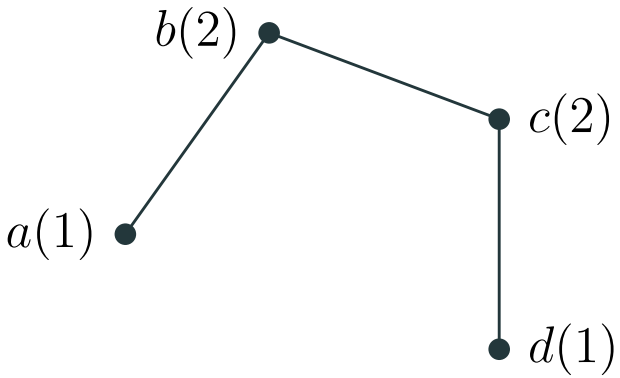


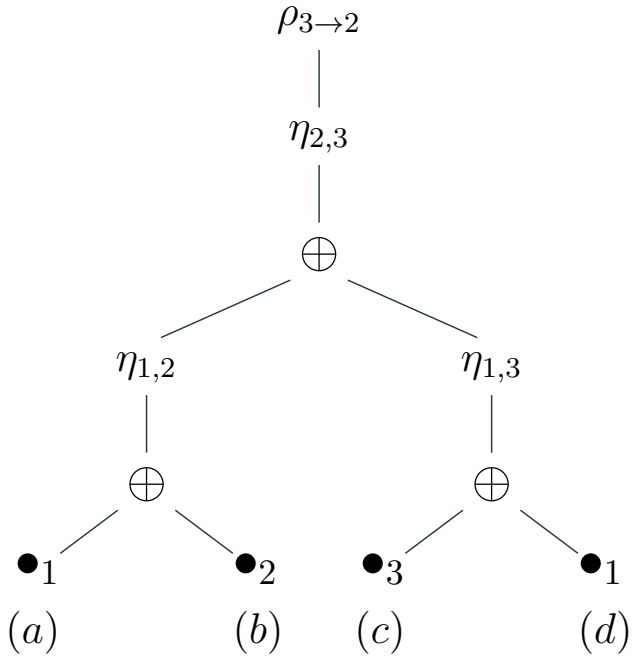
Example 2: Clique-width



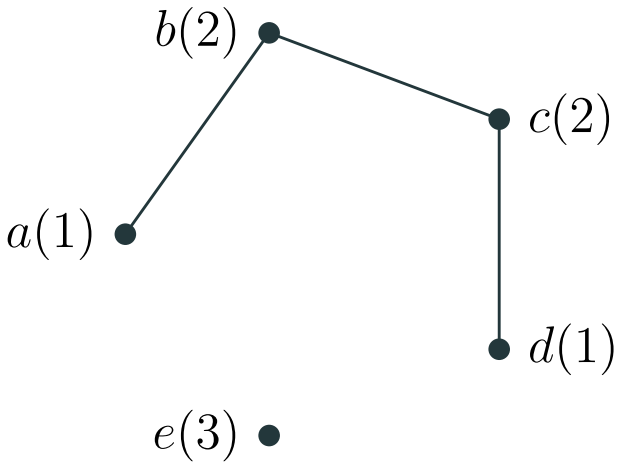


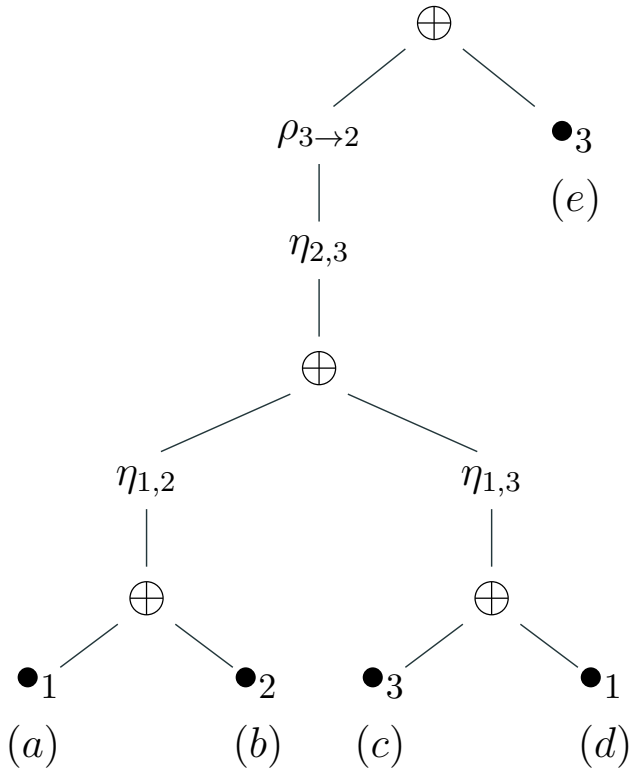
Example 2: Clique-width



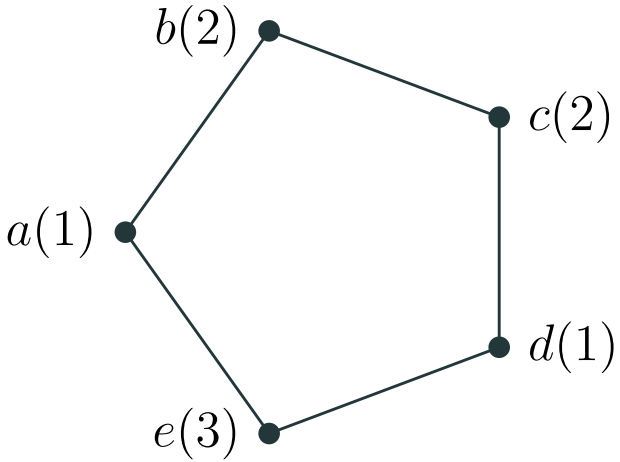


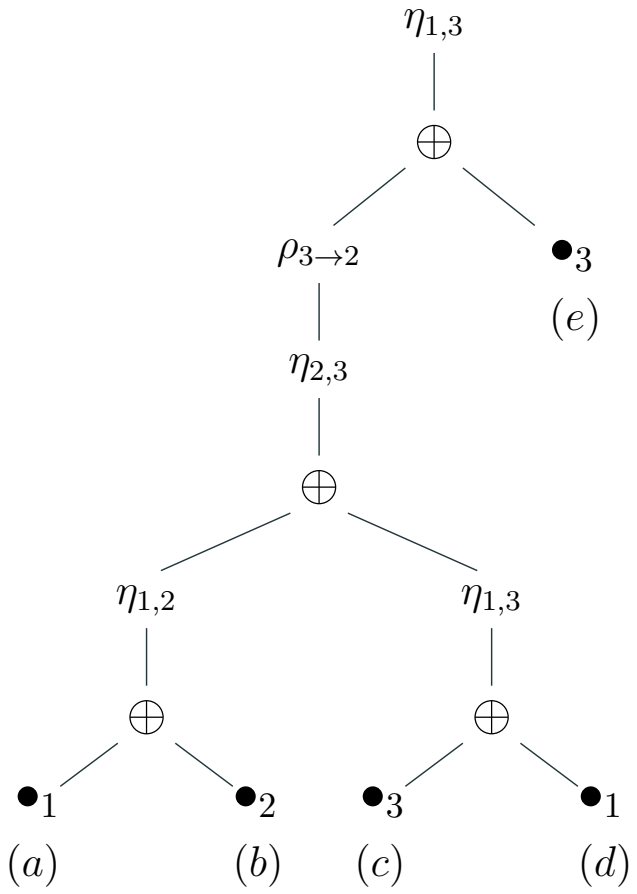
Example 2: Clique-width



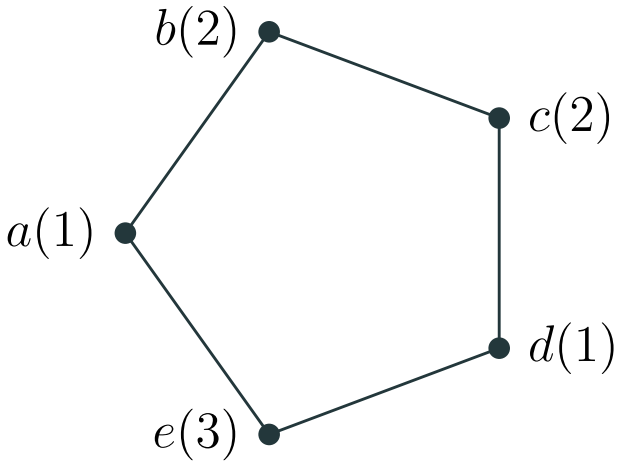


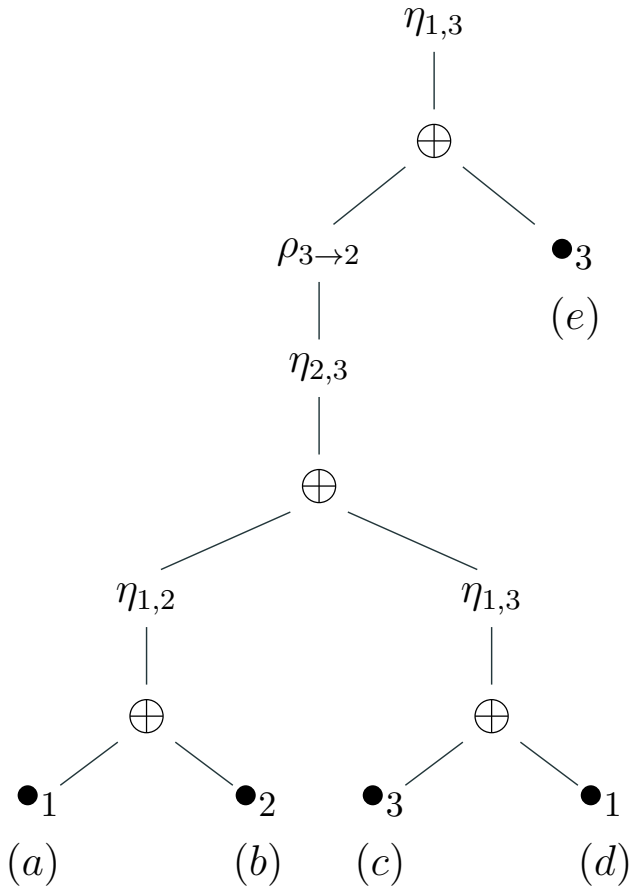
Example 2: Clique-width





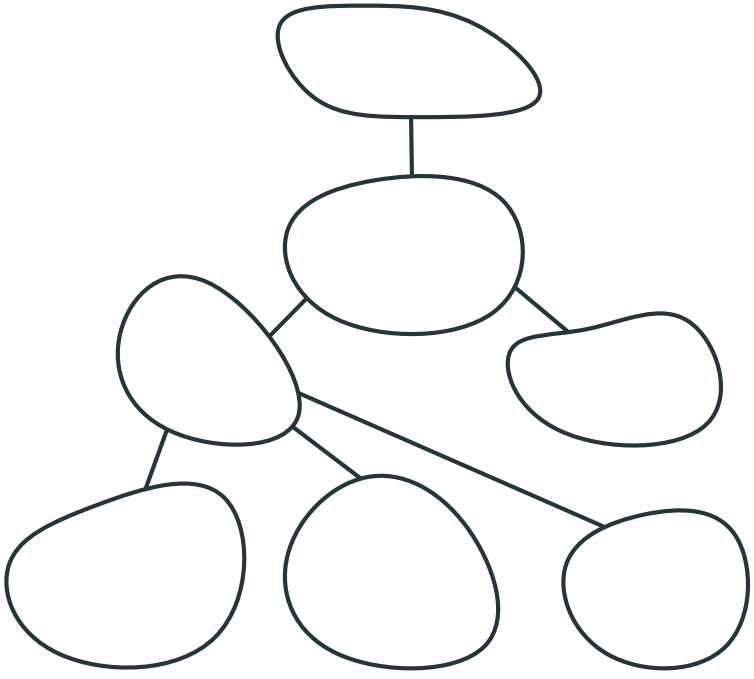
Example 2: Clique-width

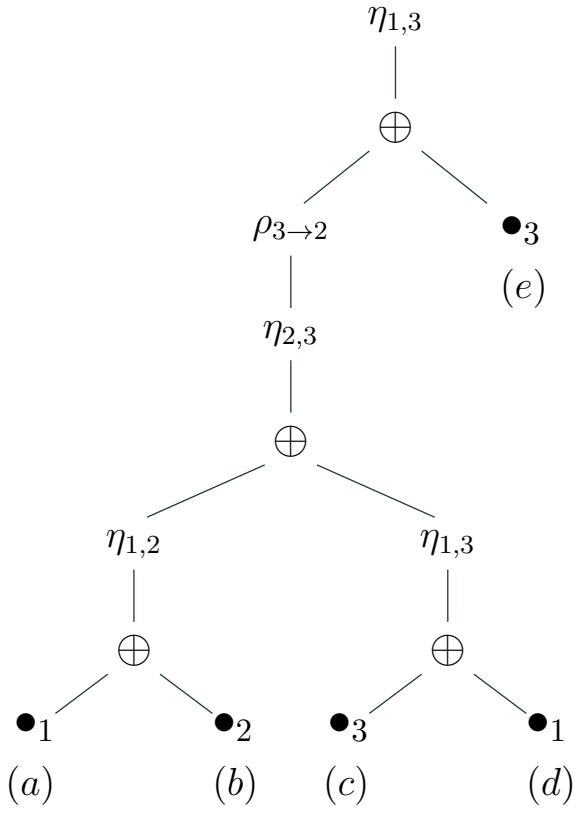




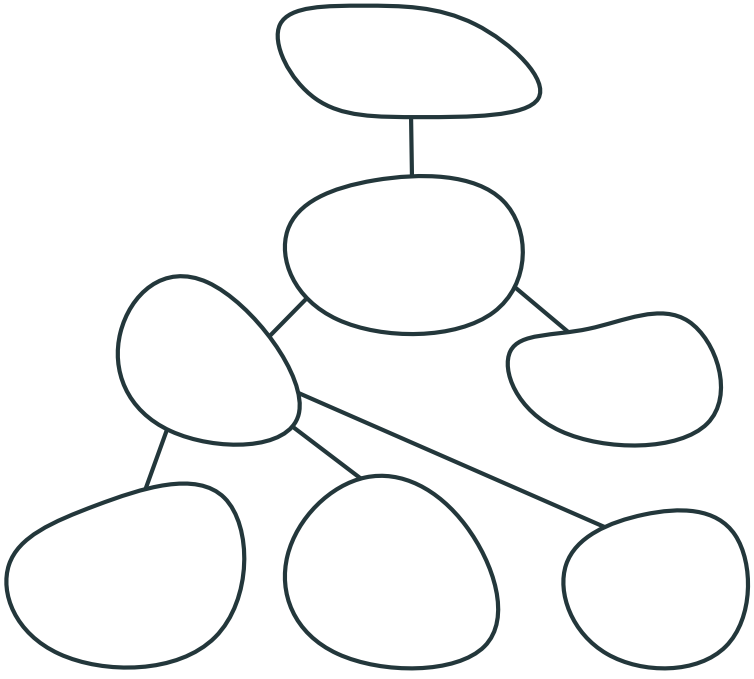
Width: number of labels

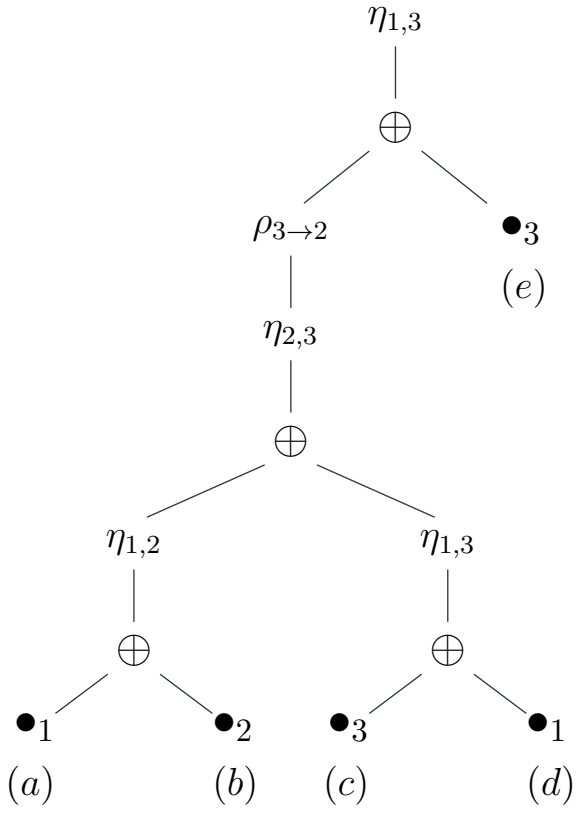
Algorithmic use of small-width decompositions





