Tight Lower Bounds for Problems Parameterized by **Rank-width**

Séminaire ALGCO, LIRMM, December 1

ETH

Benjamin Bergougnoux, University of Warsaw

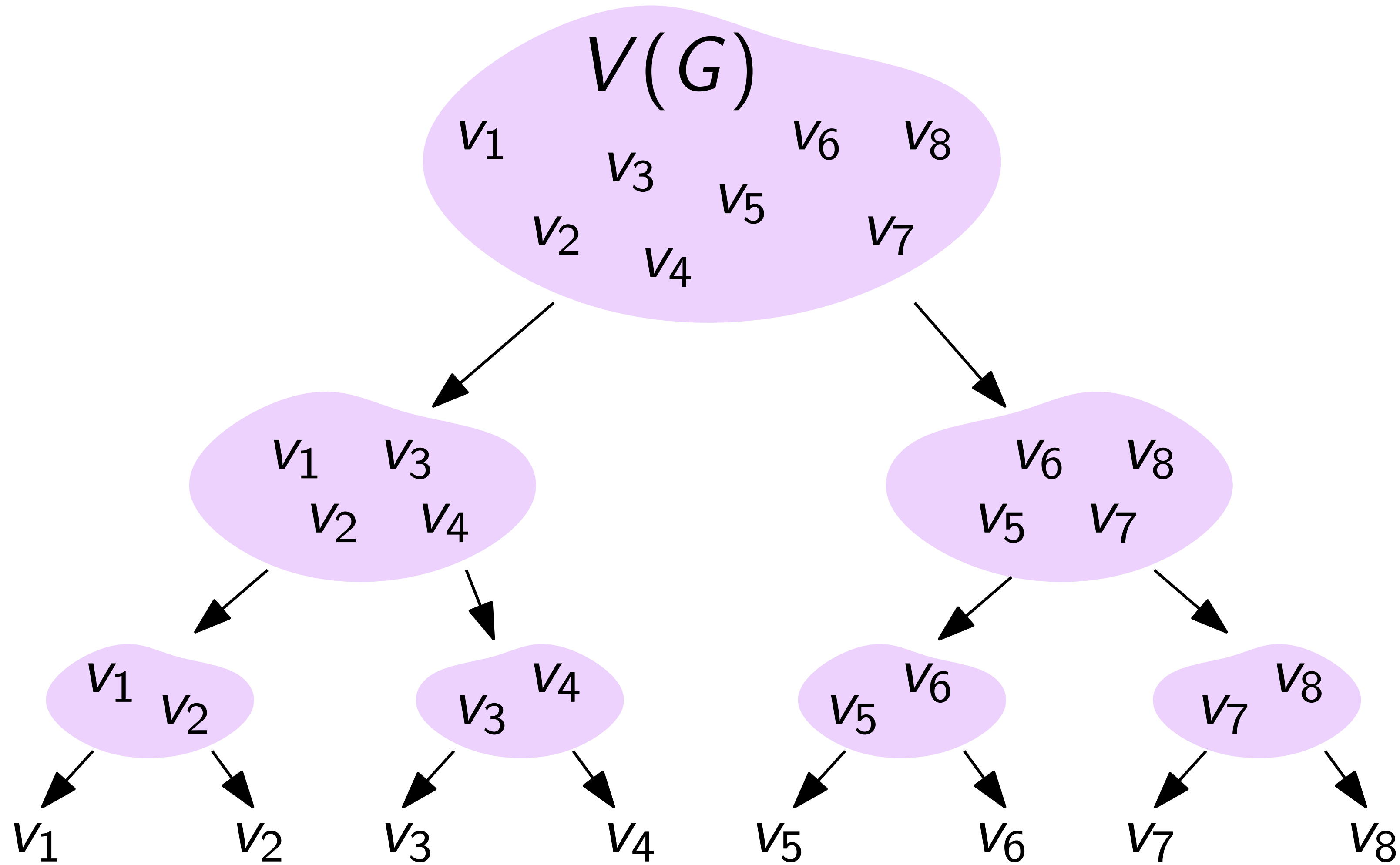
Feat **Tuukka Korhonen** and **Jesper Nederlof**

University of Bergen

University of Eindhoven

Width parameters

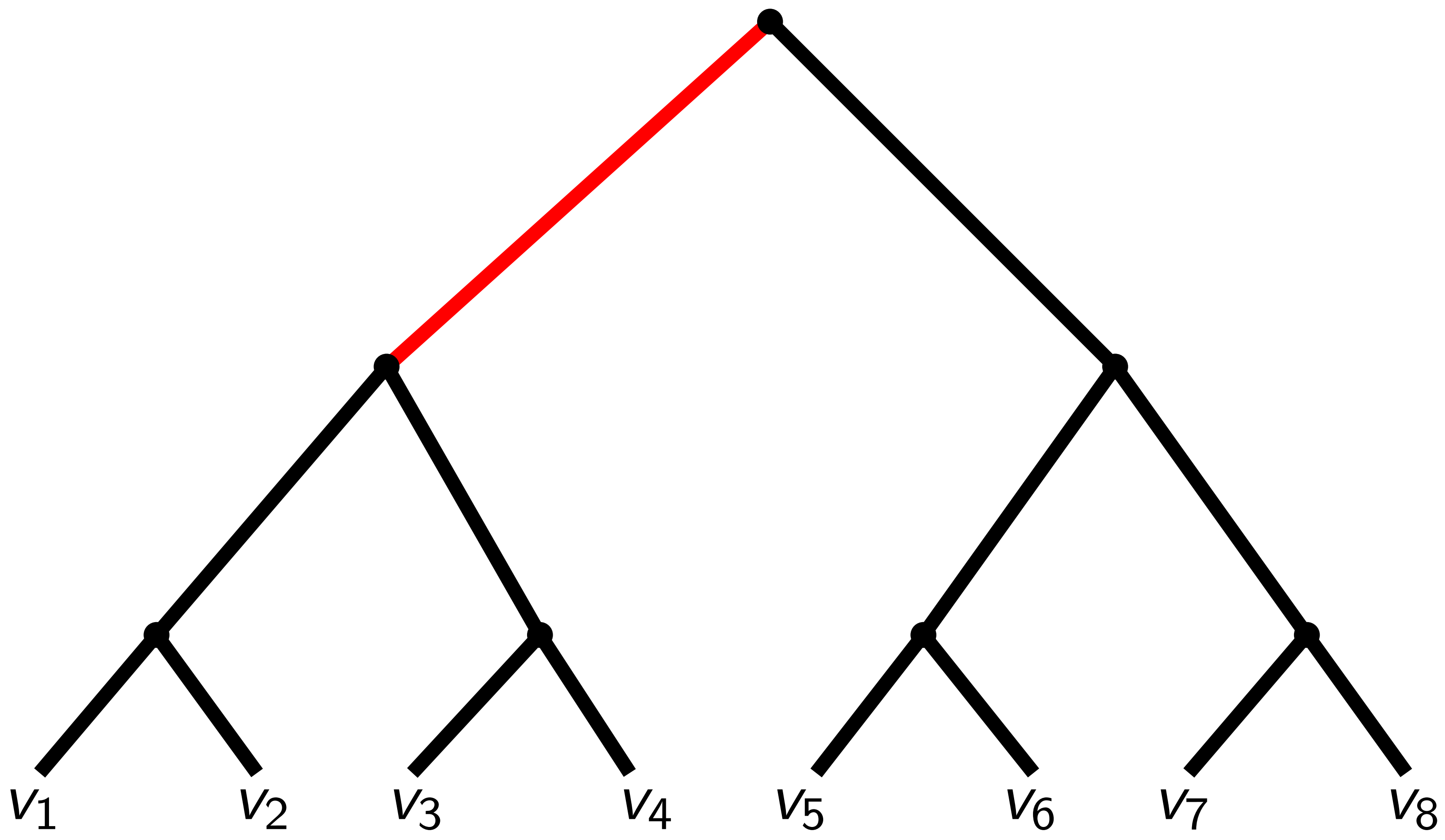
Recursively decompose a graph into **simple** cuts



Branch-decomposition: recursively **cut the vertex set** in **two**

Width parameters

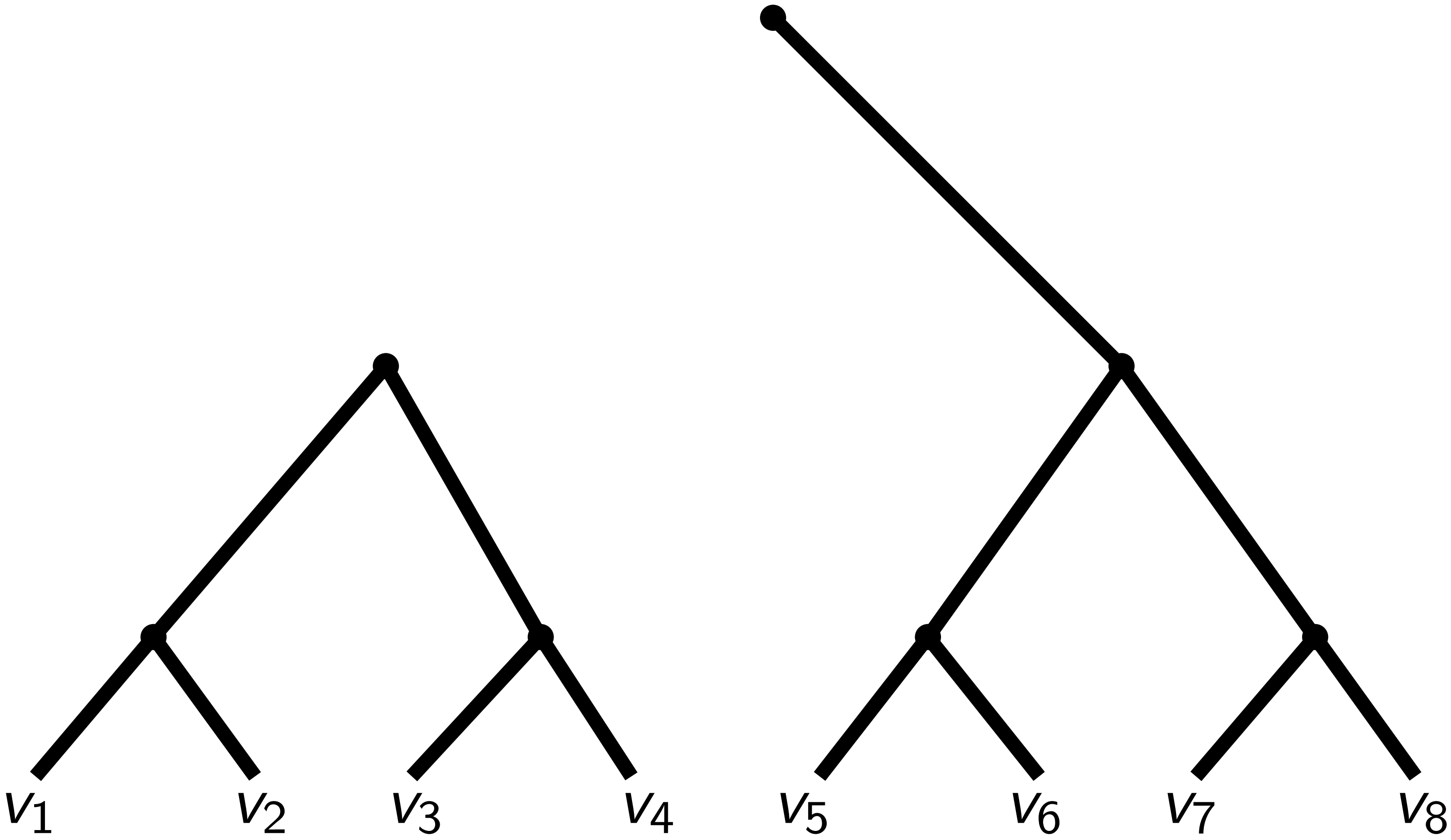
Recursively decompose a graph into **simple** cuts