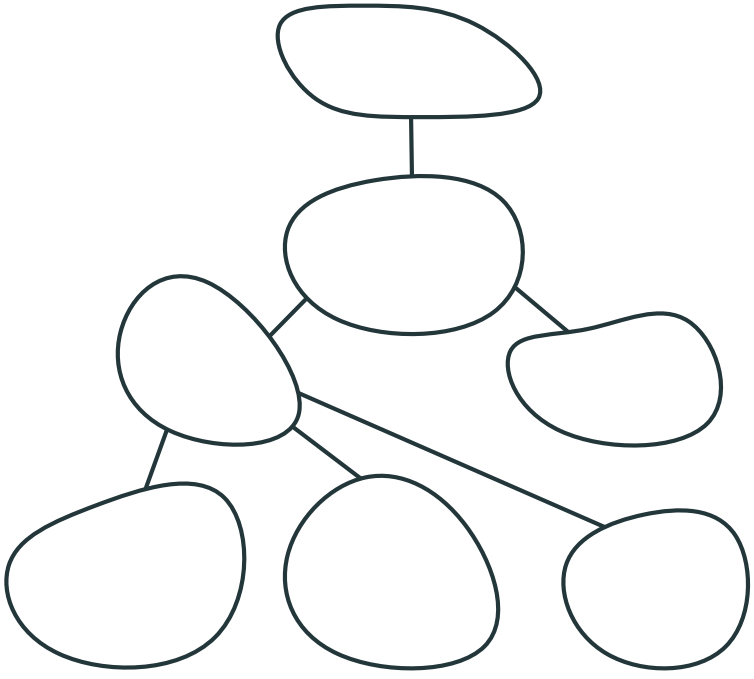
Algorithmic use of small-width decompositions

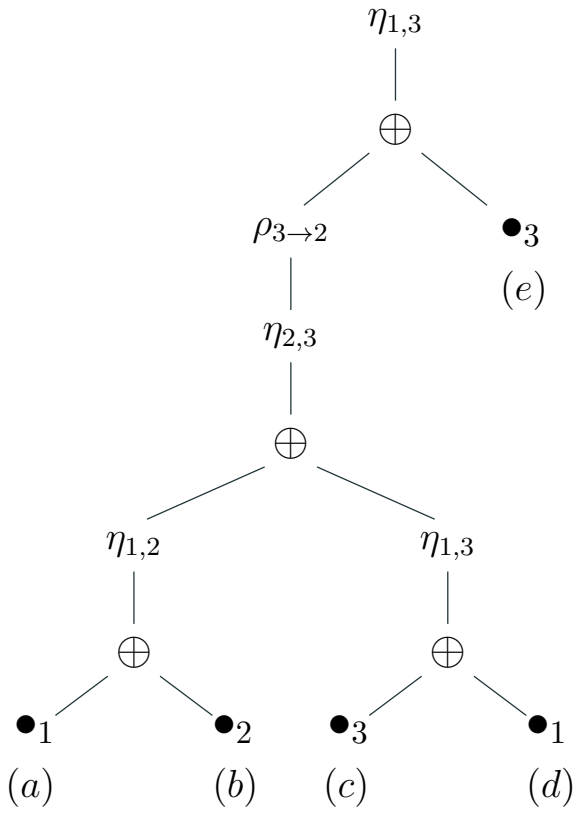




Dynamic programming over tree-structure

Algorithmic use of small-width decompositions





Dynamic programming over tree-structure;

polynomial time if width is constant.

(Some) width measures and their expressive power

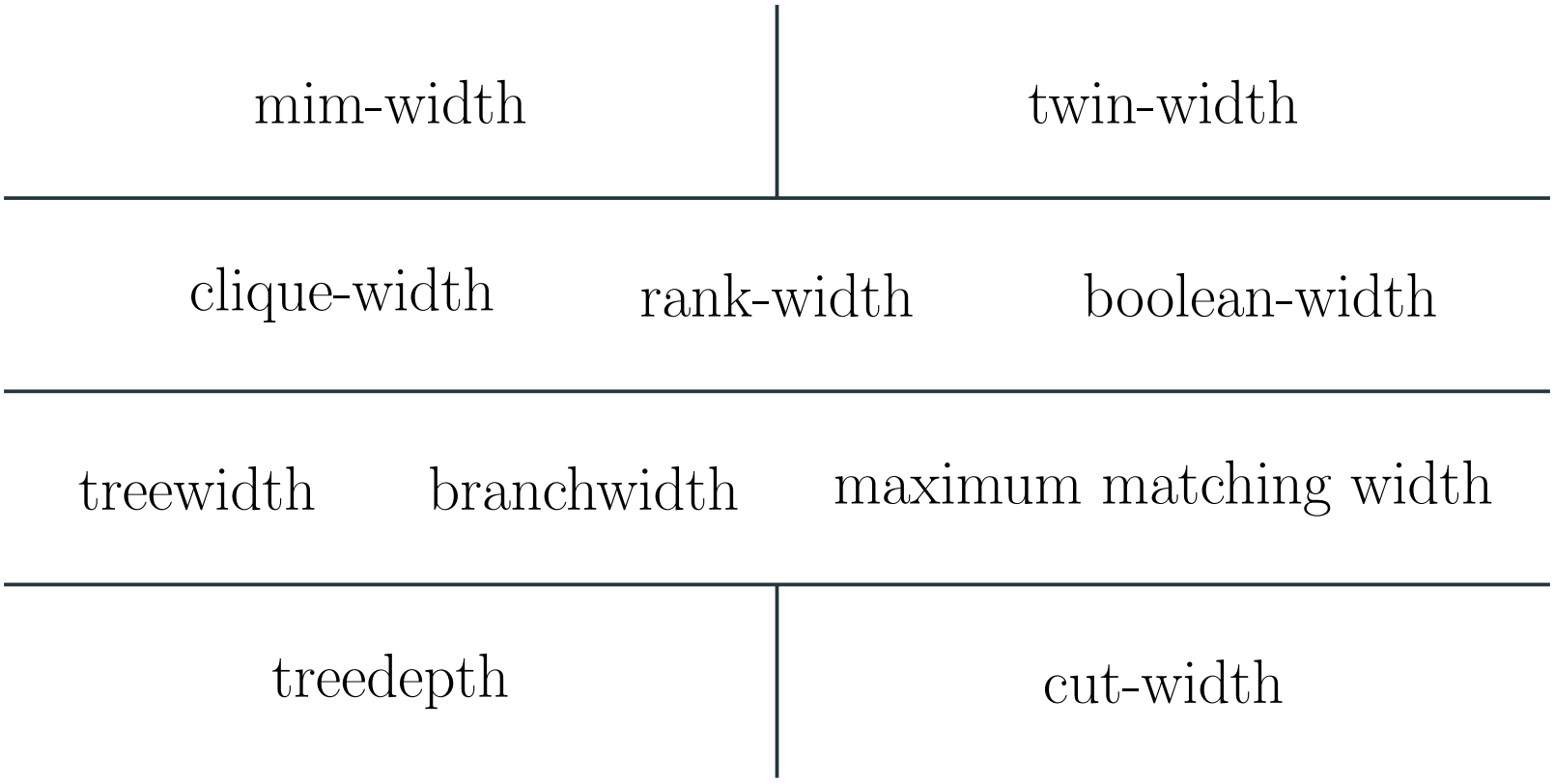
mim-width		twin-width
clique-width	rank-width	boolean-width
treewidth	branchwidth	maximum matching width
treedepth		cut-width

(Some) width measures and their expressive power

mim-width		twin-width
clique-width	rank-width	boolean-width
treewidth	branchwidth	maximum matching width
treedepth		cut-width

Width w_1 asymptotically dominates width w_2 if for all G,
$w_1(G) \leq f(w_2(G))$ for some f.

(Some) width measures and their expressive power



Width w_1 asymptotically dominates width w_2 if for all G, $w_1(G) \leq f(w_2(G))$ for some f. They are asymptotically equivalent if they dominate each other.

...Mim-WHAT??

mim-width	twin-width

clique-width rank-width boolean-width

...

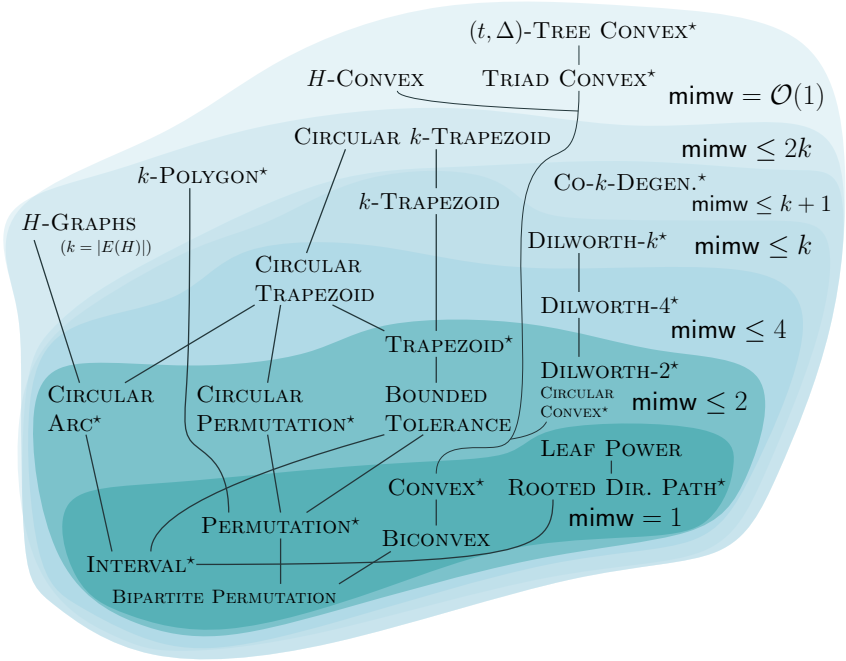
...Mim-WHAT??

mim-width	twin-width

clique-width rank-width boolean-width

...

High expressive power



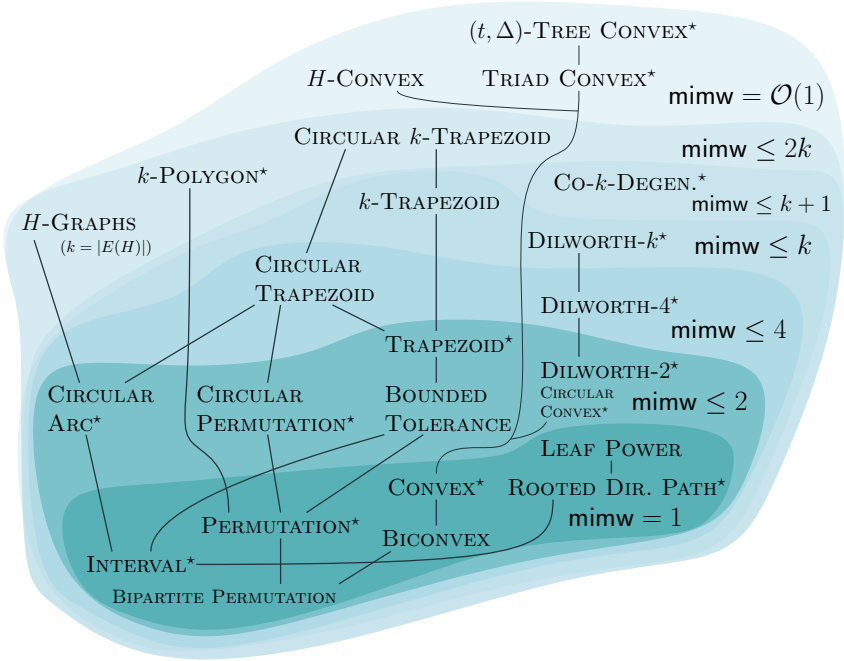
...Mim-WHAT??

mim-width | twin-width

clique-width rank-width boolean-width

. . .

High expressive power



Algorithmic applications

- Independent Set, Dominating Set, and many variants.
- H-Homomorphism, H-Covering, Odd Cycle Transversal, ...
- Feedback Vertex Set, Connected DS, Connected Vertex Cover...
- ...