Branch-decomposition: recursively **cut the vertex set** in **two**

Width parameters

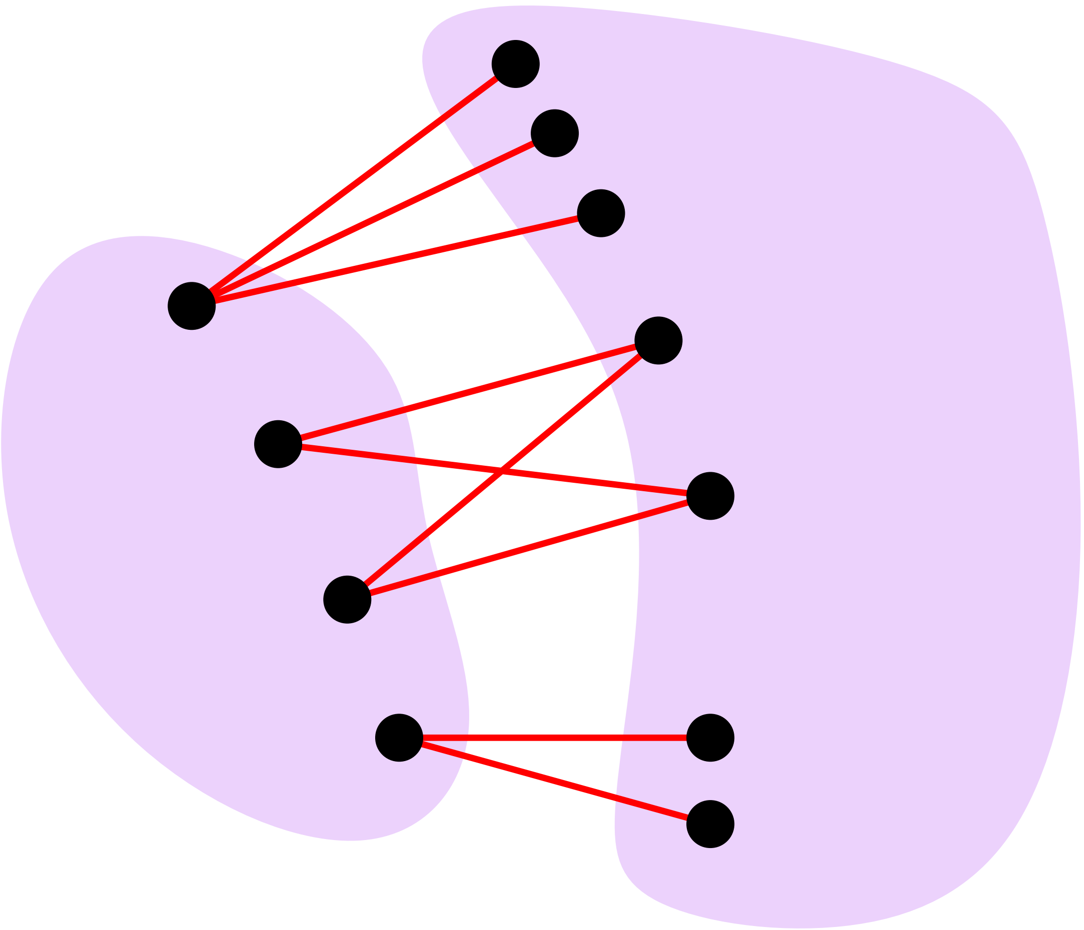
Recursively decompose a graph into **simple** cuts



Branch-decomposition: recursively **cut the vertex set** in **two**

Width parameters

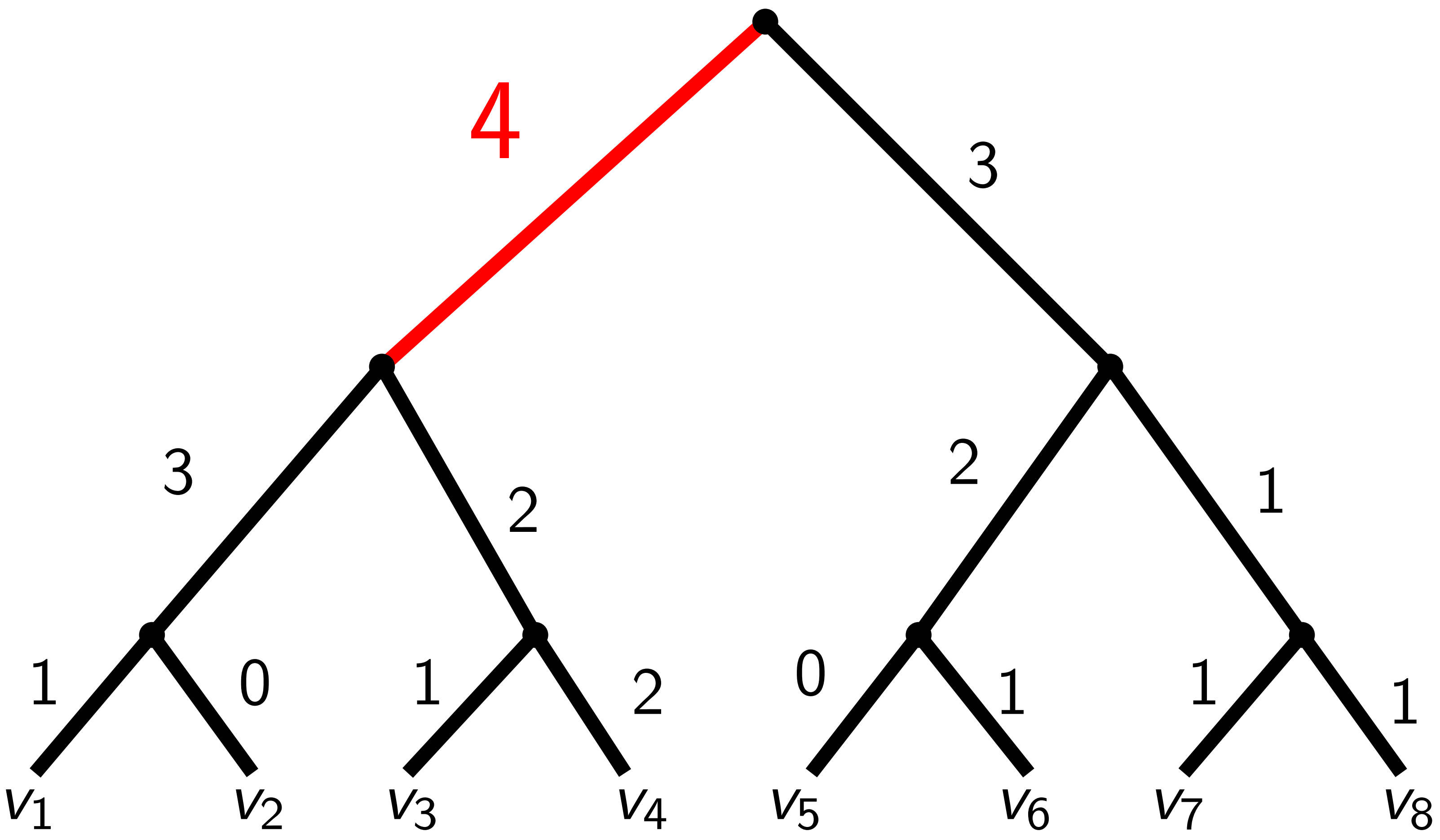
Simplicity of cuts is measured with a **cut function** f: **cut** $\to \mathbb{N}$.



Different notions of **simplicity** = different **width parameters**.

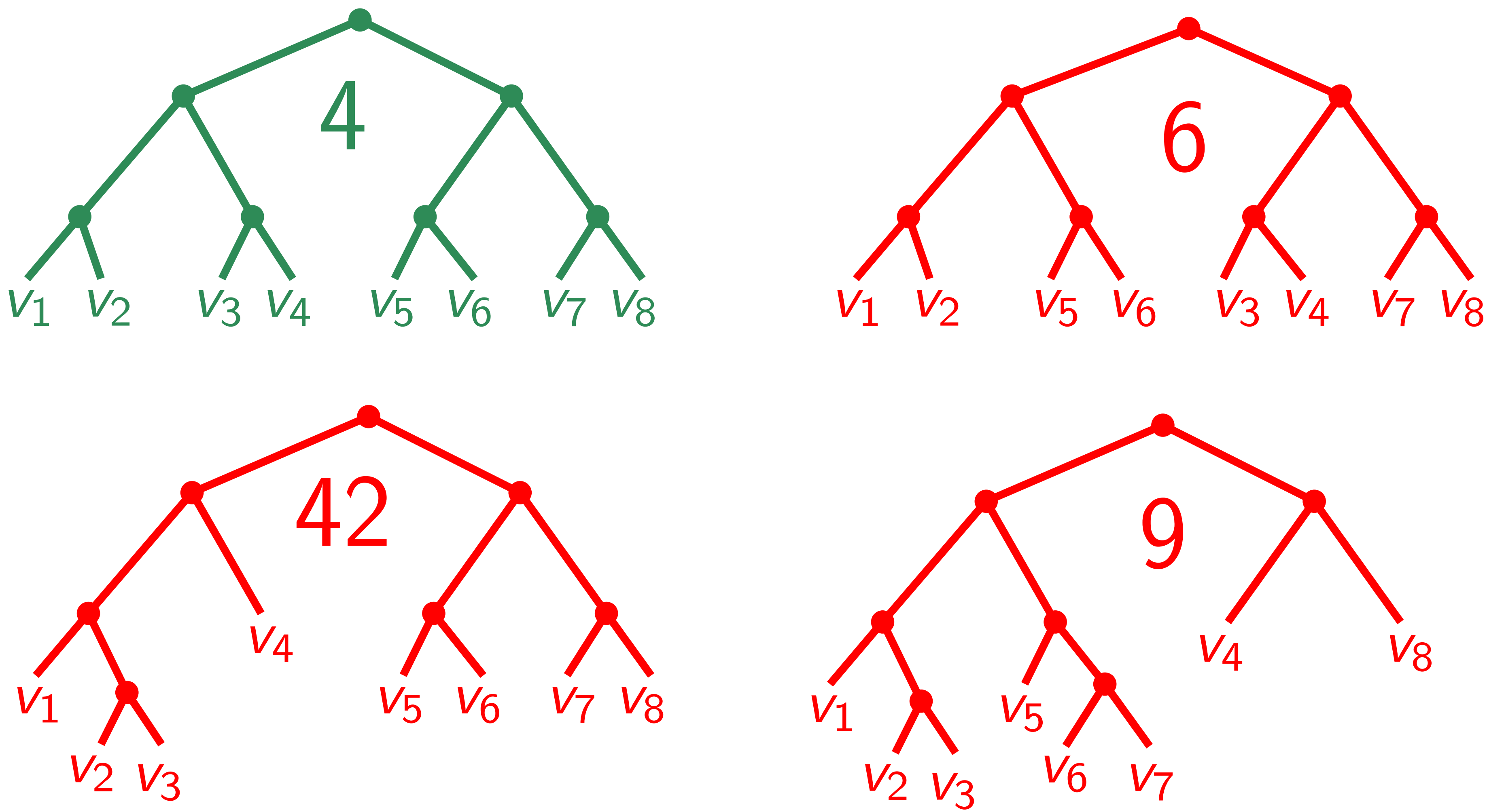
Width parameters

Width of a **decomposition** D = **max** f(cut) over the **cuts** of D.



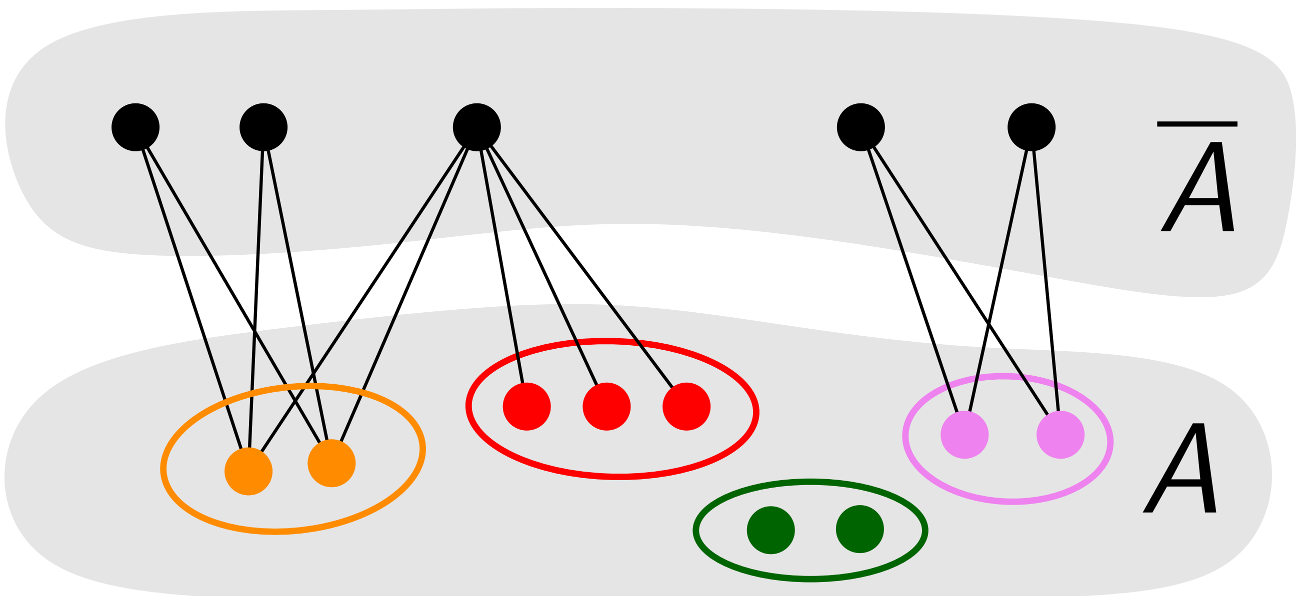
Width parameters

Width of a **graph** = **min** **widths** of its **decompositions**.



Width of a graph **class** = **max** **widths** of its **graphs**.

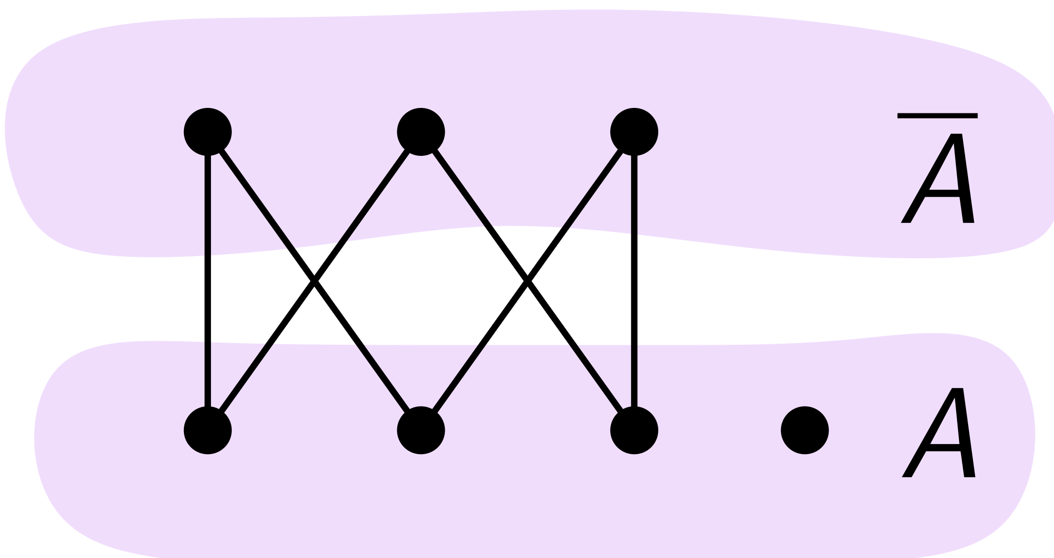
Module-width [Rao, 2006]
Defined from $\mathrm{mw}(A) := |\{N(v) \cap \overline{A} \mid v \in A\}|$.



Linearly equivalent to clique-width!

Theorem [Rao, 2006]
For all graphs G, we have $\mathrm{mw}(G) \leqslant \mathrm{cw}(G) \leqslant 2\mathrm{mw}(G)$.

Rank-width [Oum, 2006]
Defined from $\mathrm{rw}(A) :=$ the **rank of the adjacency matrix** between A and $\overline{A}$ over the binary field.

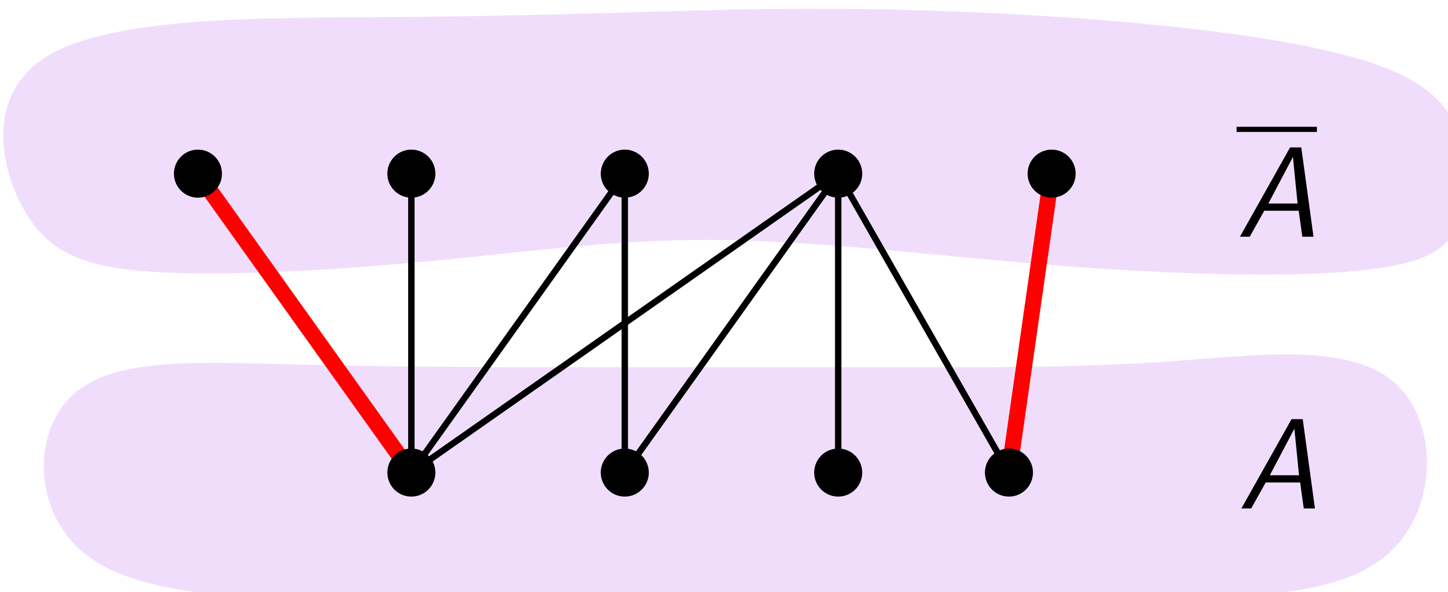


$$\begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

Equivalent to clique-width!

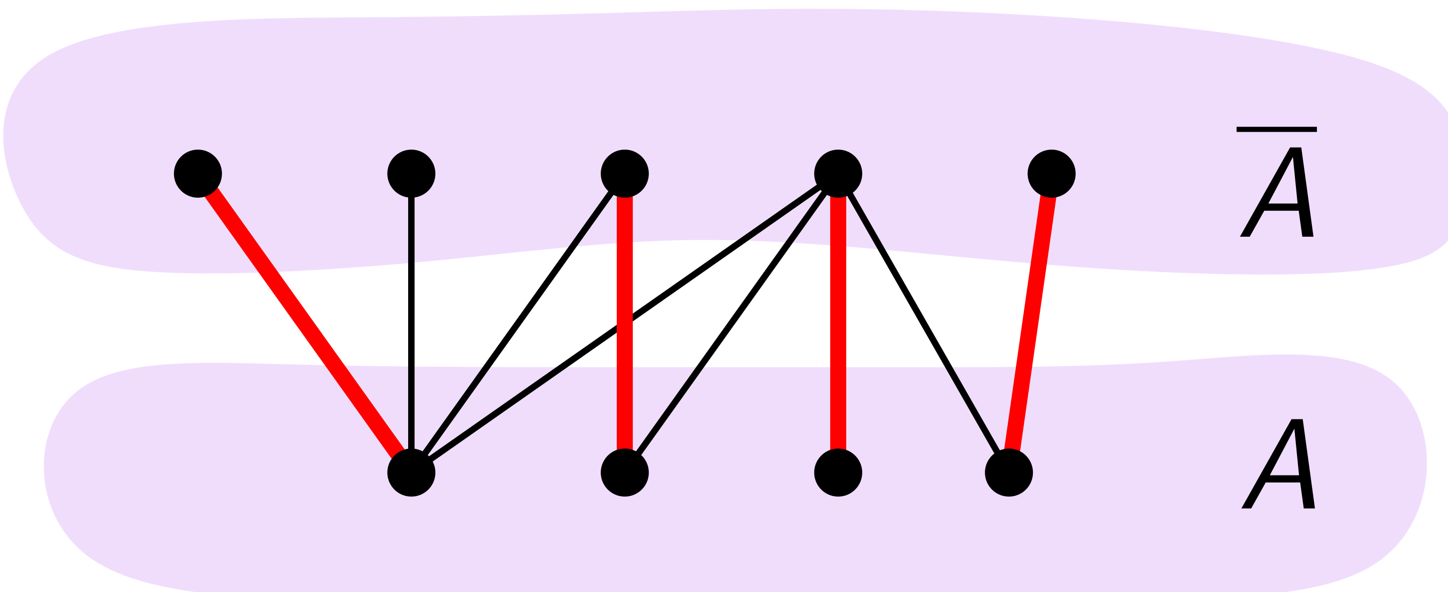
Theorem [Oum, 2006]
For all cut $(A, \overline{A})$, we have $\mathrm{rw}(A) \leq \mathrm{mw}(A) \leq 2^{\mathrm{rw}(A)} + 1$.

Mim-width [Vatshelle, 2012]
Defined from mim(A) := size of a **maximum induced matching** in the bipartite graph between A and $\overline{A}$.



Theorem [Vatshelle, 2012]
For all cut $(A, \overline{A})$, we have $\mathrm{mim}(A) \leq \mathrm{rw}(A)$.