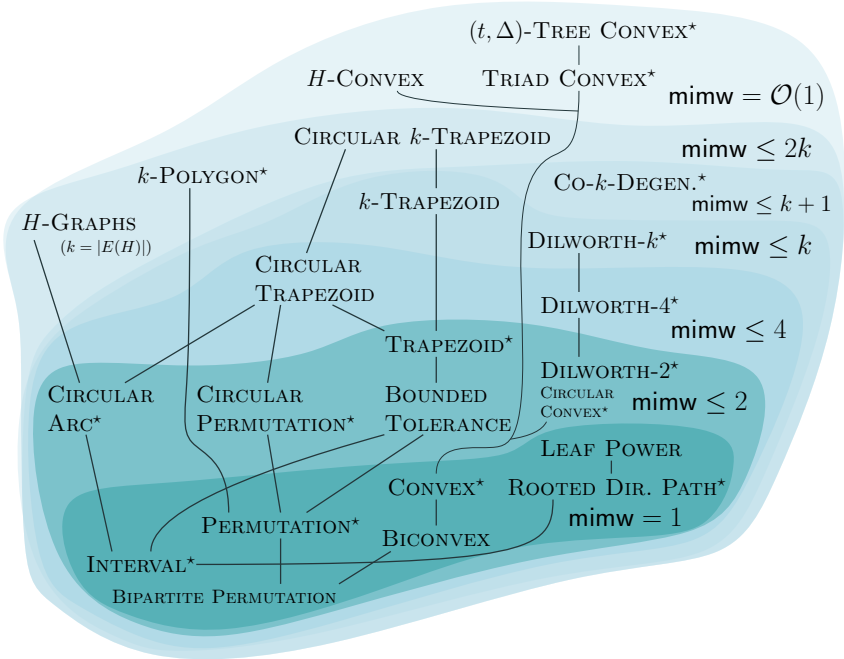
...Mim-WHAT??

mim-width	twin-width

clique-width rank-width boolean-width

...

High expressive power



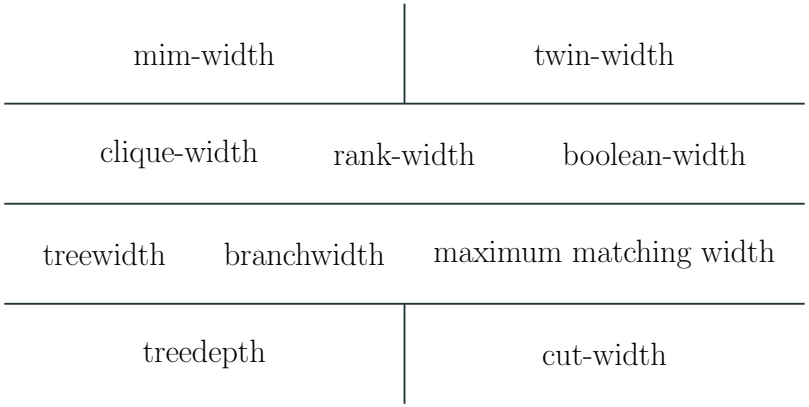
Algorithmic applications

- Independent Set, Dominating Set, and many variants.
- H-Homomorphism, H-Covering, Odd Cycle Transversal, ...
- Feedback Vertex Set, Connected DS, Connected Vertex Cover...
- ...

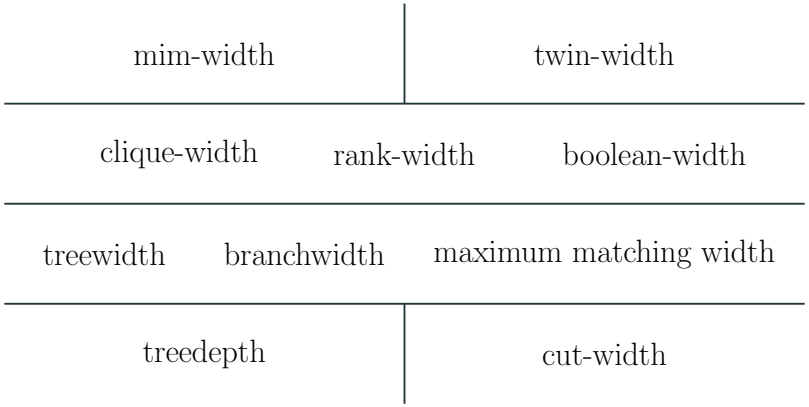
Open problem:

approximating mim-width

(Some) width measures and their expressive power II

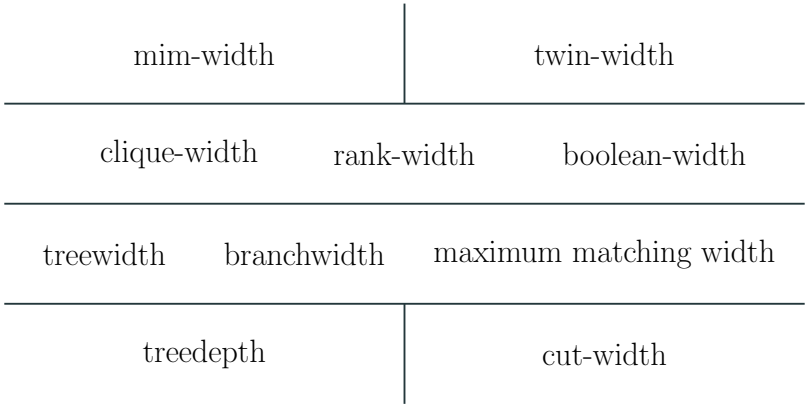


(Some) width measures and their expressive power II



What does width even mean in this context?

(Some) width measures and their expressive power II

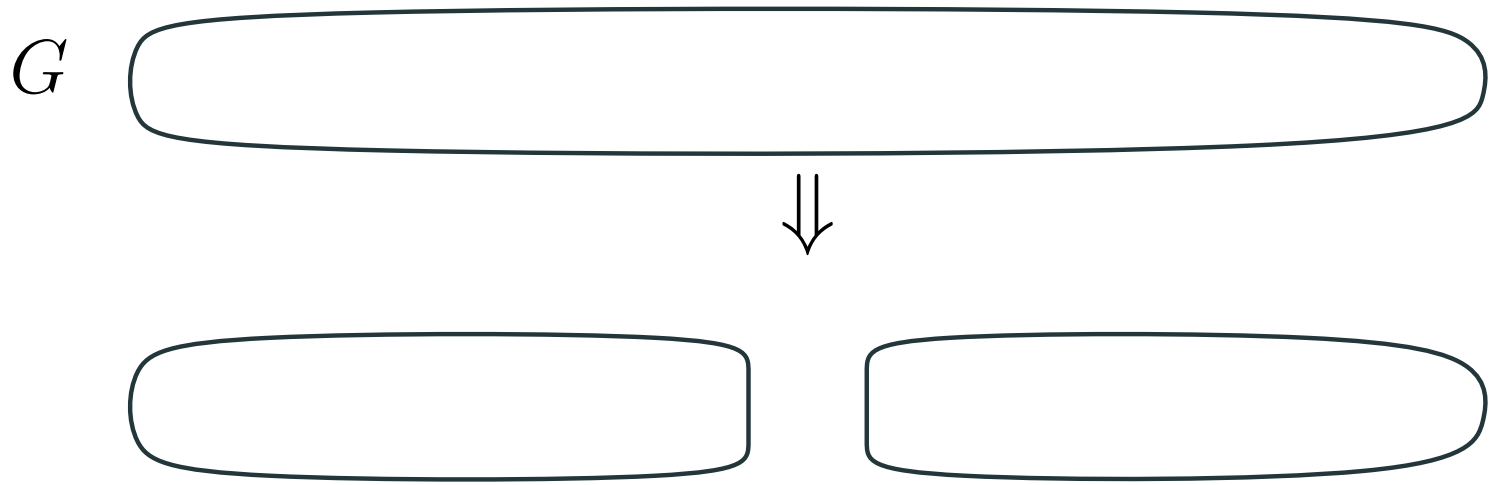


What does width even mean in this context?

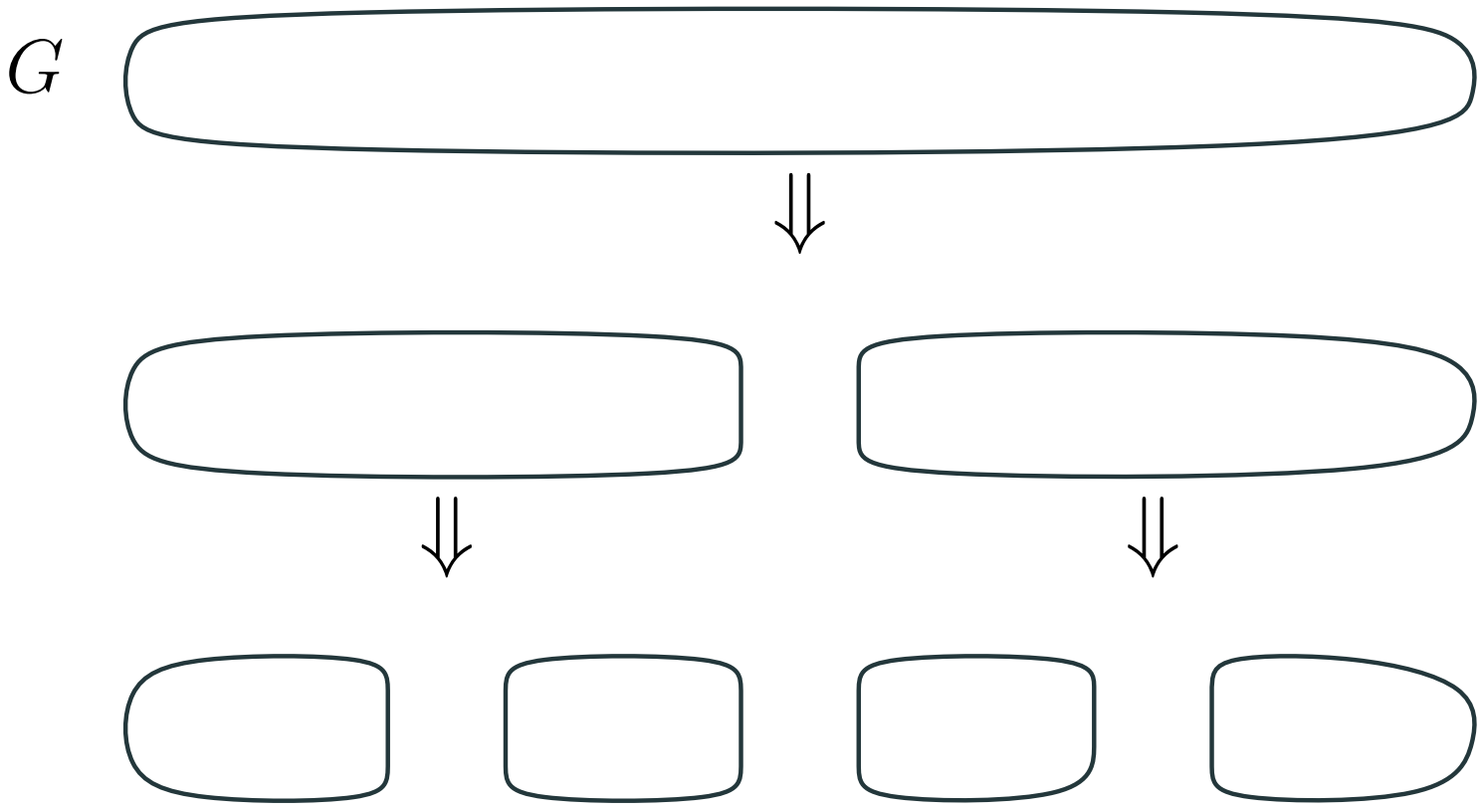
Search for unifying theories.

Step 1: Unifying the decomposition method

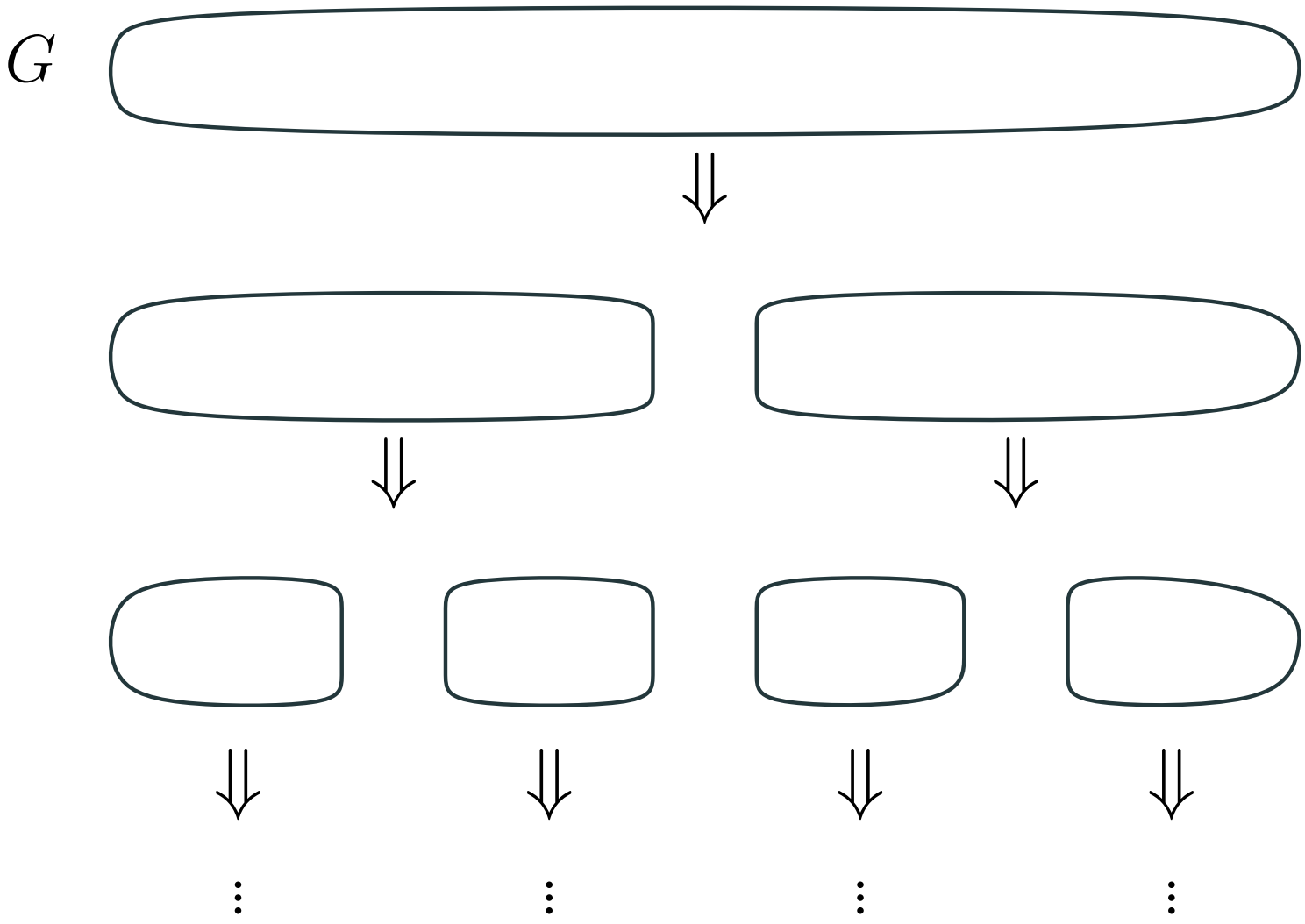
Step 1: Unifying the decomposition method



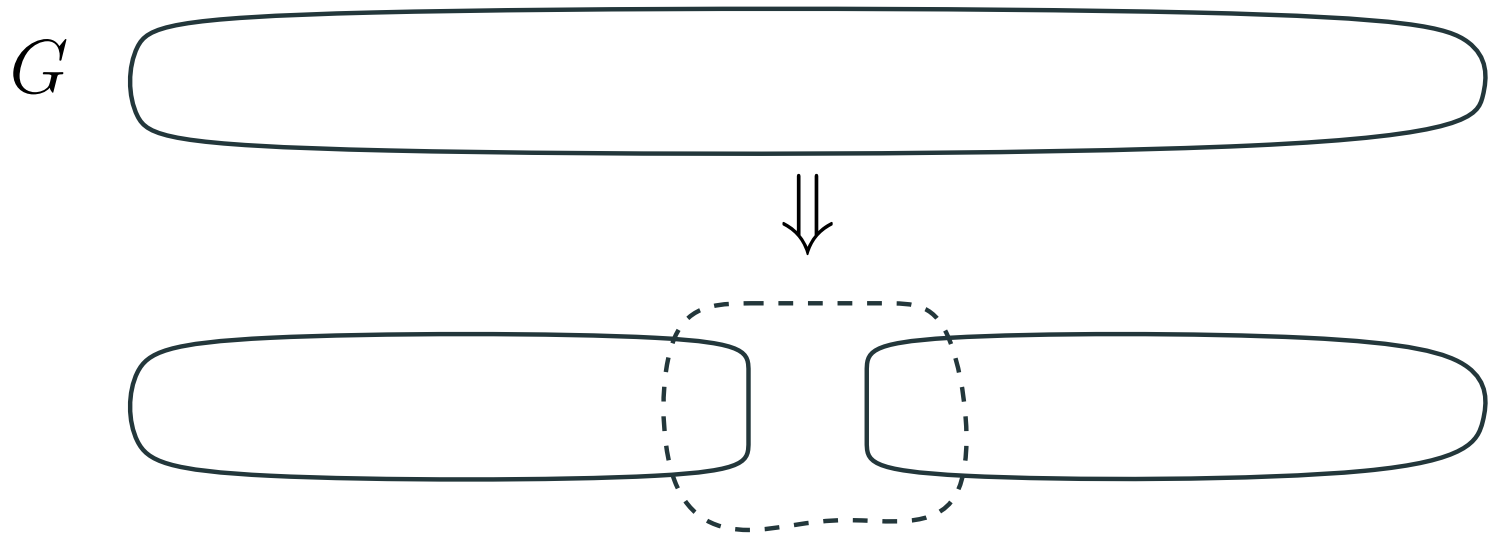
Step 1: Unifying the decomposition method



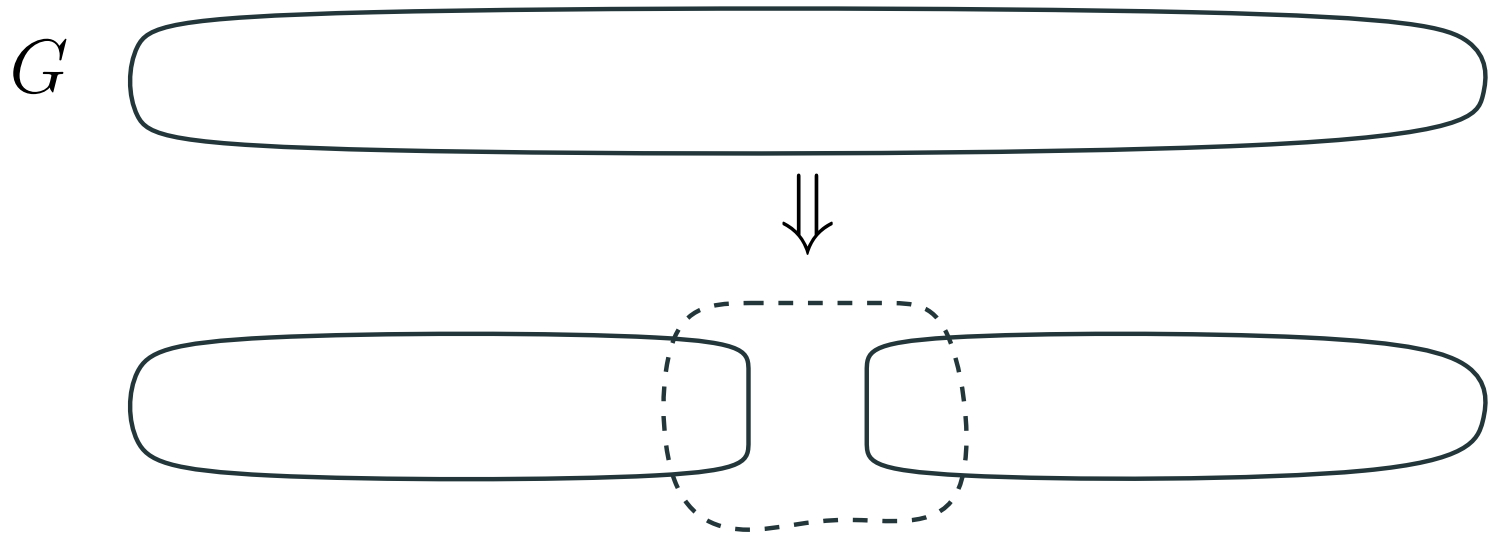
Step 1: Unifying the decomposition method



Step 1: Unifying the decomposition method

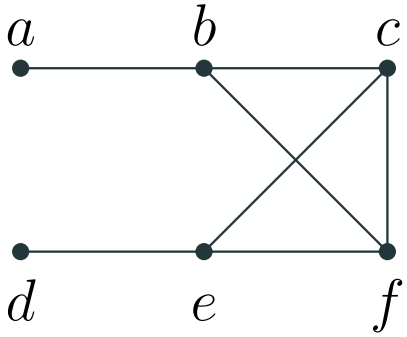


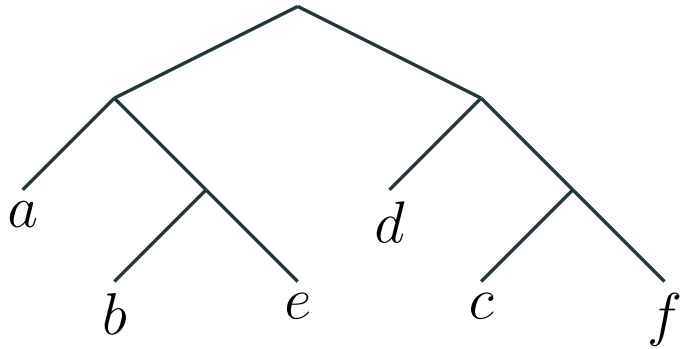
Step 1: Unifying the decomposition method



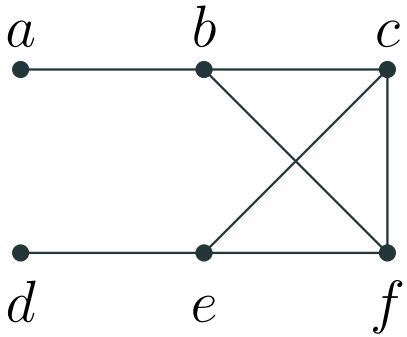
Width: Complexity of the cuts appearing during the decomposition process.

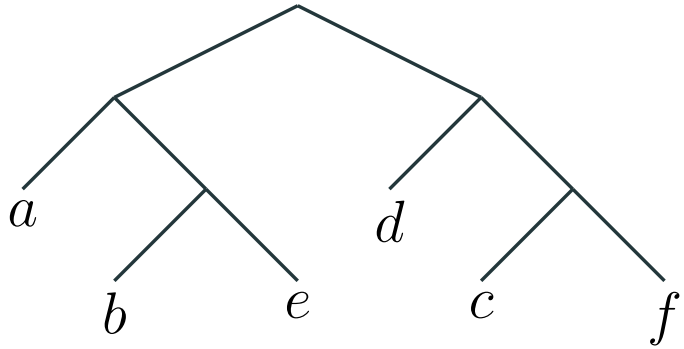
Step 1: Branch decompositions





Step 1: Branch decompositions

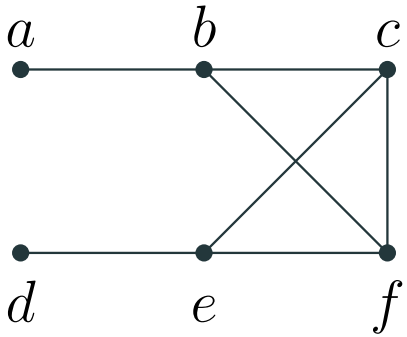


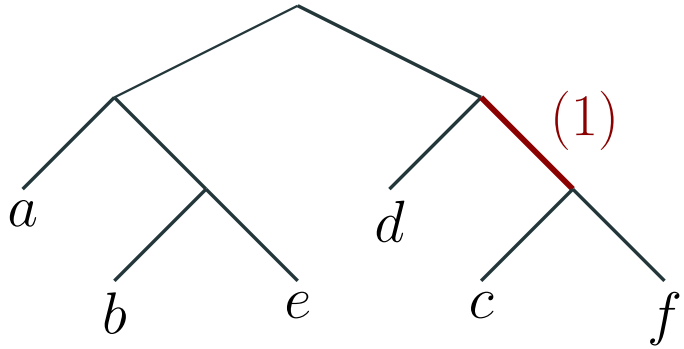


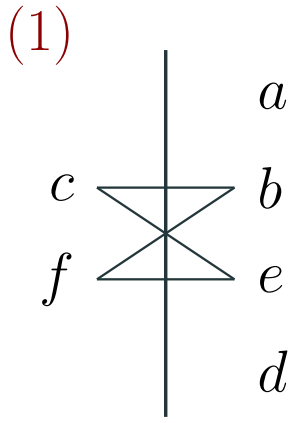
Maximum matching width ($\equiv$ treewidth):

Max. matching size across the cut.

Step 1: Branch decompositions



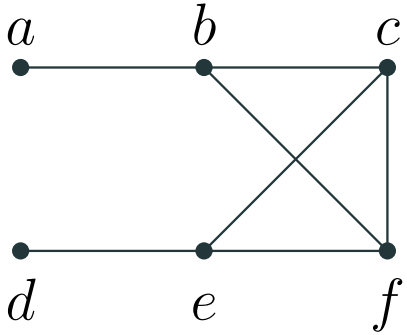


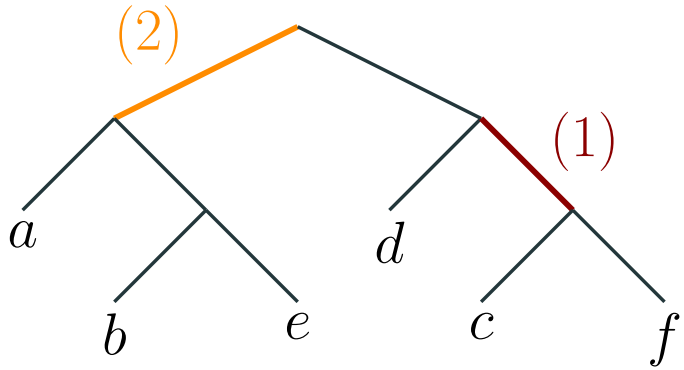


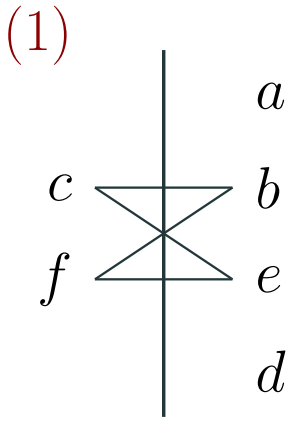
Maximum matching width ($\equiv$ treewidth):

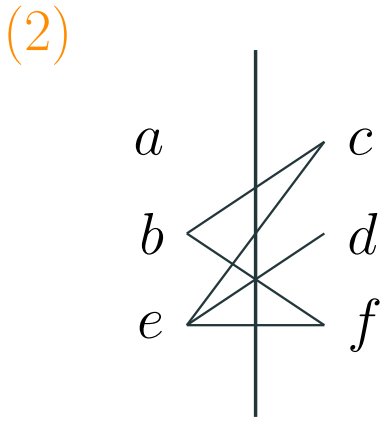
Max. matching size across the cut.

Step 1: Branch decompositions





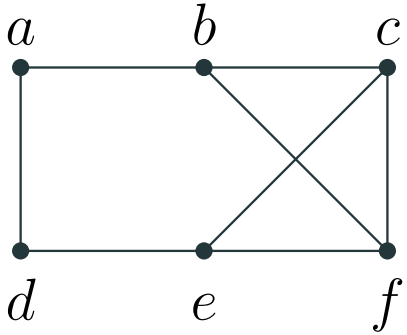


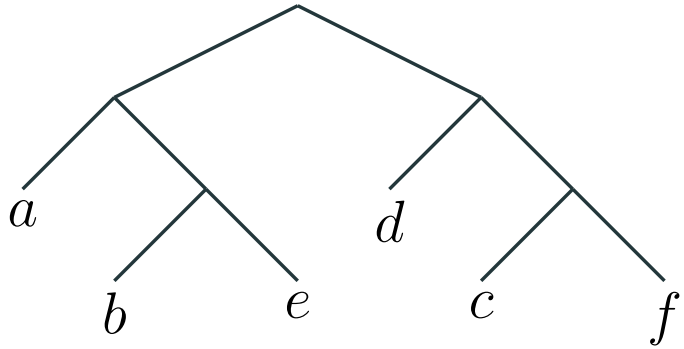


Maximum matching width ($\equiv$ treewidth):

Max. matching size across the cut.