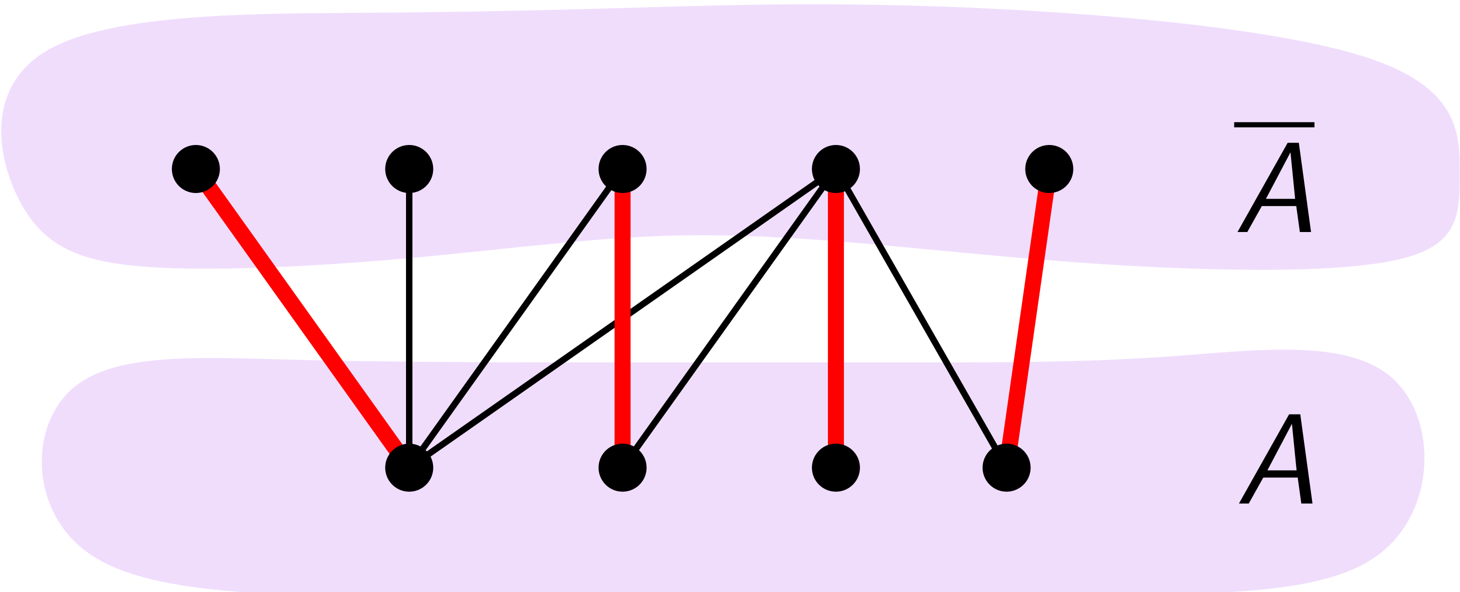
Maximum matching width [Vatshelle, 2012]
Defined from $\mathrm{mmw}(A) :=$ size of a **maximum matching** in the bipartite graph between A and $\overline{A}$.



Linearly equivalent to tree-width!

Theorem [Vatshelle, 2012]
For all graph G, we have $\frac{1}{3}\mathrm{tw}(G) + 1 \leq \mathrm{mmw}(G) \leqslant \mathrm{tw}(G)$.

Maximum matching width [Vatshelle, 2012]
Defined from $\mathrm{mmw}(A) :=$ size of a **maximum matching** in the bipartite graph between A and $\overline{A}$.



Theorem [Vatshelle, 2012]
For all cut $(A, \overline{A})$, we have $\mathrm{mim}(A) \leq \mathrm{rw}(A) \leqslant \mathrm{mmw}(A)$.

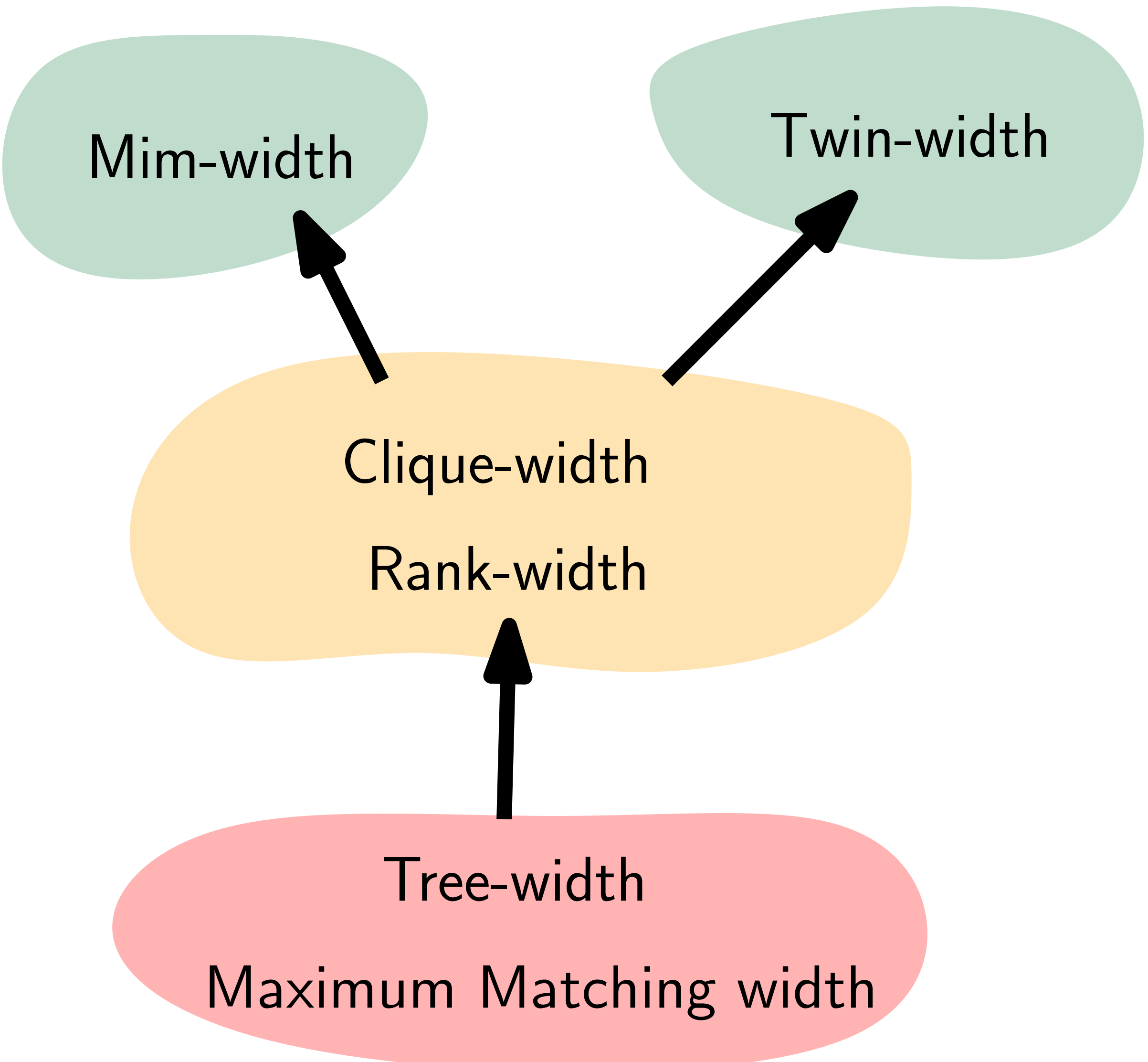
Comparing widths

- **Modeling power**

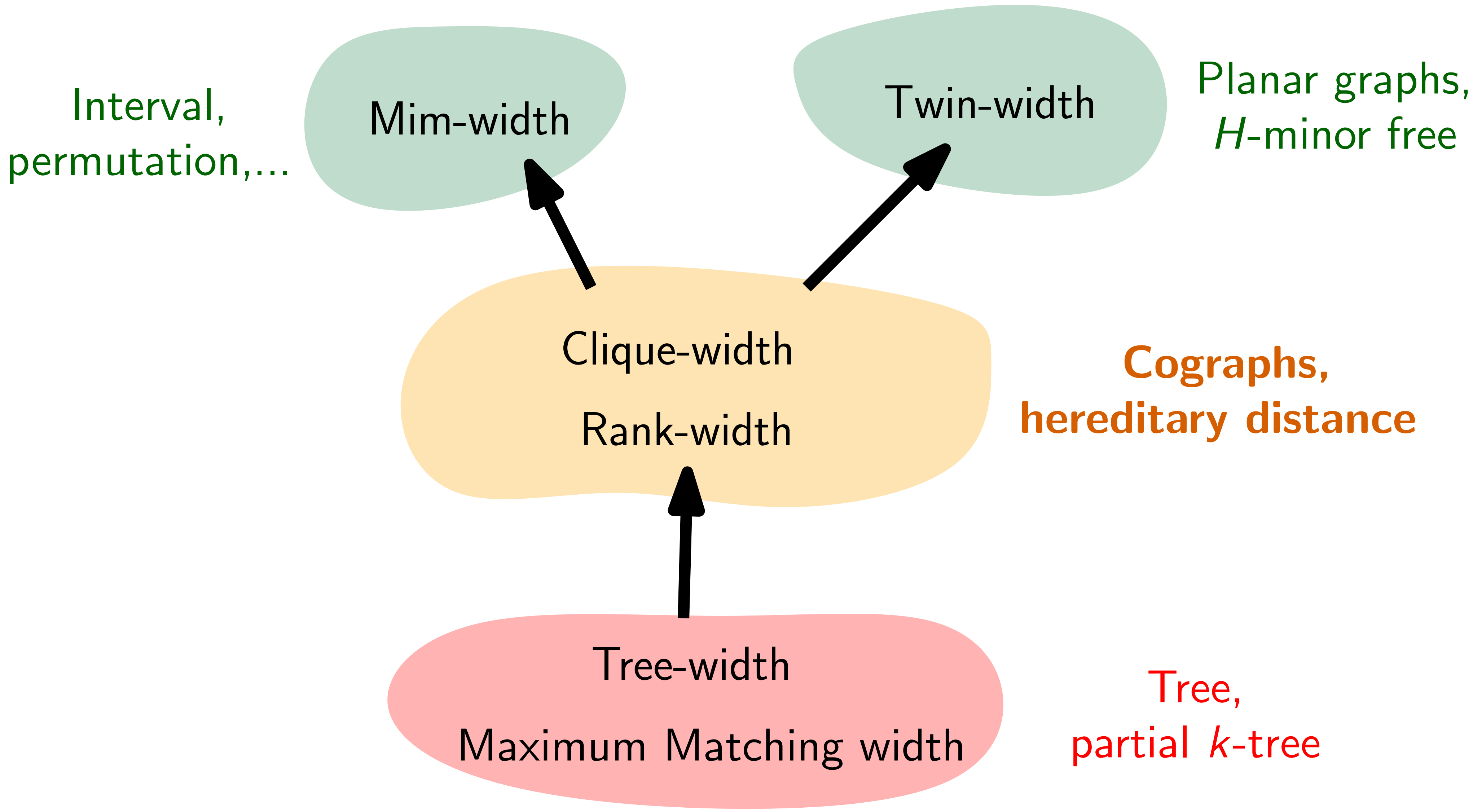
- **Algorithmic applications**

- **Complexity of computing a good decomposition**
  - NP-hardness everywhere
  - We know **efficient FPT** approximation algorithms for **tree-width** and **rank-width**

Modeling Power



Modeling Power



Computing Good Decomposition

Theorem [Oum and Seymour, 2006]
Rank-width can be **3-approximated** in time $8^{\mathrm{rw}} n^{O(1)}$.

Theorem [Korhonen and Fomin, 2021]
Rank-width can be **2-approximated** in time $2^{2^{O(\mathrm{rw})}} n^2$.

$\mathrm{rw}(A)$ is **symmetric** and **submodular**

$$\mathrm{rw}(X) + \mathrm{rw}(Y) \geq \mathrm{rw}(X \cap Y) + \mathrm{rw}(X \cup Y)$$

Meta-Algorithmic Applications

mim-width A&C DN

twin-width FO

clique-width rank-width MSO_1

treewidth mm-width MSO_2

Efficient Algorithms

Theorem [Oum, 2006]
For all cut $(A, \overline{A})$, we have $\mathrm{mw}(A) \leq 2^{\mathrm{rw}(A)} + 1$.

- $2^{O(\mathrm{cw})} n^{O(1)}$ time algo. $\Rightarrow$ $2^{2^{O(\mathrm{rw})}} n^{O(1)}$ time algo.

Efficient Algorithms

Theorem [Oum, 2006]
For all cut $(A, \overline{A})$, we have $\mathrm{mw}(A) \leq 2^{\mathrm{rw}(A)} + 1$.

- $2^{O(\mathrm{cw})} n^{O(1)}$ time algo. $\Rightarrow$ $2^{2^{O(\mathrm{rw})}} n^{O(1)}$ time algo.

Theorem [Bui-Xuan, Telle and Vatshelle, 2010]
Independent Set and **Dominating Set** can be solved in time $2^{O(\mathrm{rw}^2)} n^{O(1)}$.

Efficient Algorithms

Theorem [Oum, 2006]
For all cut $(A, \overline{A})$, we have $\mathrm{mw}(A) \leq 2^{\mathrm{rw}(A)} + 1$.