Step 1: Branch decompositions

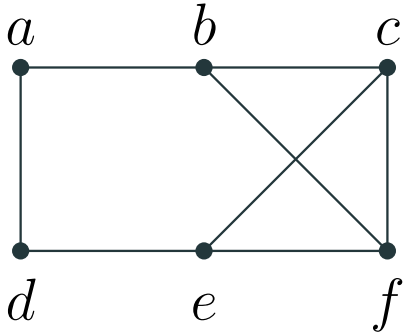


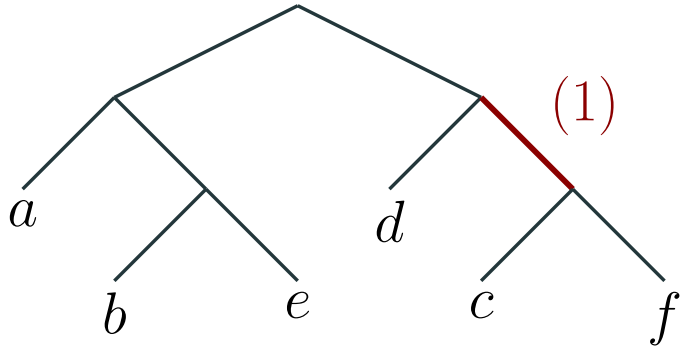


Rank-width ($\equiv$ clique-width):

GF(2)-rank of binary adjacency matrix of the cut.

Step 1: Branch decompositions





---

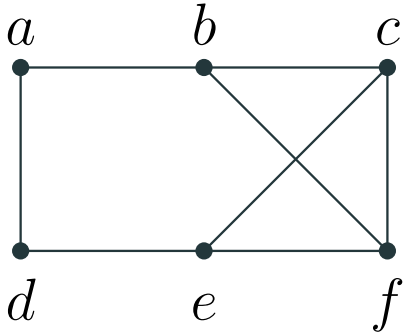
(1)

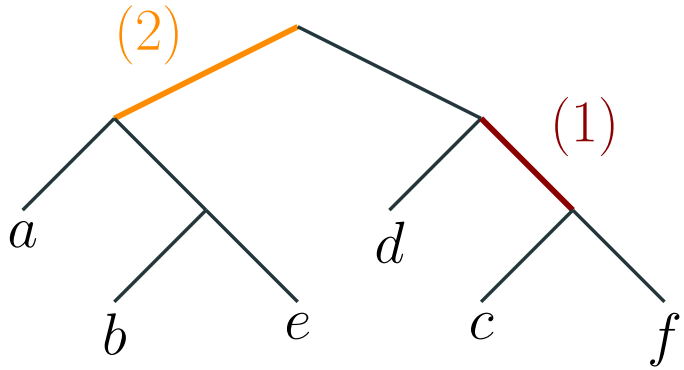
	a	b	d	e
c	0	1	0	1
f	0	1	0	1

Rank-width ($\equiv$ clique-width):

GF(2)-rank of binary adjacency matrix of the cut.

Step 1: Branch decompositions





---

(1)

	a	b	d	e
c	0	1	0	1
f	0	1	0	1

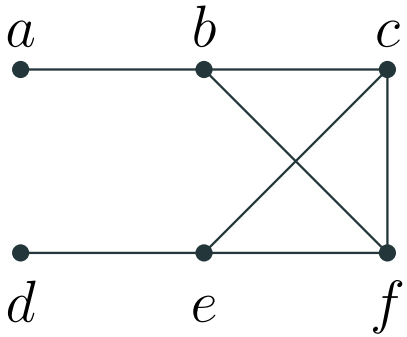
(2)

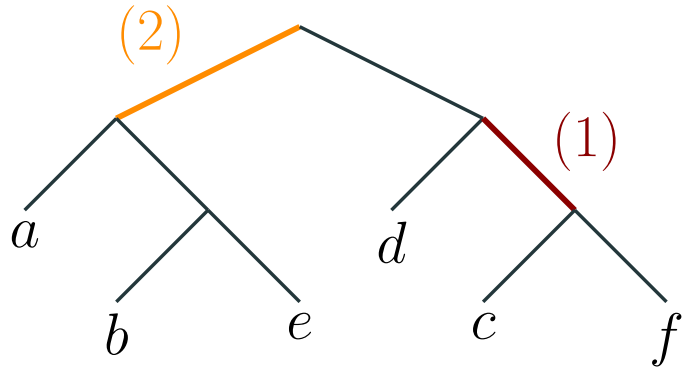
	c	d	f
a	0	1	0
b	1	0	1
e	1	1	1

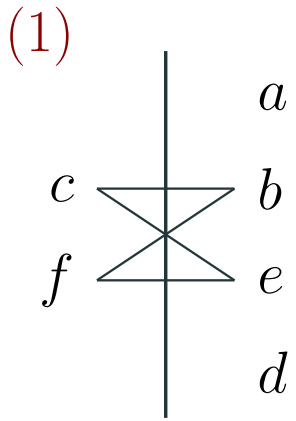
Rank-width ($\equiv$ clique-width):

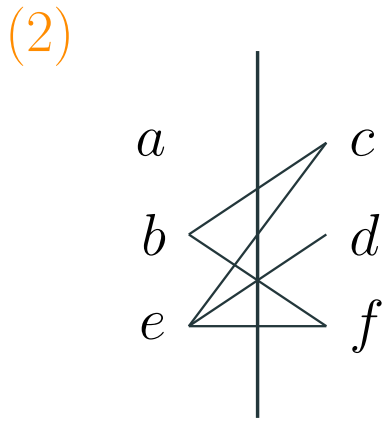
GF(2)-rank of binary adjacency matrix of the cut.

Step 1: Branch decompositions









MIM-width:

Max. size of an **induced** matching across the cut.

Step 2: Unifying the way of measuring complexity of cuts

All three measures admit asymptotically-equivalent characterizations via branchwidth... but the cut-functions are fundamentally different.

Do we really need to consider all possible kinds of cut-functions?

Step 2: Unifying the way of measuring complexity of cuts

$\mathscr{F}$-Branchwidth

For a family of bipartite graphs $\mathscr{F}$, $\mathscr{F}$-branchwidth measures the complexity of a cut (A, B) as

$$\max\{|F| : F \in \mathscr{F} \text{ is an induced subgraph in } G[A, B]\}.$$

Step 2: Unifying the way of measuring complexity of cuts

$\mathscr{F}$-Branchwidth

For a family of bipartite graphs $\mathscr{F}$, $\mathscr{F}$-branchwidth measures the complexity of a cut (A, B) as

$$\max\{|V(F) \cap A| : F \in \mathscr{F} \text{ is an induced subgraph in } G[A, B]\}.$$

Step 2: Unifying the way of measuring complexity of cuts

$\mathscr{F}$-Branchwidth

For a family of bipartite graphs $\mathscr{F}$, $\mathscr{F}$-branchwidth measures the complexity of a cut (A, B) as

$$\max\{|V(F) \cap A| : F \in \mathscr{F} \text{ is an induced subgraph in } G[A, B]\}.$$

- Symmetry: Allow only bipartite graphs with equally sized sides.

Step 2: Unifying the way of measuring complexity of cuts

$\mathscr{F}$-Branchwidth

For a family of bipartite graphs $\mathscr{F}$, $\mathscr{F}$-branchwidth measures the complexity of a cut (A, B) as

$$\max\{|V(F) \cap A| : F \in \mathscr{F} \text{ is an induced subgraph in } G[A, B]\}.$$

- Symmetry: Allow only bipartite graphs with equally sized sides.
- Unnatural to “skip” values.

Step 2: Unifying the way of measuring complexity of cuts

$\mathscr{F}$-Branchwidth

For a family of bipartite graphs $\mathscr{F}$, $\mathscr{F}$-branchwidth measures the complexity of a cut (A, B) as

$$\max\{|V(F) \cap A| : F \in \mathscr{F} \text{ is an induced subgraph in } G[A, B]\}.$$

- Symmetry: Allow only bipartite graphs with equally sized sides.
- Unnatural to "skip" values.
- The structure of a graph in $\mathscr{F}$ "witnessing" width k *should* say something about the structure of graphs witnessing width $k' < k$.

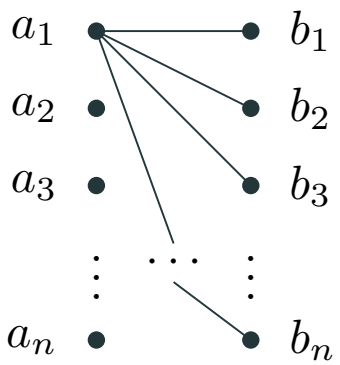
PH

Partner-hereditary (ph)

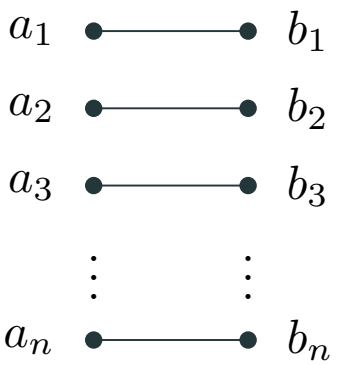
A family of bipartite graphs $\mathscr{F}$ is partner-hereditary (ph) if: For each $F \in \mathscr{F}$, fix a bipartition $(\{a_1, \ldots, a_n\}, \{b_1, \ldots, b_n\})$. Then, for all $I \subseteq [n]$, $F[\bigcup_{i \in I} \{a_i, b_i\}] \in \mathscr{F}$.

Partner-hereditary (ph)

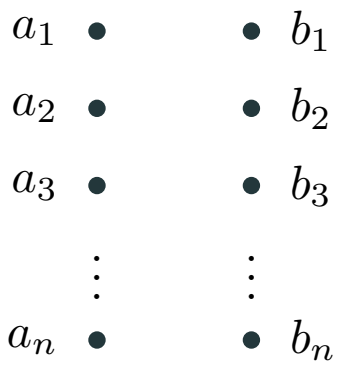
A family of bipartite graphs $\mathscr{F}$ is partner-hereditary (ph) if: For each $F \in \mathscr{F}$, fix a bipartition $(\{a_1, \ldots, a_n\}, \{b_1, \ldots, b_n\})$. Then, for all $I \subseteq [n]$, $F[\bigcup_{i \in I} \{a_i, b_i\}] \in \mathscr{F}$.



$\mathcal{F}_1$



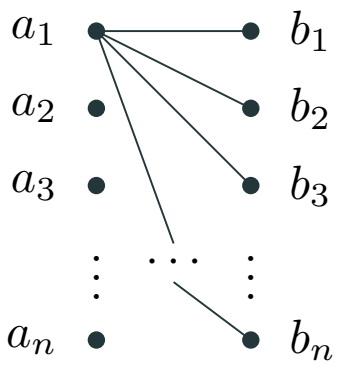
$\mathcal{F}_2$



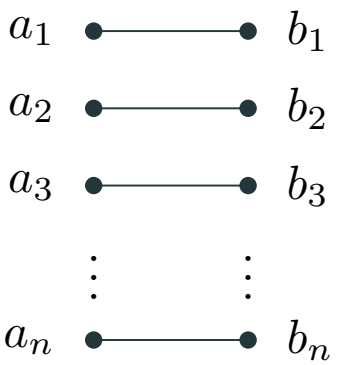
$\mathcal{F}_\emptyset$

Partner-hereditary (ph)

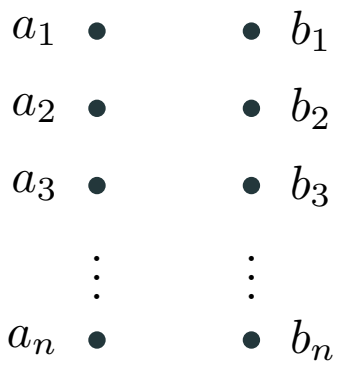
A family of bipartite graphs $\mathscr{F}$ is partner-hereditary (ph) if: For each $F \in \mathscr{F}$, fix a bipartition $(\{a_1, \ldots, a_n\}, \{b_1, \ldots, b_n\})$. Then, for all $I \subseteq [n]$, $F[\bigcup_{i \in I} \{a_i, b_i\}] \in \mathscr{F}$.



$\mathcal{F}_1$



$\mathcal{F}_2$

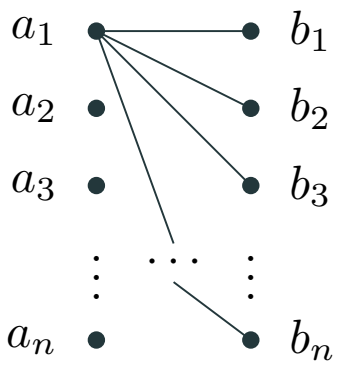


$\mathcal{F}_\emptyset$

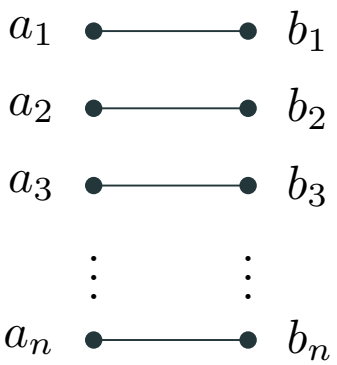
- $\mathscr{F}_1$ is not ph.

Partner-hereditary (ph)

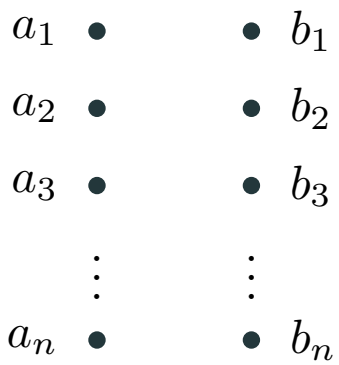
A family of bipartite graphs $\mathscr{F}$ is partner-hereditary (ph) if: For each $F \in \mathscr{F}$, fix a bipartition $(\{a_1, \ldots, a_n\}, \{b_1, \ldots, b_n\})$. Then, for all $I \subseteq [n]$, $F[\bigcup_{i \in I} \{a_i, b_i\}] \in \mathscr{F}$.



$\mathcal{F}_1$



$\mathcal{F}_2$

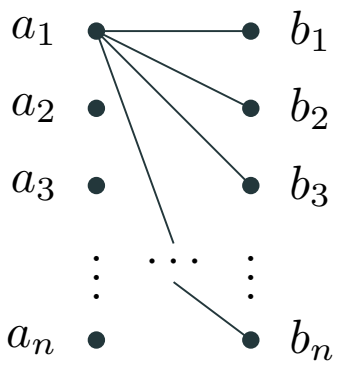


$\mathcal{F}_\emptyset$

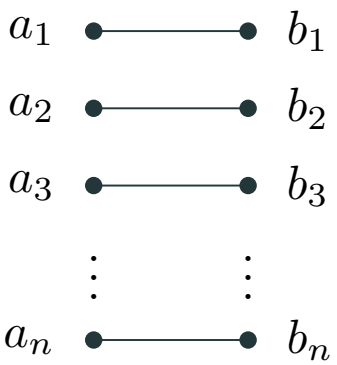
- $\mathscr{F}_1$ is not ph.
- $\mathscr{F}_2$ is ph.