- $2^{O(\mathrm{cw})} n^{O(1)}$ time algo. $\Rightarrow 2^{2^{O(\mathrm{rw})}} n^{O(1)}$ time algo.

Theorem [Bui-Xuan, Telle and Vatshelle, 2010]
Independent Set and **Dominating Set** can be solved in time $2^{O(\mathrm{rw}^2)} n^{O(1)}$.

Theorem [Ganian and Hliněný 2010]
Feedback Vertex Set can be solved in time $2^{O(\mathrm{rw}^2)} n^{O(1)}$.

Generalizations

Problems that can be solved in time $2^{O(\mathrm{rw}^2)} n^{O(1)}$

Locally Checkable Vertex Subset (LCVS)

Independent Set	Max. Induced Matching
Dominating Set	Perfect Code
Induced Matching	Total Dominating Set

[Bui-Xuan, Telle and Vatshelle, 2013]

Locally Checkable Vertex Partitioning (LCVP)

k-Coloring	Odd Cycle Transversal
H-Homomorphism	Perfect Matching Cut
H-Covering	...

[Bui-Xuan, Telle and Vatshelle, 2013]

Generalizations

Problems that can be solved in time $2^{O(\mathrm{rw}^2)} n^{O(1)}$

Locally Checkable Vertex Subset (LCVS)

Independent Set — Max. Induced Matching
Dominating Set — Perfect Code
Induced Matching — Total Dominating Set

[Bui-Xuan, Telle and Vatshelle, 2013]

Locally Checkable Vertex Partitioning (LCVP)

k-Coloring — Odd Cycle Transversal
H-Homomorphism — Perfect Matching Cut
H-Covering — ...

[Bui-Xuan, Telle and Vatshelle, 2013]

Connected, Acyclic LCVS

Connected, Acyclic LCVP

Connected Dominating Set — Feedback Vertex Set
Connected Vertex Cover — Longest Induced Path

[Bergougnoux and Kante, 2019]

Generalizations

Problems that can be solved in time $2^{O(\mathsf{rw}^2)} n^{O(1)}$

Locally Checkable Vertex Subset (LCVS)

Independent Set, Max. Induced Matching, Dominating Set, Perfect Code, Induced Matching, Total Dominating Set

[Bui-Xuan, Telle and Vatshelle, 2013]

Locally Checkable Vertex Partitioning (LCVP)

k-Coloring, Odd Cycle Transversal, H-Homomorphism, Perfect Matching Cut, H-Covering, $\cdots$

[Bui-Xuan, Telle and Vatshelle, 2013]

A&C DN

[Bergougnoux, Dreier and Jaffke, 2022+]

Connected, Acyclic LCVS

Connected, Acyclic LCVP

Connected Dominating Set, Feedback Vertex Set, Connected Vertex Cover, Longest Induced Path

[Bergougnoux and Kante, 2019]

Lower Bounds

ETH (roughly) [Impagliazzo and Paturi, 2001]
There is no $2^{o(n)} n^{O(1)}$ time algorithm for **3-CNF SAT**.

Linear reductions [Folklore]
Under ETH, there is no $2^{o(n)} n^{O(1)}$ time algorithm for:

- Independent Set
- Dominating Set
- Maximum Induced Matching
- Feedback Vertex Set
- ...

Lower Bounds

ETH (roughly) [Impagliazzo and Paturi, 2001]
There is no $2^{o(n)} n^{O(1)}$ time algorithm for **3-CNF SAT**.

Linear reductions [Folklore]
Under ETH, there is no $2^{o(n)} n^{O(1)}$ time algorithm for...

$$\mathrm{tw}, \mathrm{cw}, \mathrm{rw} \leqslant n$$

Corrolary
For each $k \in \{\mathrm{tw}, \mathrm{cw}, \mathrm{rw}\}$, under ETH, there is no $2^{o(k)} n^{O(1)}$ time algorithm for

- Independent Set
- ...

Results on Independet Set

Best known:	Upper bound	ETH lower bound
tree-width clique-width	$2^{O(k)} n^{O(1)}$ [Folklore]	$2^{o(k)} n^{O(1)}$ [Folklore]
rank-width	$2^{O(k^2)} n^{O(1)}$ [Bui-Xuan et al., 2012]	$2^{o(k)} n^{O(1)}$ [Folklore]
mim-width	$n^{O(k)}$ [Bui-Xuan et al., 2013]	$n^{o(k/\log k)}$ [Bakkane and Jaffke, 2022+]

Results on Independet Set

Best known:	Upper bound	ETH lower bound
tree-width clique-width	$2^{O(k)} n^{O(1)}$ [Folklore]	$2^{o(k)} n^{O(1)}$ [Folklore]
rank-width	$2^{O(k^2)} n^{O(1)}$ [Bui-Xuan et al., 2012]	$2^{o(k^2)} n^{O(1)}$ [Us, 2022+]
mim-width	$n^{O(k)}$ [Bui-Xuan et al., 2013]	$n^{o(k/\log k)}$ [Bakkane and Jaffke, 2022+]

Our results

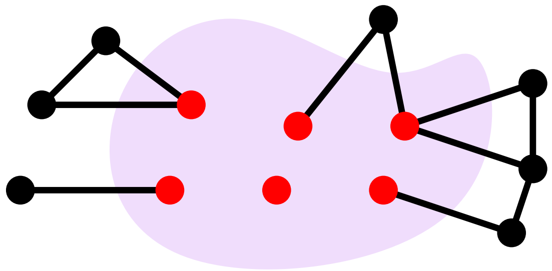
Theorem [B., Korhonen and Nederlof, 2022+]
Under ETH, there are no $2^{o(\mathrm{rw}^2)} n^{O(1)}$ time algorithms for

- Independent Set
- **Weighted** Dominating Set
- Maximum Induced Matching
- Feedback Vertex Set

The best known algorithms for these problems are **optimal** under ETH.