Partner-hereditary (ph)

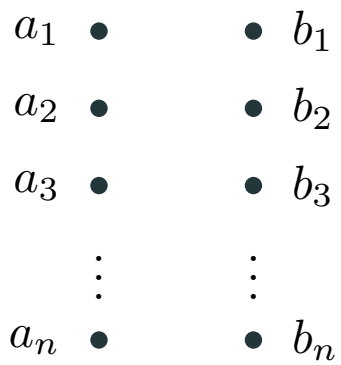
A family of bipartite graphs $\mathscr{F}$ is partner-hereditary (ph) if: For each $F \in \mathscr{F}$, fix a bipartition $(\{a_1, \ldots, a_n\}, \{b_1, \ldots, b_n\})$. Then, for all $I \subseteq [n]$, $F[\bigcup_{i \in I} \{a_i, b_i\}] \in \mathscr{F}$.



$\mathcal{F}_1$



$\mathcal{F}_2$



$\mathcal{F}_\emptyset$

- $\mathscr{F}_1$ is not ph.
- $\mathscr{F}_2$ is ph.
- $\mathscr{F}_1 \cup \mathscr{F}_\emptyset$ is ph.

What does this capture?

Width	$\mathscr{F}$
Treewidth	Matchings, antimatchings, balanced chains, complete bipartite

What does this capture?

Width	$\mathscr{F}$
Treewidth	Matchings, antimatchings, balanced chains, complete bipartite
Clique-width	Matchings, antimatchings, balanced chains

What does this capture?

Width	$\mathscr{F}$
Treewidth	Matchings, antimatchings, balanced chains, complete bipartite
Clique-width	Matchings, antimatchings, balanced chains
Mim-width	Matchings

Back to $\mathscr{F}$-branchwidth

Size-identifiable (si)

A ph graph family $\mathscr{F}$ is size-identifiable (si) if for all n, there is precisely one graph of order $2n$ in $\mathscr{F}$.

Back to $\mathscr{F}$-branchwidth

Size-identifiable (si)

A ph graph family $\mathscr{F}$ is size-identifiable (si) if for all n, there is precisely one graph of order $2n$ in $\mathscr{F}$.

Theorem

Let $\mathscr{F}$ be a ph graph family. Let $\mathscr{F}^{\downarrow}$ be union of all si graph families contained in $\mathscr{F}$ up to constantly many exceptions. Then, $\mathscr{F}$-bw and $\mathscr{F}^{\downarrow}$-bw are asymptotically equivalent.

Back to $\mathscr{F}$-branchwidth

Size-identifiable (si)

A ph graph family $\mathscr{F}$ is size-identifiable (si) if for all n, there is precisely one graph of order $2n$ in $\mathscr{F}$.

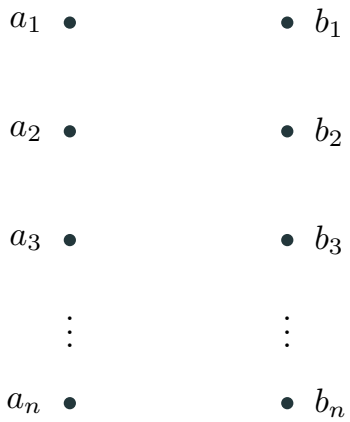
Theorem

Let $\mathscr{F}$ be a ph graph family. Let $\mathscr{F}^{\downarrow}$ be union of all si graph families contained in $\mathscr{F}$ up to constantly many exceptions. Then, $\mathscr{F}$-bw and $\mathscr{F}^{\downarrow}$-bw are asymptotically equivalent.

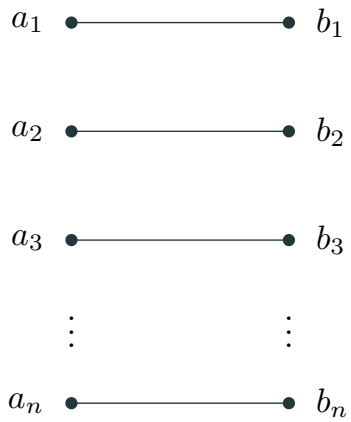
Theorem

There are only six si ph graph families.

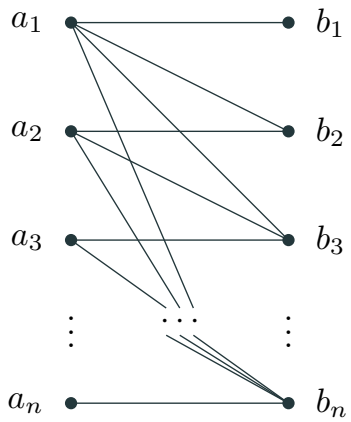
The si ph graph families



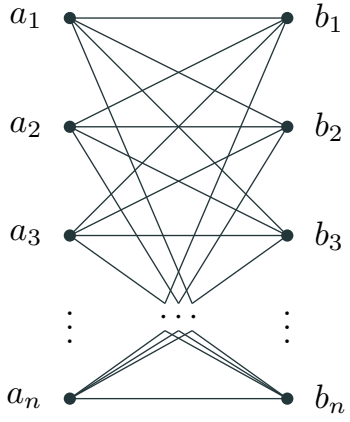
$\mathcal{F}_{\emptyset}$



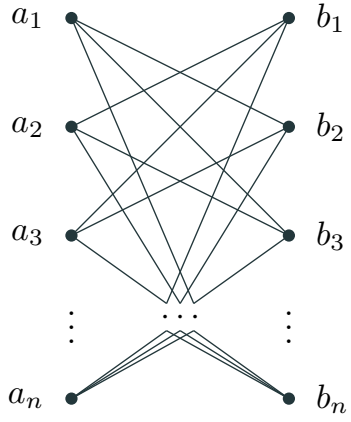
$\mathcal{F}_{\mathsf{match}}$



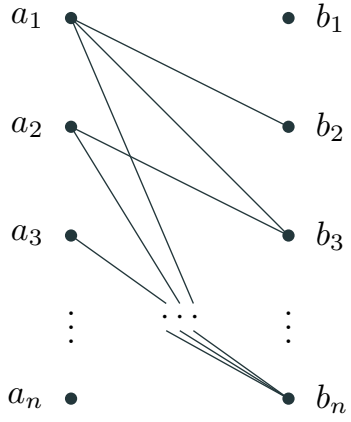
$\mathcal{F}_{\mathsf{chain}}$



$\mathcal{F}_{\mathsf{complete}}$



$\mathcal{F}_{\mathsf{antimatch}}$

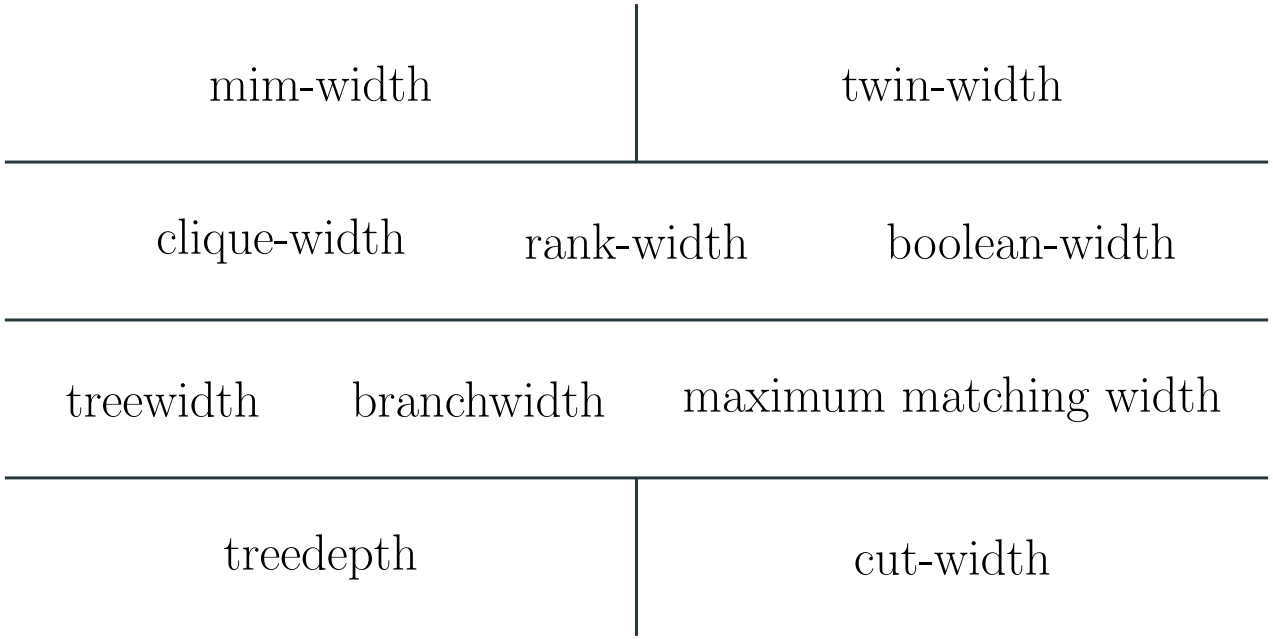


$\mathcal{F}_{\mathsf{schain}}$

What does this capture?

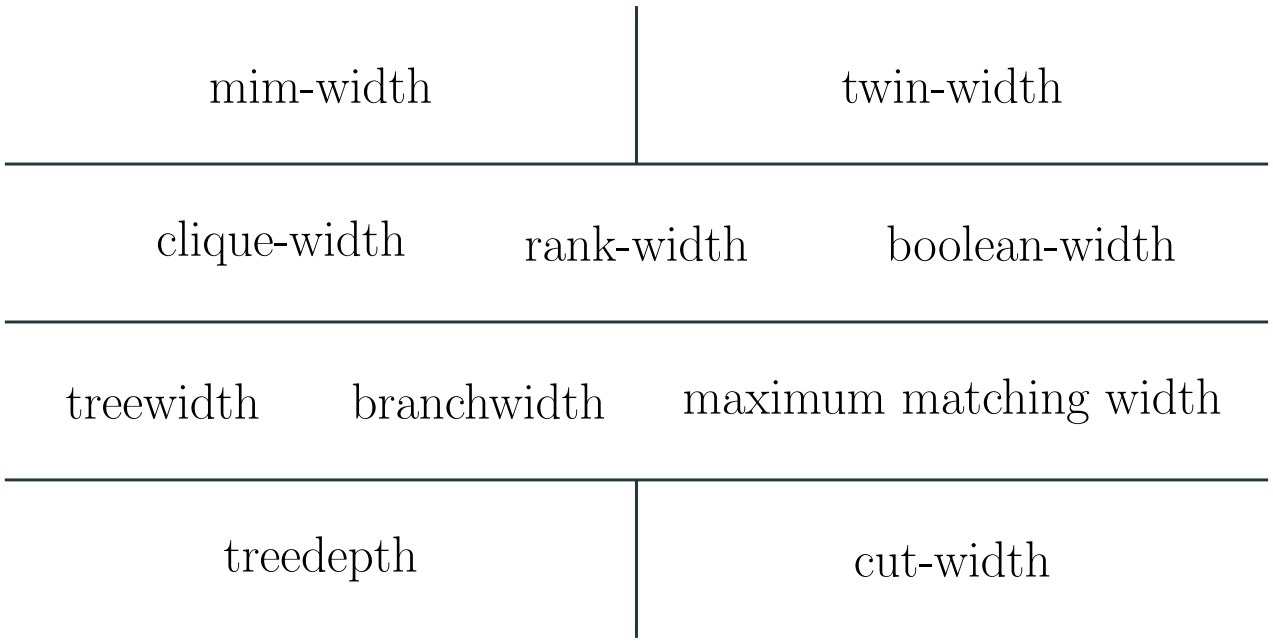
Width	$\mathscr{F}$
Treewidth	Matchings, antimatchings, balanced chains, complete bipartite
Clique-width	Matchings, antimatchings, balanced chains
Mim-width	Matchings

Recap

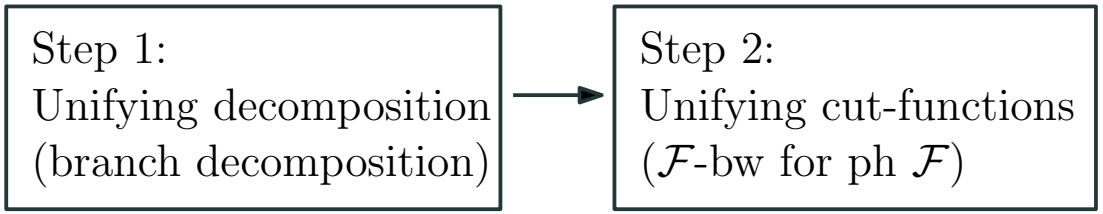

mim-width
twin-width
clique-width
rank-width
boolean-width
treewidth
branchwidth
maximum matching width
treedepth
cut-width

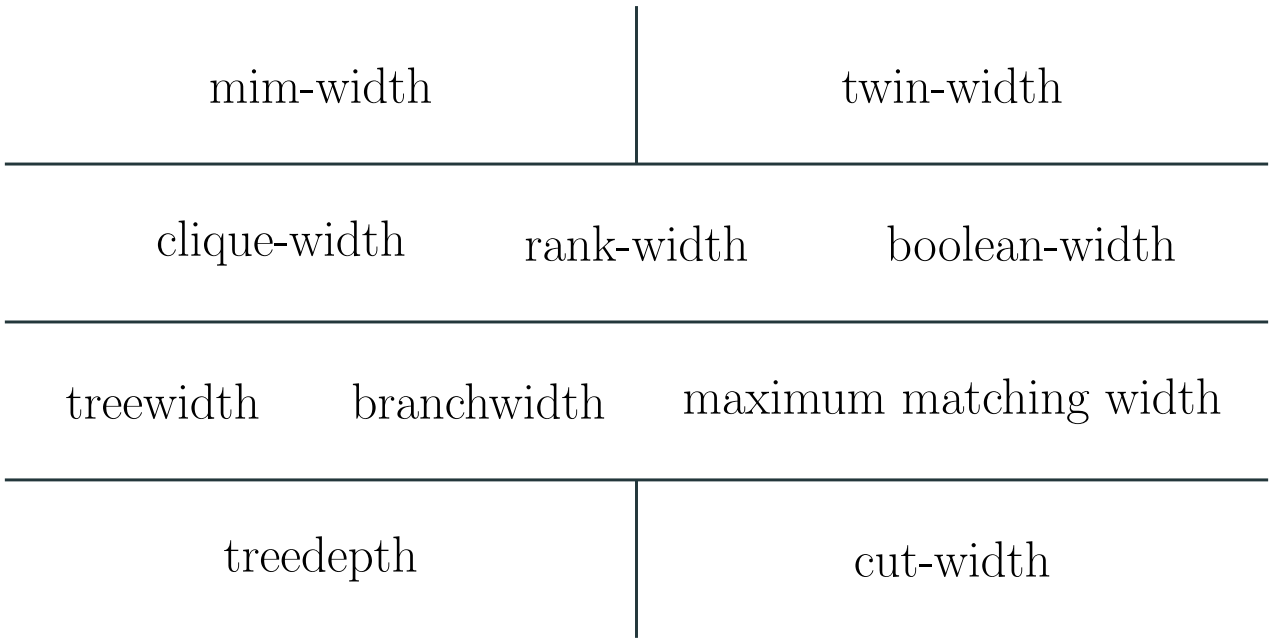
Recap

Step 1:
Unifying decomposition
(branch decomposition)

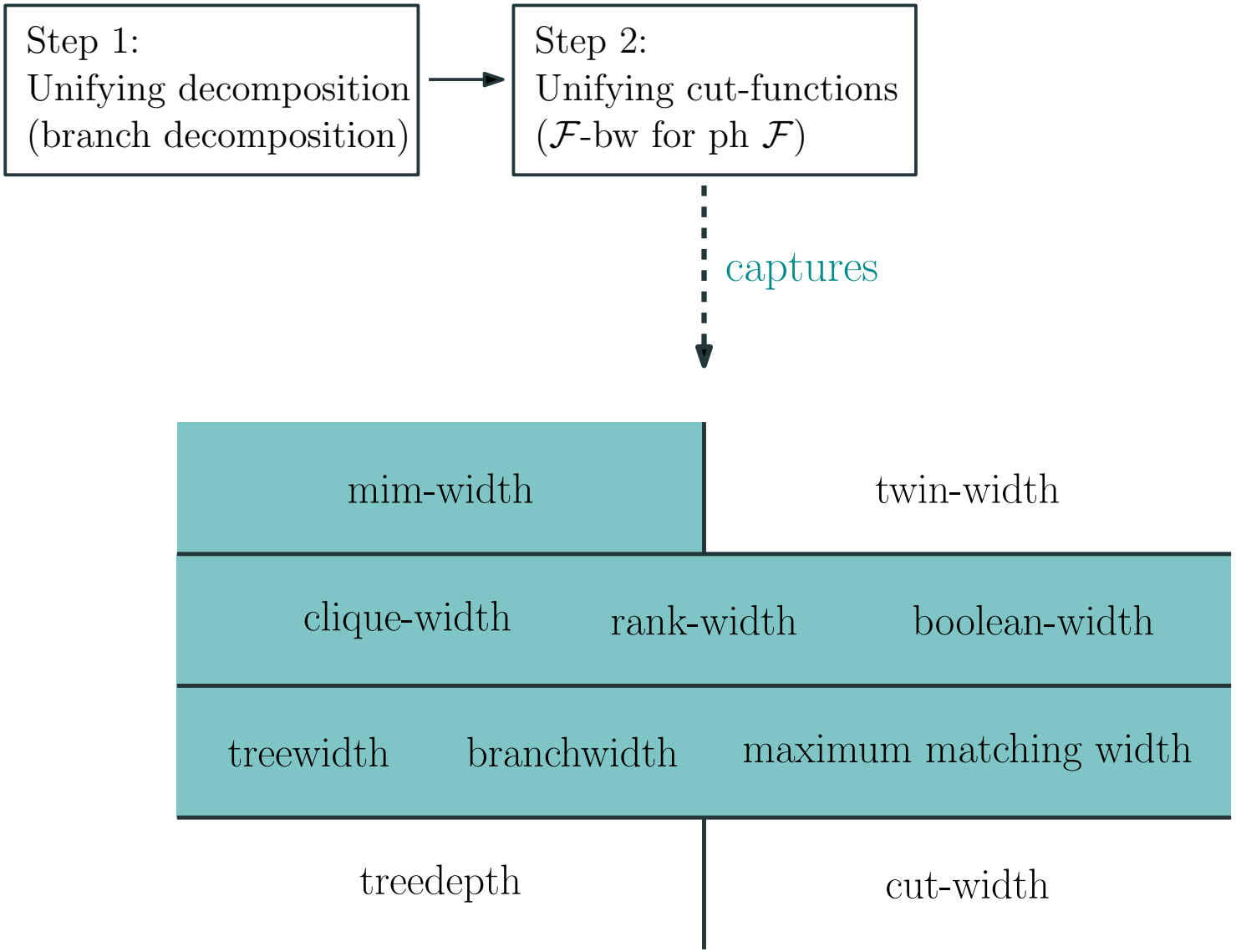


Recap


Step 1:
Unifying decomposition
(branch decomposition)
Step 2:
Unifying cut-functions
($\mathcal{F}$-bw for ph $\mathcal{F}$)


mim-width
twin-width
clique-width
rank-width
boolean-width
treewidth
branchwidth
maximum matching width
treedepth
cut-width

Recap


Step 1:
Unifying decomposition
(branch decomposition)
Step 2:
Unifying cut-functions
($\mathcal{F}$-bw for ph $\mathcal{F}$)
captures
mim-width
twin-width
clique-width
rank-width
boolean-width
treewidth
branchwidth
maximum matching width
treedepth
cut-width

Recap

Step 1:
Unifying decomposition
(branch decomposition)

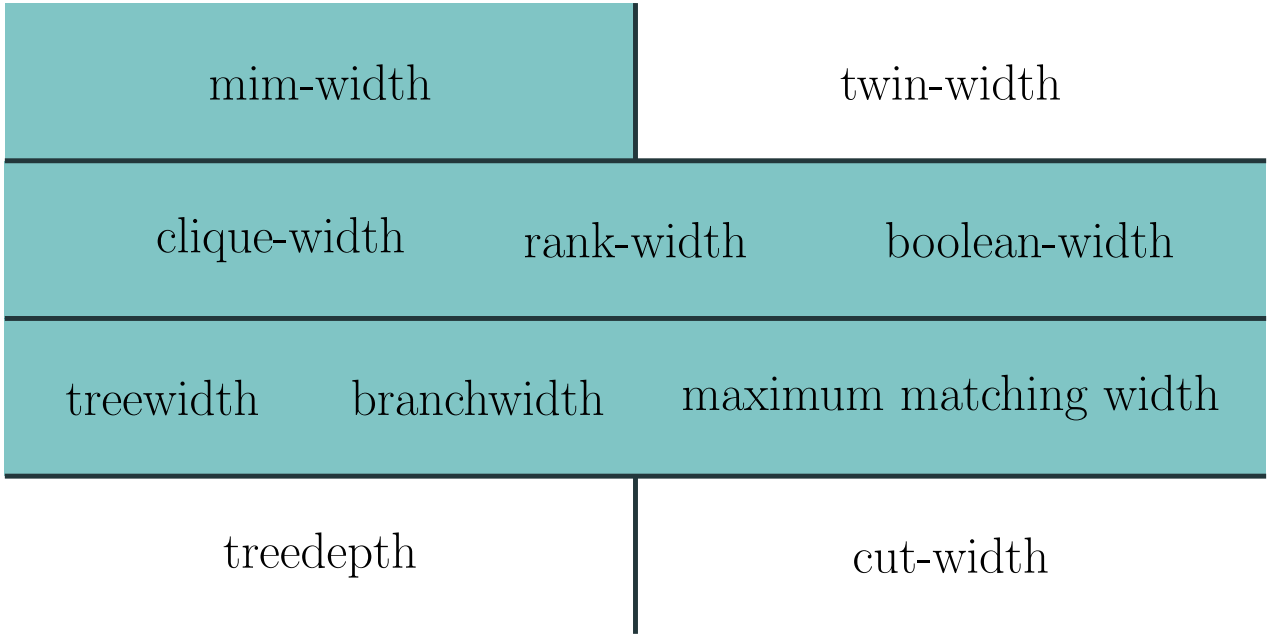
→

Step 2:
Unifying cut-functions
($\mathcal{F}$-bw for ph $\mathcal{F}$)

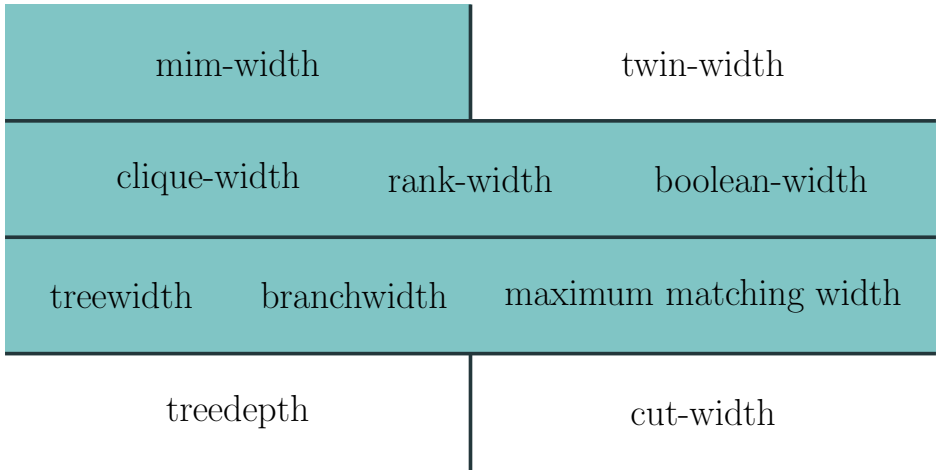
→

Only **six** ph classes "truly matter" for $\mathcal{F}$-bw.