Holds also for **linear** rank-width

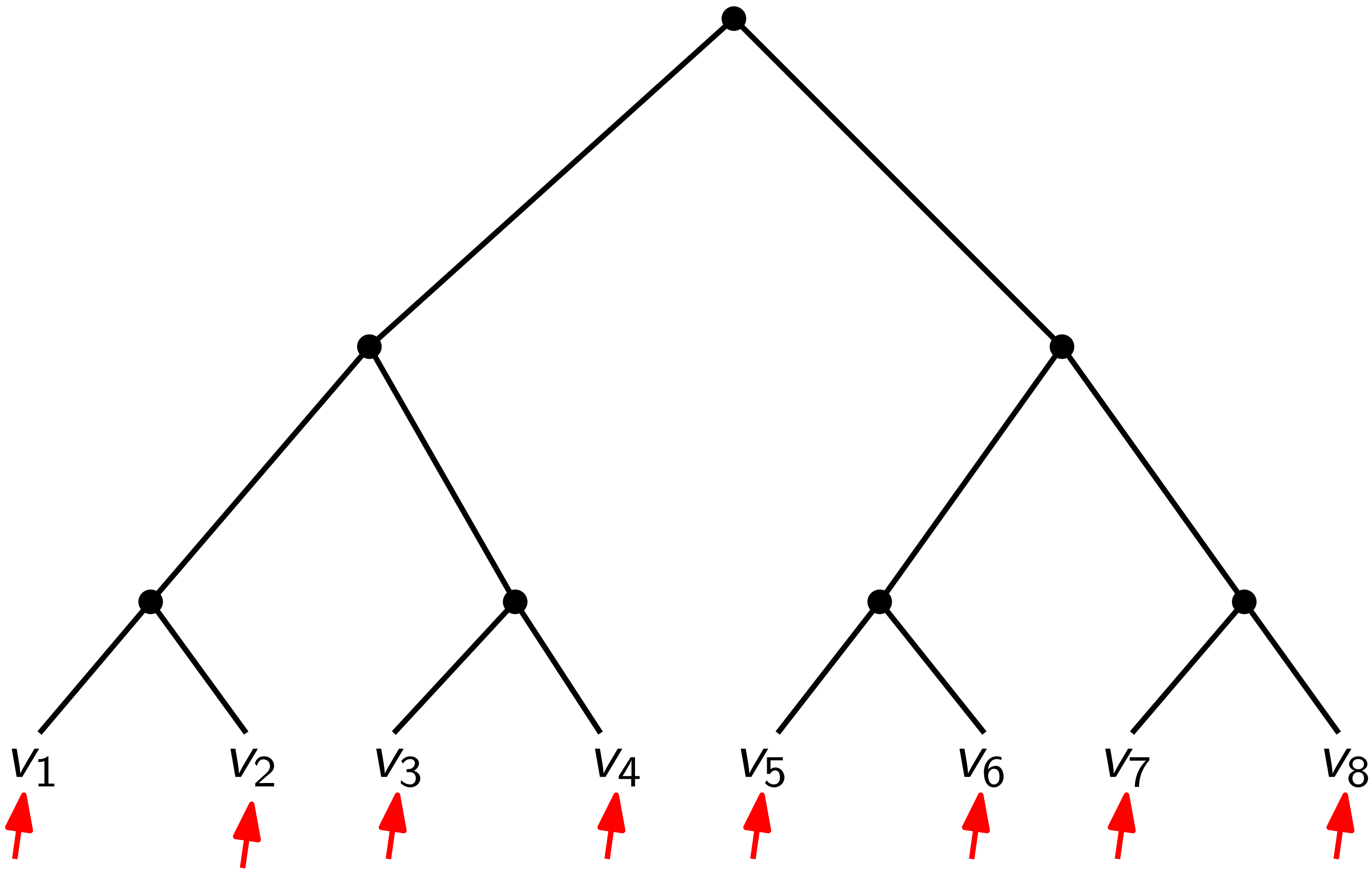
Algo. for Independent Set

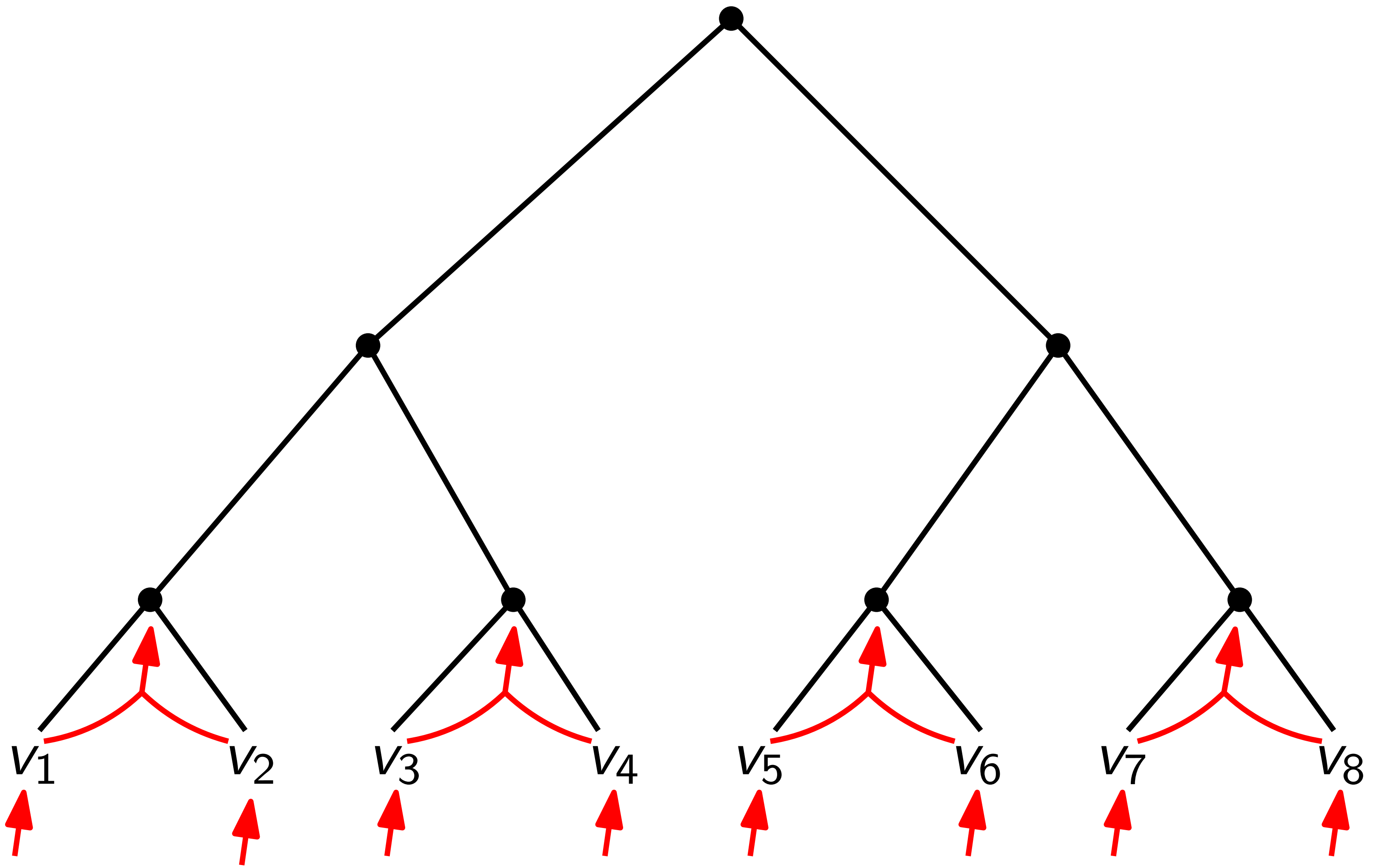


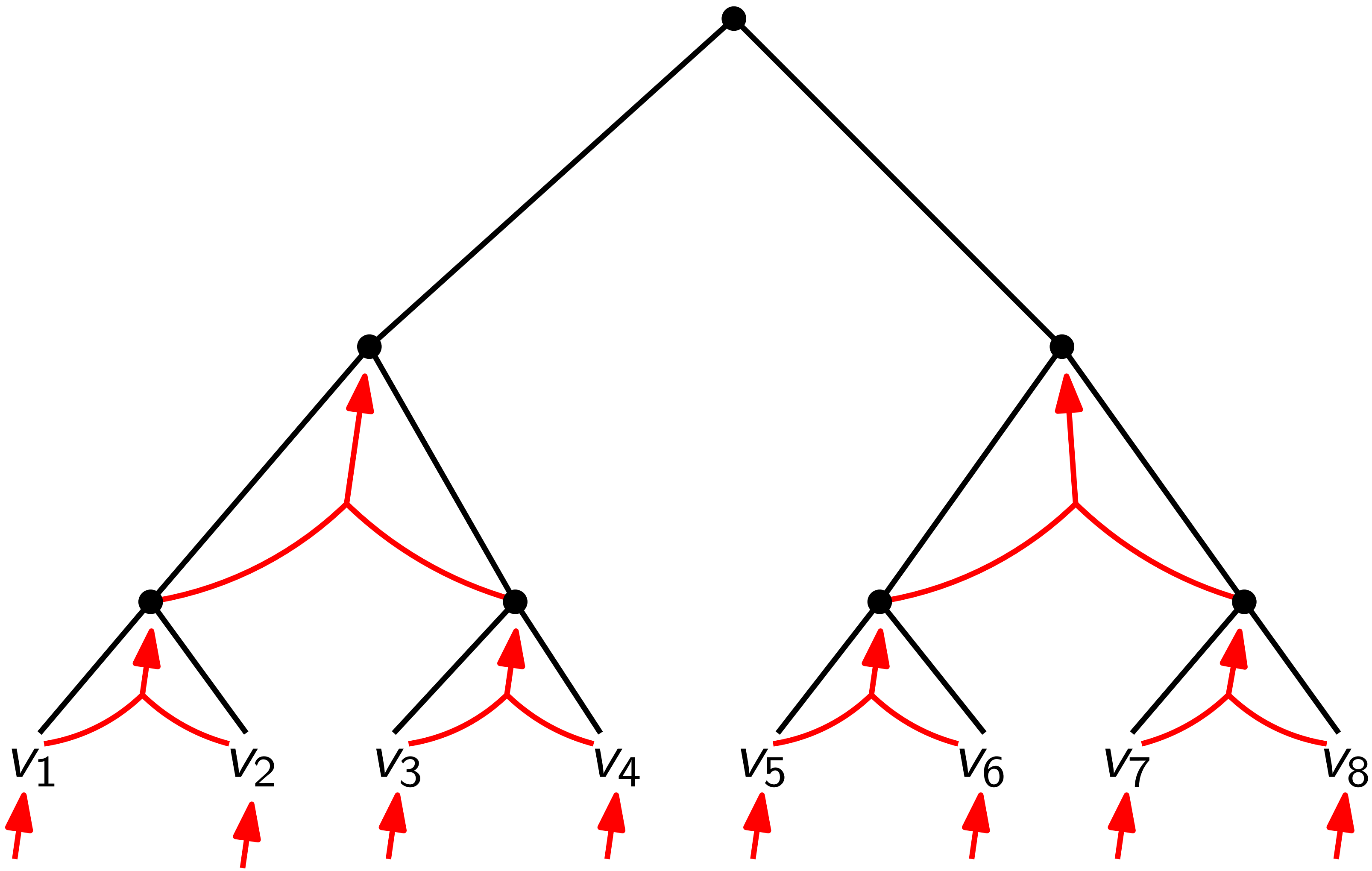
Theorem [Bui-Xuan, Telle and Vatshelle, 2010]
Independent Set can be solved in time $2^{O(\mathrm{rw}^2)} n^{O(1)}$.

Theorem [Bui-Xuan, Telle and Vatshelle, 2013]
There is an algorithm for IS whose runtime is upper bounded by

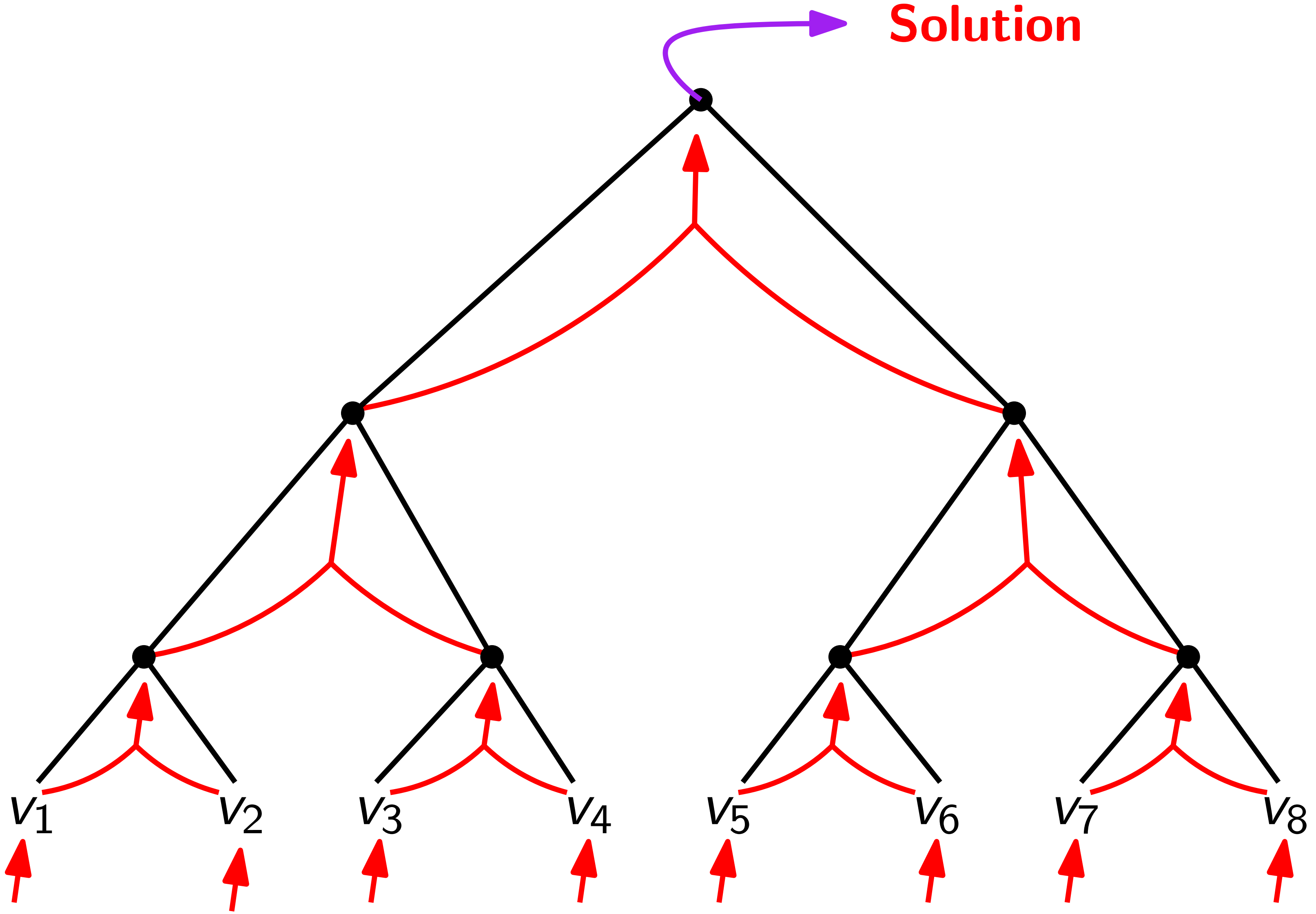
- $2^{O(\mathrm{tw})} \cdot n^{O(1)}$
- $2^{O(\mathrm{cw})} \cdot n^{O(1)}$
- $2^{O(\mathrm{rw}^2)} \cdot n^{O(1)}$
- $n^{O(\mathrm{mim})}$