captures

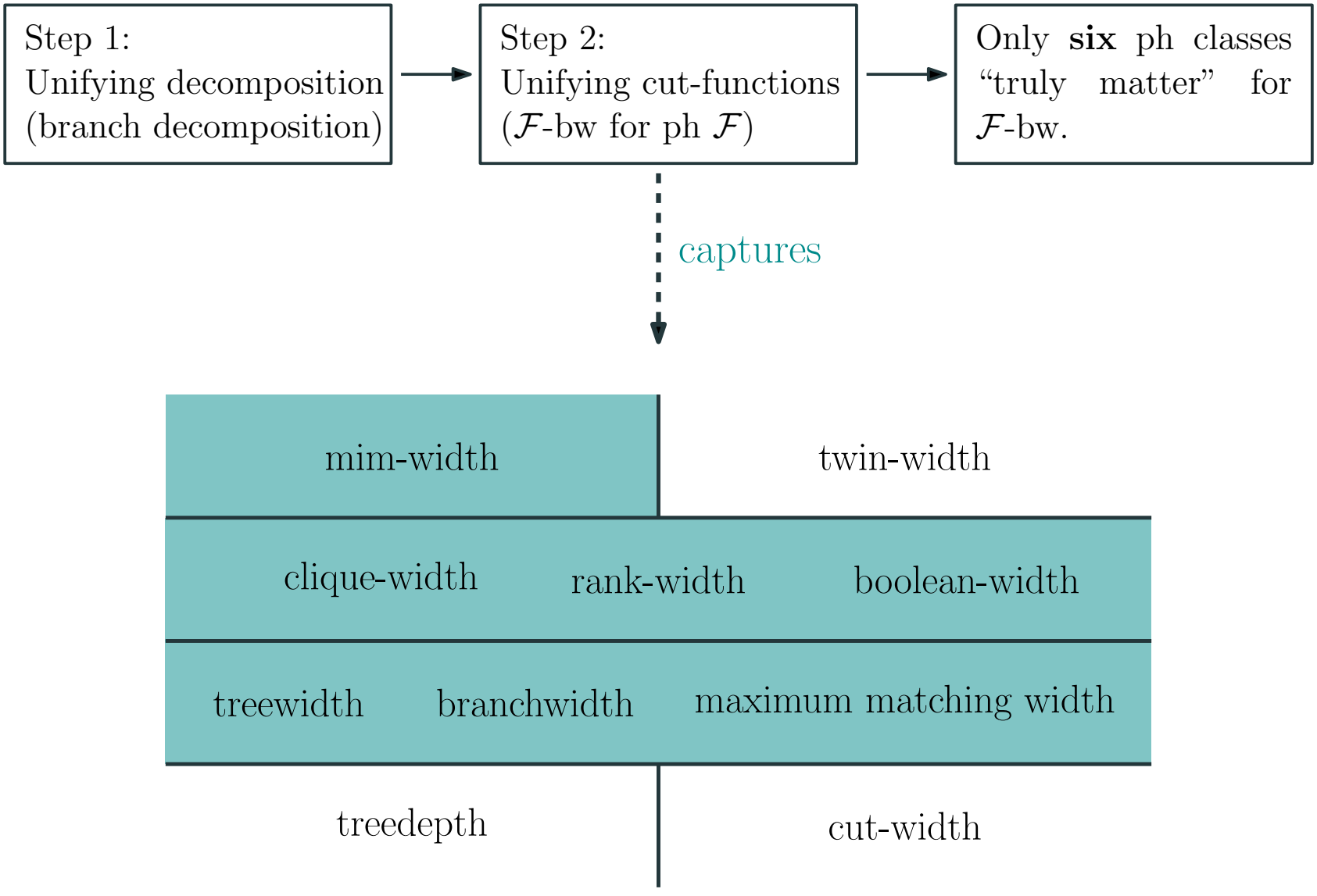


Approximating $\mathscr{F}$-branchwidth

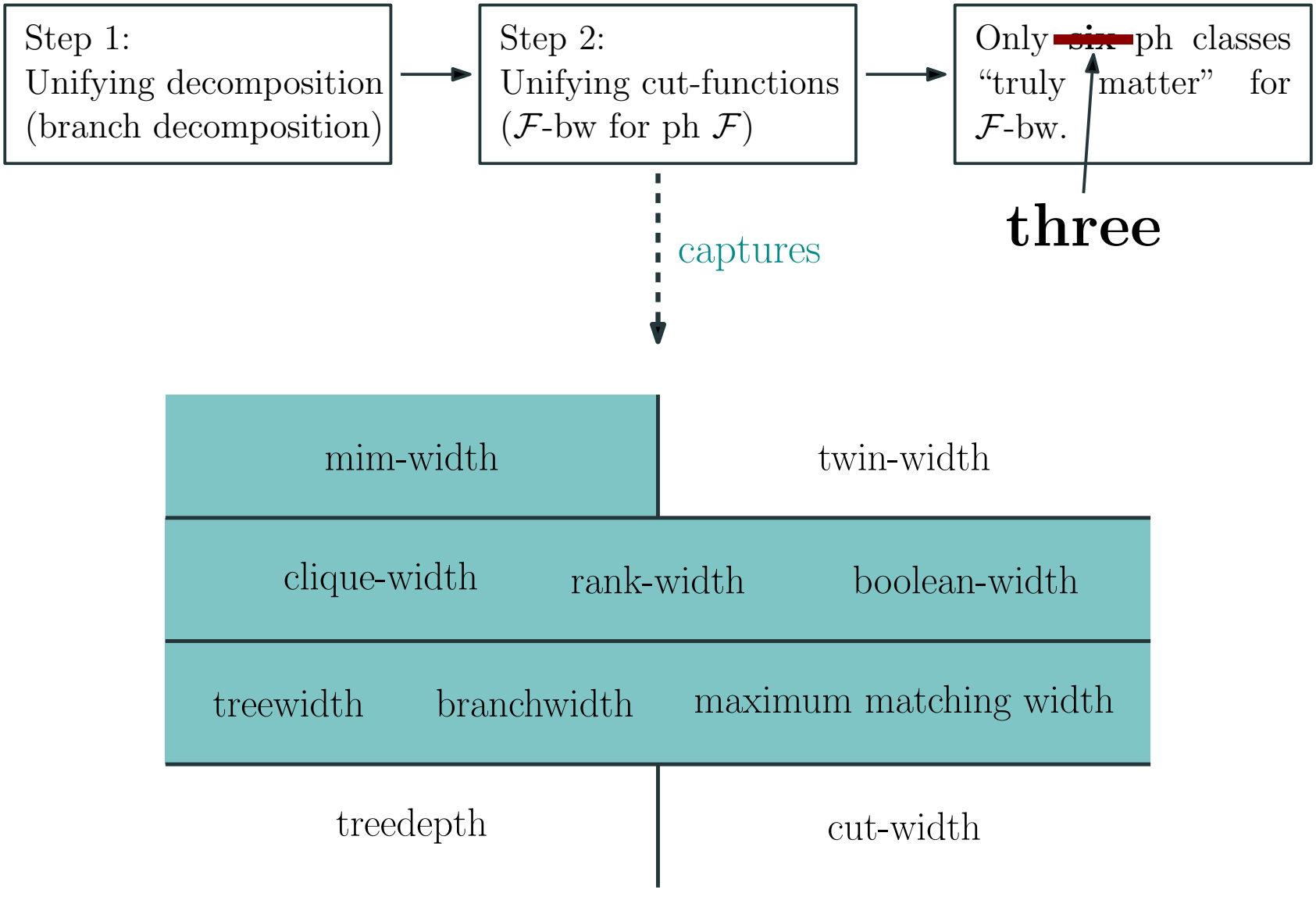


- Until now: $\mathscr{F}$-branchwidth can be used to characterize width measures.
- From now: Use $\mathscr{F}$-branchwidth to compute approximately-optimal decompositions for width measures.

Recap


Step 1:
Unifying decomposition
(branch decomposition)
Step 2:
Unifying cut-functions
($\mathcal{F}$-bw for ph $\mathcal{F}$)
Only **six** ph classes "truly matter" for $\mathcal{F}$-bw.
captures
mim-width
twin-width
clique-width
rank-width
boolean-width
treewidth
branchwidth
maximum matching width
treedepth
cut-width

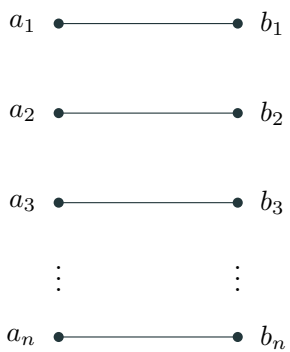
Recap


Step 1:
Unifying decomposition
(branch decomposition)
Step 2:
Unifying cut-functions
($\mathcal{F}$-bw for ph $\mathcal{F}$)
Only ~~six~~ ph classes
"truly matter" for
$\mathcal{F}$-bw.
three
captures
mim-width
twin-width
clique-width
rank-width
boolean-width
treewidth
branchwidth
maximum matching width
treedepth
cut-width

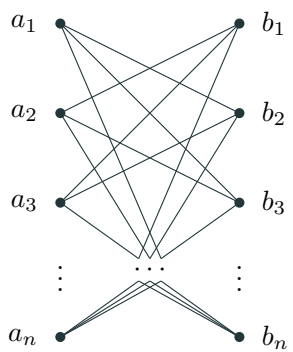
Primal families

Theorem

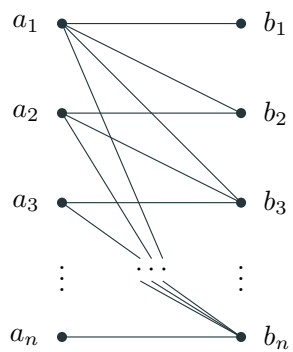
Let $\mathscr{F}$ be any union of si ph graph families. Let $\mathscr{F}^*$ be the union of the classes of matchings, antimatchings, and chains contained in $\mathscr{F}$. Then an optimal $\mathscr{F}^*$-branch decomposition of any graph G has $\mathscr{F}$-branchwidth at most $3 \cdot \mathscr{F}\text{-bw}(G) + 1$.



$\mathcal{F}_{\text{match}}$



$\mathcal{F}_{\text{antimatch}}$

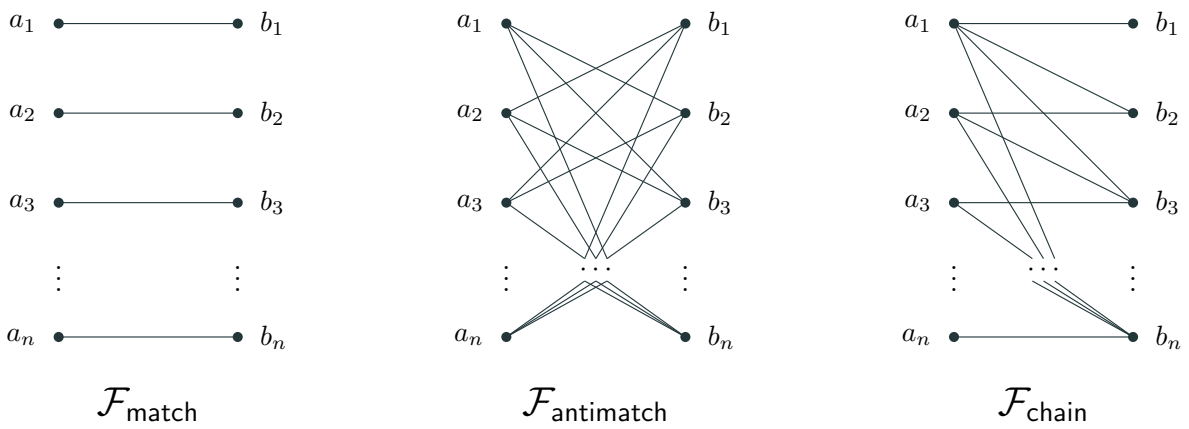


$\mathcal{F}_{\text{chain}}$

Primal families

Theorem

Let $\mathscr{F}$ be any union of si ph graph families. Let $\mathscr{F}^*$ be the union of the classes of matchings, antimatchings, and chains contained in $\mathscr{F}$. Then an optimal $\mathscr{F}^*$-branch decomposition of any graph G has $\mathscr{F}$-branchwidth at most $3 \cdot \mathscr{F}\text{-bw}(G) + 1$.



It suffices to give an algorithm computing $\mathscr{F}^*$-branchwidth for unions $\mathscr{F}^*$ of these three primal graph families.

Approximating $\mathscr{F}^*$-branchwidth

Theorem

Let $\mathscr{F}^*$ be a union of primal graph families. The problem of computing the $\mathscr{F}^*$-branchwidth of a graph G

1. *is fixed-parameter tractable parameterized by the treewidth plus the maximum degree of G,*
2. *is fixed-parameter tractable parameterized by the treedepth of G, and*
3. *has a linear kernel parameterized by the feedback edge set number of G.*

Approximating $\mathscr{F}^*$-branchwidth

Theorem

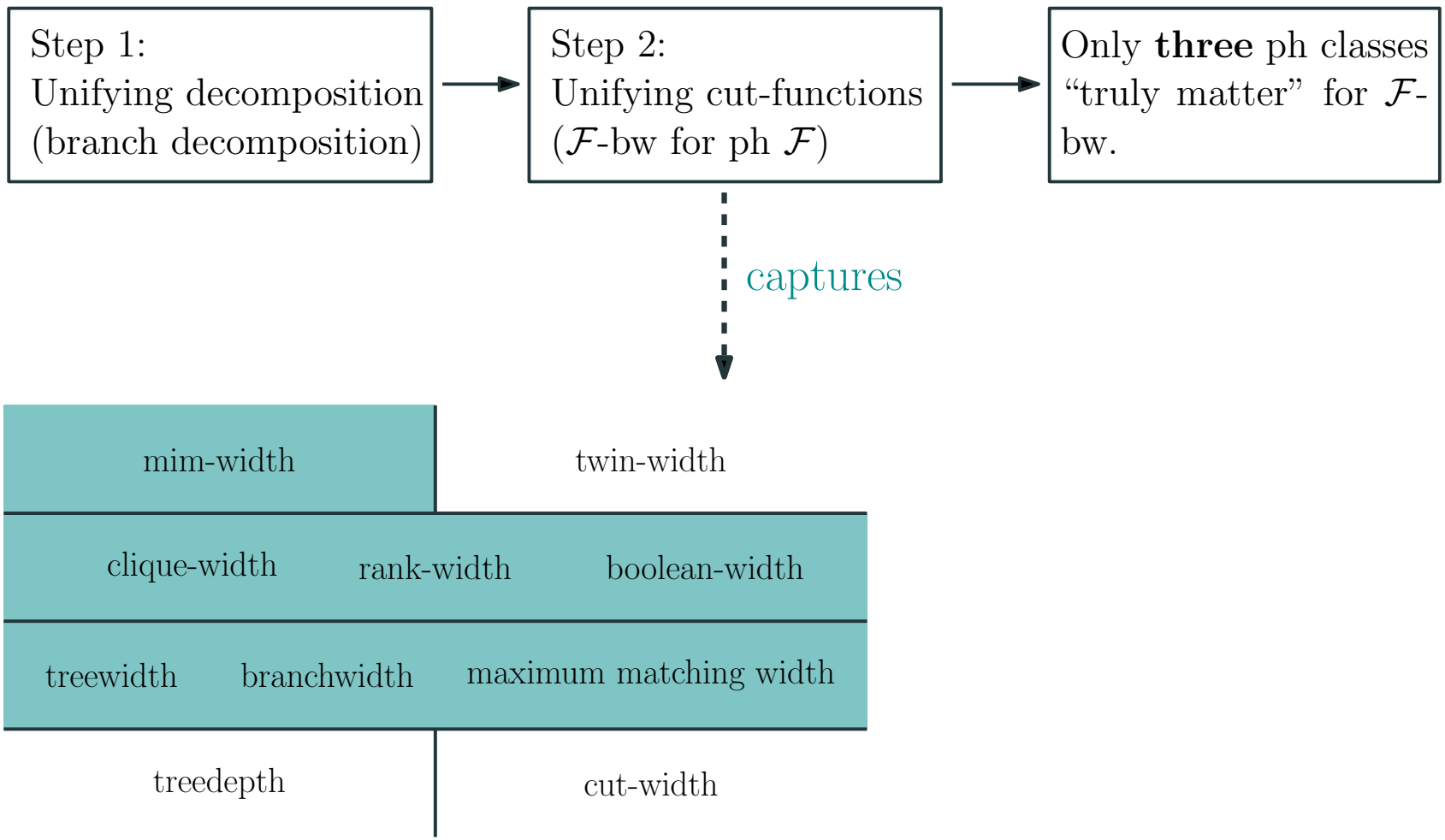
Let $\mathscr{F}^*$ be a union of primal graph families. The problem of computing the $\mathscr{F}^*$-branchwidth of a graph G

1. *is fixed-parameter tractable parameterized by the treewidth plus the maximum degree of G,*
2. *is fixed-parameter tractable parameterized by the treedepth of G, and*
3. *has a linear kernel parameterized by the feedback edge set number of G.*

Consequence

We can compute the mim-width under any of the above structural parameterizations exactly.

Conclusion


Step 1:
Unifying decomposition
(branch decomposition)
Step 2:
Unifying cut-functions
($\mathcal{F}$-bw for ph $\mathcal{F}$)
Only **three** ph classes "truly matter" for $\mathcal{F}$-bw.
captures
mim-width
twin-width
clique-width
rank-width
boolean-width
treewidth
branchwidth
maximum matching width
treedepth
cut-width

Conclusion

Step 1:
Unifying decomposition
(branch decomposition)

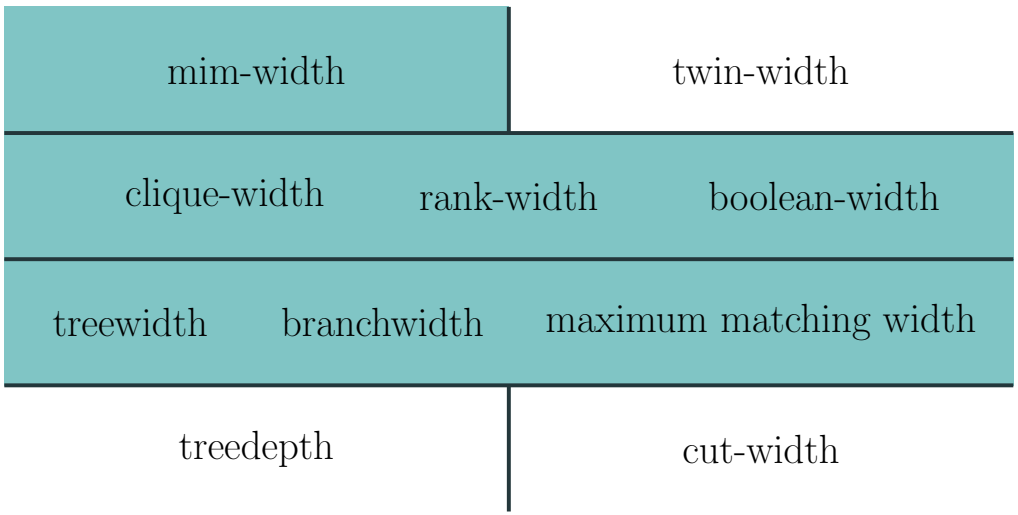
→

Step 2:
Unifying cut-functions
($\mathcal{F}$-bw for ph $\mathcal{F}$)

→

Only **three** ph classes "truly matter" for $\mathcal{F}$-bw.

captures



On graphs of

- low treewidth + max. degree,
- treedepth, or
- FES number,

we can approximate any $\mathcal{F}$-bw efficiently.

Thank You!