v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8


v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8

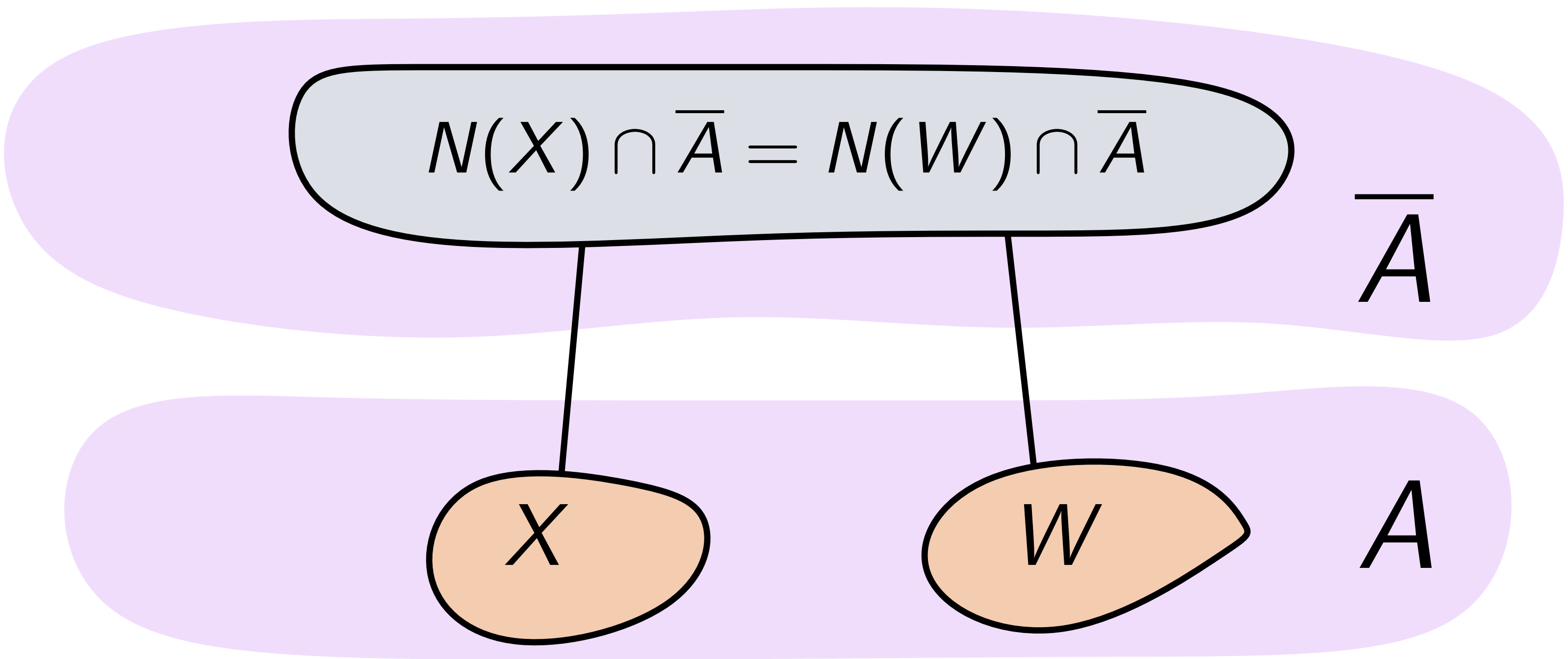


v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8

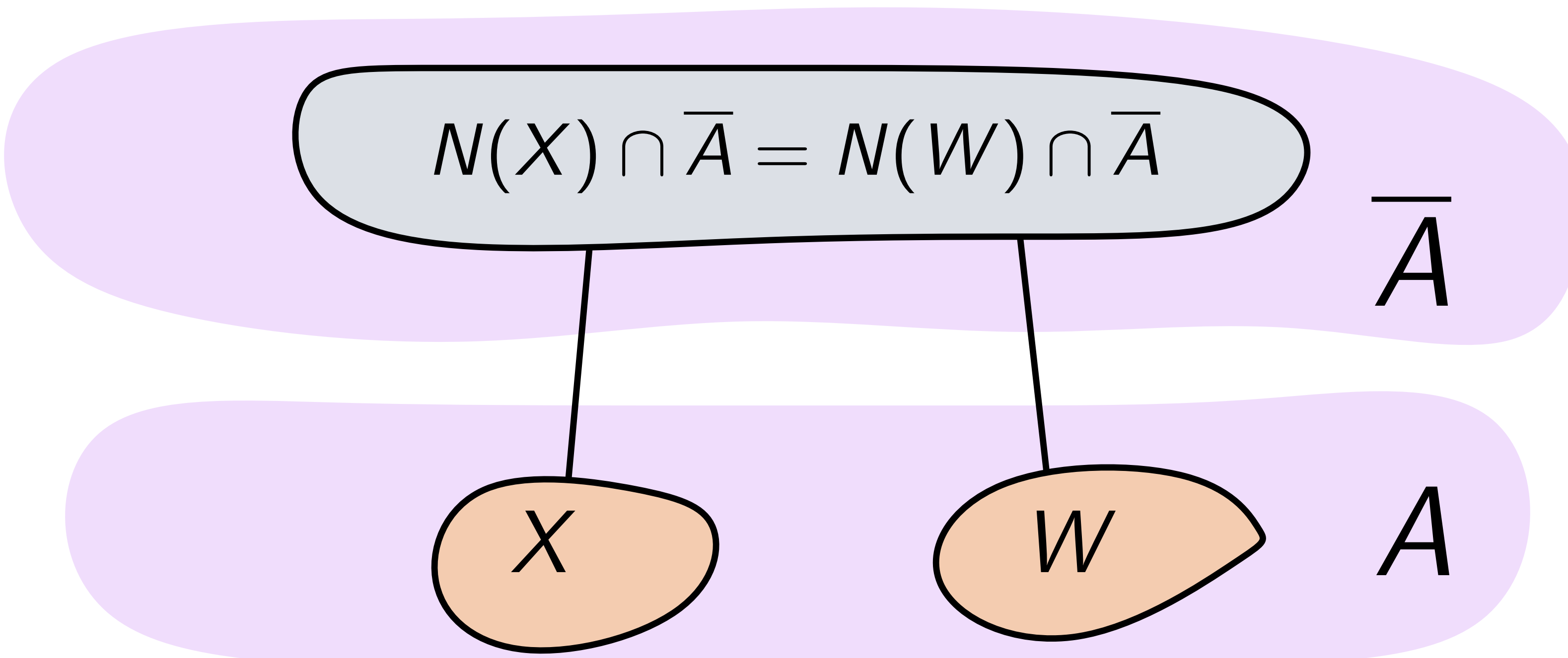


Solution
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8

- For a cut $(A, \overline{A})$, partial solutions = **independent sets** of $G[A]$
- Two partial solutions $X, W \subseteq A$ are **equivalent** if



- For a cut $(A, \overline{A})$, partial solutions $=$ **independent sets** of $G[A]$
- Two partial solutions $X, W \subseteq A$ are **equivalent** if



For every pair X, W of **equivalent** partial solutions and $Y \subseteq \overline{A}$

$$X \cup Y \text{ is a solution} \iff W \cup Y \text{ is a solution}$$

For every cut $(A, \overline{A})$ and each **equivalence class** $\mathcal{C}$
compute a partial solution $X \in \mathcal{C}$ **of maximum size**.

Theorem [Vatshelle, 2013]
The nb. of **eq. classes** $|N(X) \cap \overline{A} \mid X \subseteq A\}|$ is at most

- $2^{\mathrm{mmw}(A)}$
- $2^{\mathrm{mw}(A)}$
- $2^{\mathrm{rw}(A)^2}$
- $n^{\mathrm{mim}(A)}$

For every cut $(A, \overline{A})$ and each **equivalence class** $\mathcal{C}$
compute a partial solution $X \in \mathcal{C}$ **of maximum size**.

Theorem [Bui-Xuan, Telle and Vatshelle, 2013]

The running time of this algorithm is upper bounded by

- $2^{O(\mathrm{tw})} \cdot n^{O(1)}$
- $2^{O(\mathrm{cw})} \cdot n^{O(1)}$
- $2^{O(\mathrm{rw}^2)} \cdot n^{O(1)}$
- $n^{O(\mathrm{mim})}$

This is tight under ETH for clique-width, tree-width and **rank-width**!

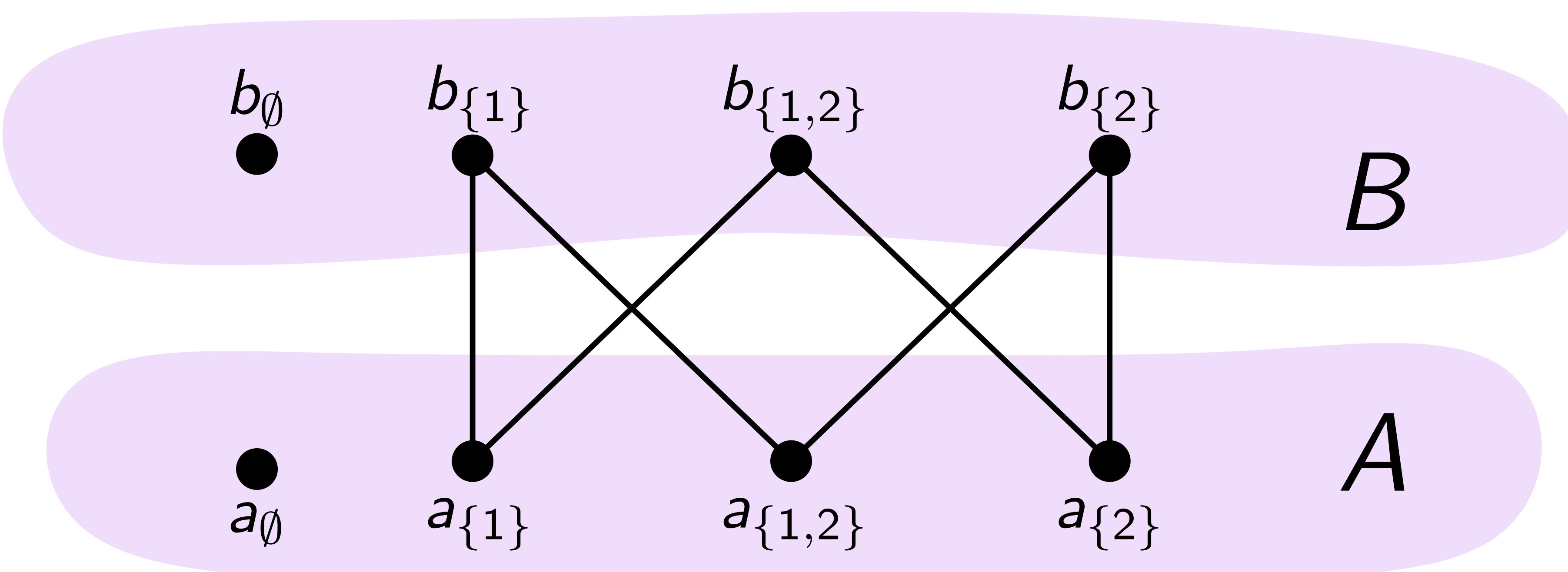
Lower bound

Theorem [B., Korhonen and Nederlof, 2022+]
Under ETH, there are no $2^{o(\mathrm{rw}^2)} n^{O(1)}$ time algorithms for Independent Set

Universal rank cuts

Universal $2k$-rank cut

- $A := \{a_s \mid s \subseteq [2k]\}$
- $B := \{b_s \mid s \subseteq [2k]\}$
- a_s and b_t are adjacent if and only if $|s \cap t|$ is **odd**



Universal rank cuts

The **universal $2k$-rank cut** has rank-width $2k$

Theorem [Bui-Xuan, Telle and Vatshelle, 2010]
The **universal $2k$-rank cut** is the unique (inclusion-wise) maximal cut of rank $2k$ with no twin vertices

Universal rank cuts

The **universal $2k$-rank cut** has rank-width $2k$

Theorem [Bui-Xuan, Telle and Vatshelle, 2010]
The **universal $2k$-rank cut** is the unique (inclusion-wise) maximal cut of rank $2k$ with no twin vertices

Theorem [Bui-Xuan, Telle and Vatshelle, 2011]

$$|\{N(X) \cap B \mid X \subseteq A\}| = 2^{\Omega(k^2)}$$

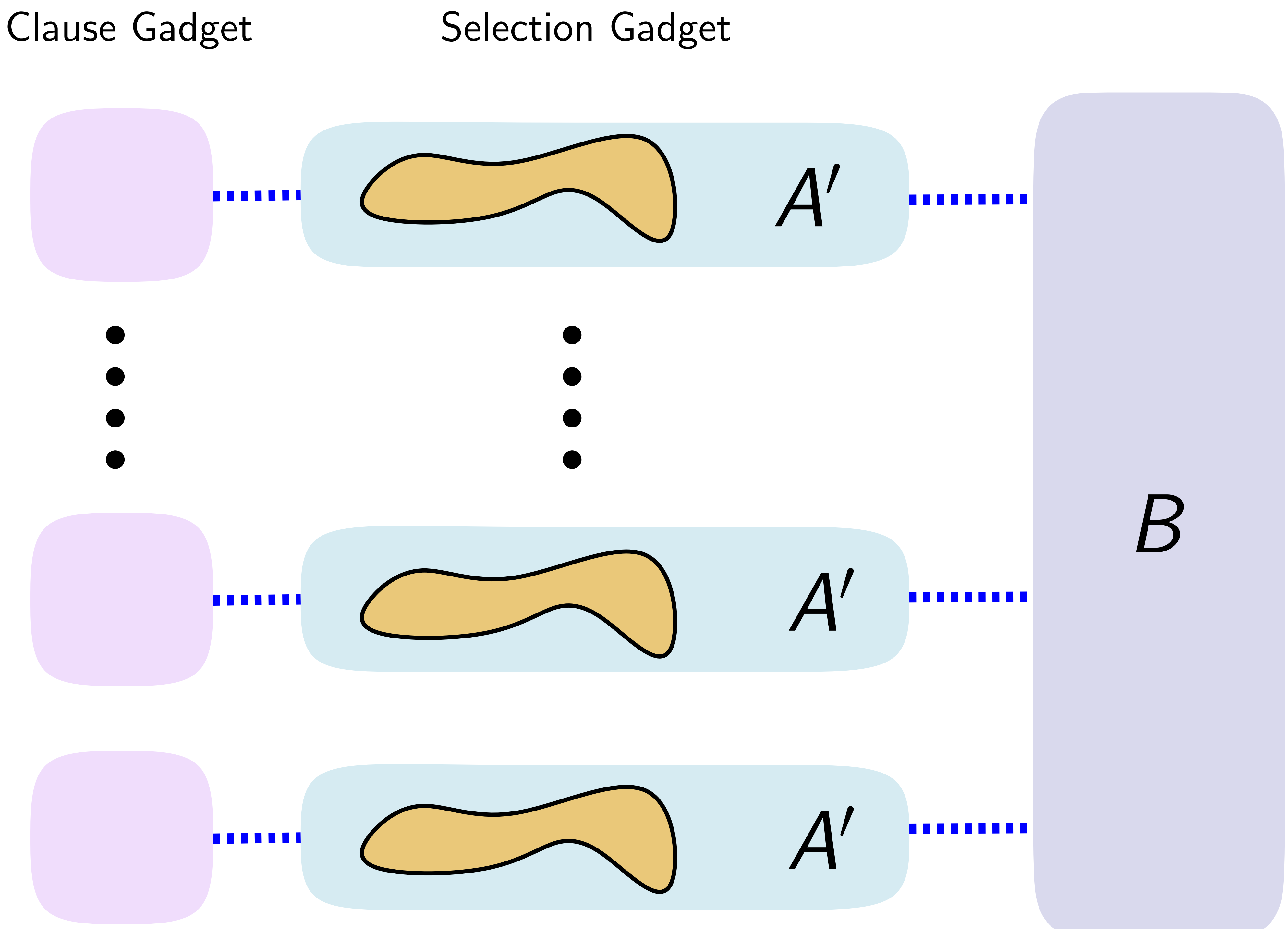
Overview

Reduction from 3-CNF SAT with k^2 **variables**

Lemma
Under ETH, there is no $2^{o(k^2)}(k+m)^{O(1)}$ time algorithm for 3-CNF SAT with k^2 variables

Universal $2k$-rank cut

- $A := \{a_s \mid s \subseteq [2k]\}$
- $B := \{b_s \mid s \subseteq [2k]\}$
- a_s and b_t are adjacent if and only if $|s \cap t|$ is **odd**


Clause Gadget
Selection Gadget
A'
A'
A'
B

Selection Gadget

$$\mathrm{var}(\varphi) := \{v_{i,j} \mid i \in [k] \wedge j \in [k+1, 2k]\}$$

with $k = 3$

$$\begin{array}{ccc} v_{1,4} & v_{1,5} & v_{1,6} \\ \hdashline v_{2,4} & v_{2,5} & v_{2,6} \\ \hdashline v_{3,4} & v_{3,5} & v_{3,6} \end{array}$$

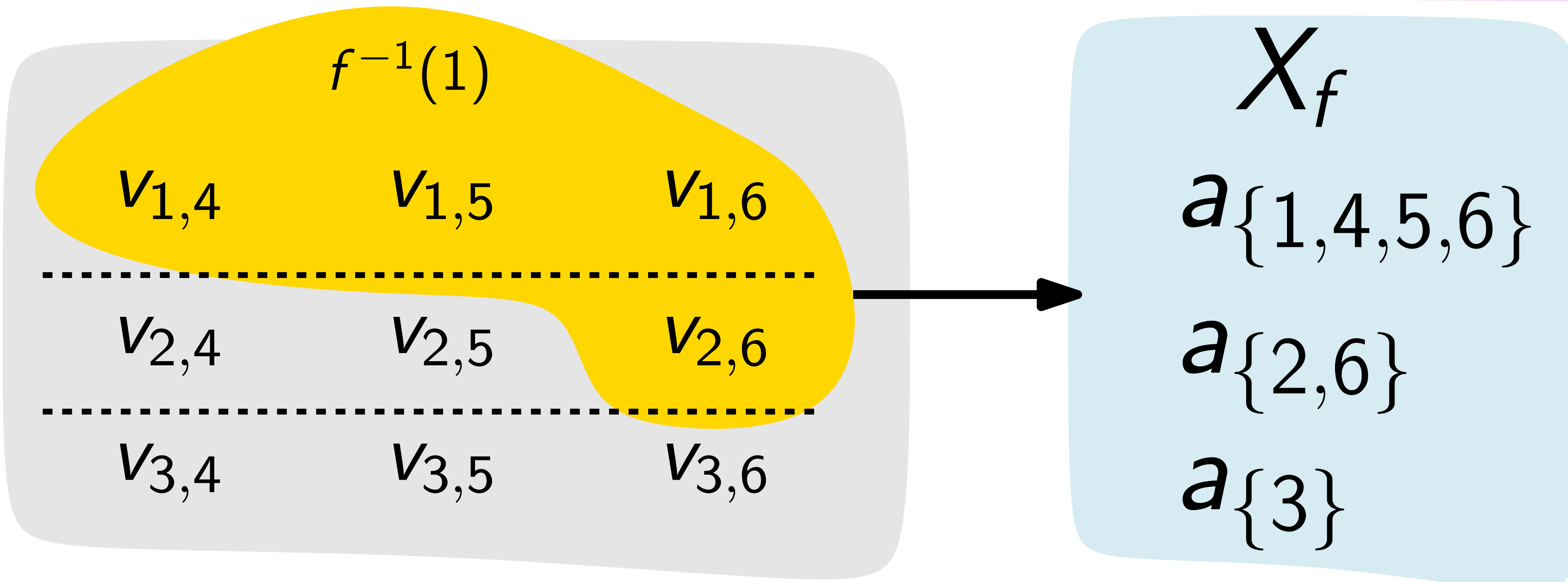
Every **interpretation** $f : \mathrm{var}(\varphi) \to \{0, 1\}$ is associated with $X_f \subseteq A$

$$X_f = \{a_{s_1}, \ldots, a_{s_k}\}$$

$$s_i = \{i\} \cup \{j \in [k+1, 2k] \mid f(v_{i,j}) = 1\}$$

Selection Gadget

$$\mathrm{var}(\varphi) := \{v_{i,j} \mid i \in [k] \wedge j \in [k+1, 2k]\}$$



Every **interpretation** $f : \mathrm{var}(\varphi) \to \{0, 1\}$ is associated with $X_f \subseteq A$

$$X_f = \{a_{s_1}, \ldots, a_{s_k}\}$$

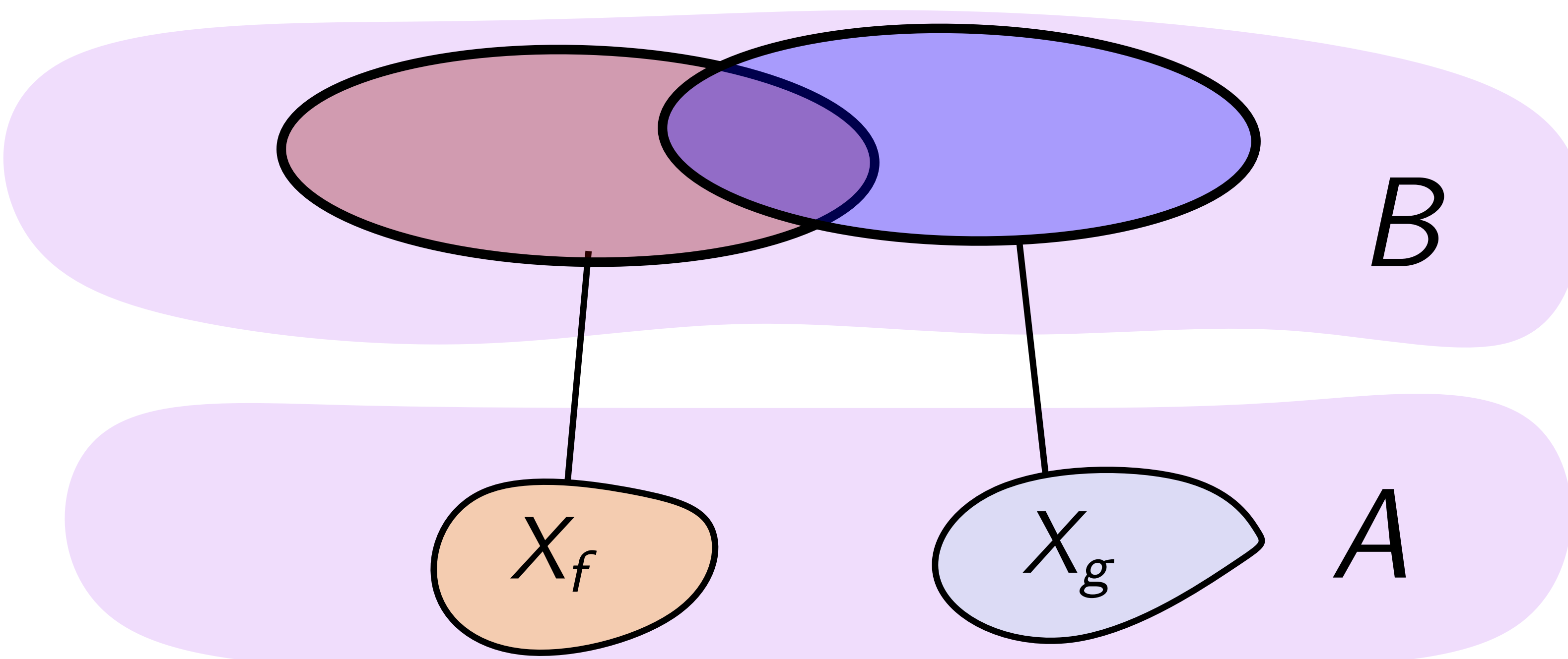
$$s_i = \{i\} \cup \{j \in [k+1, 2k] \mid f(v_{i,j}) = 1\}$$

Selection Gadget

Lemma

For every pair of **distinct interpretations** f, g, the **neighborhoods** of X_f and X_g

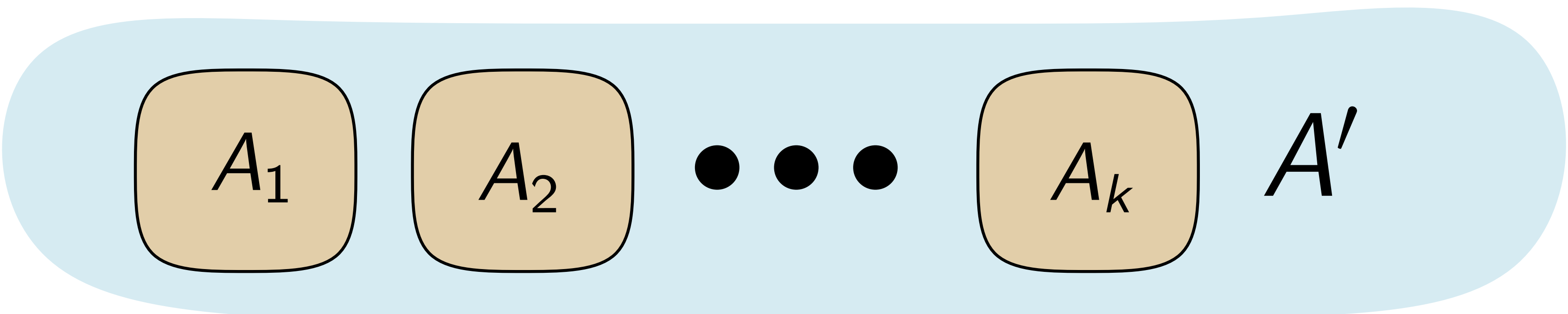
- are different
- have the same size $(2^{2k} - 2^k)$



Selection Gadget

For every $i \in [k]$, let $A_i = \{a_s \in A^{2k} \mid s \cap [k] = \{i\}\}$ and

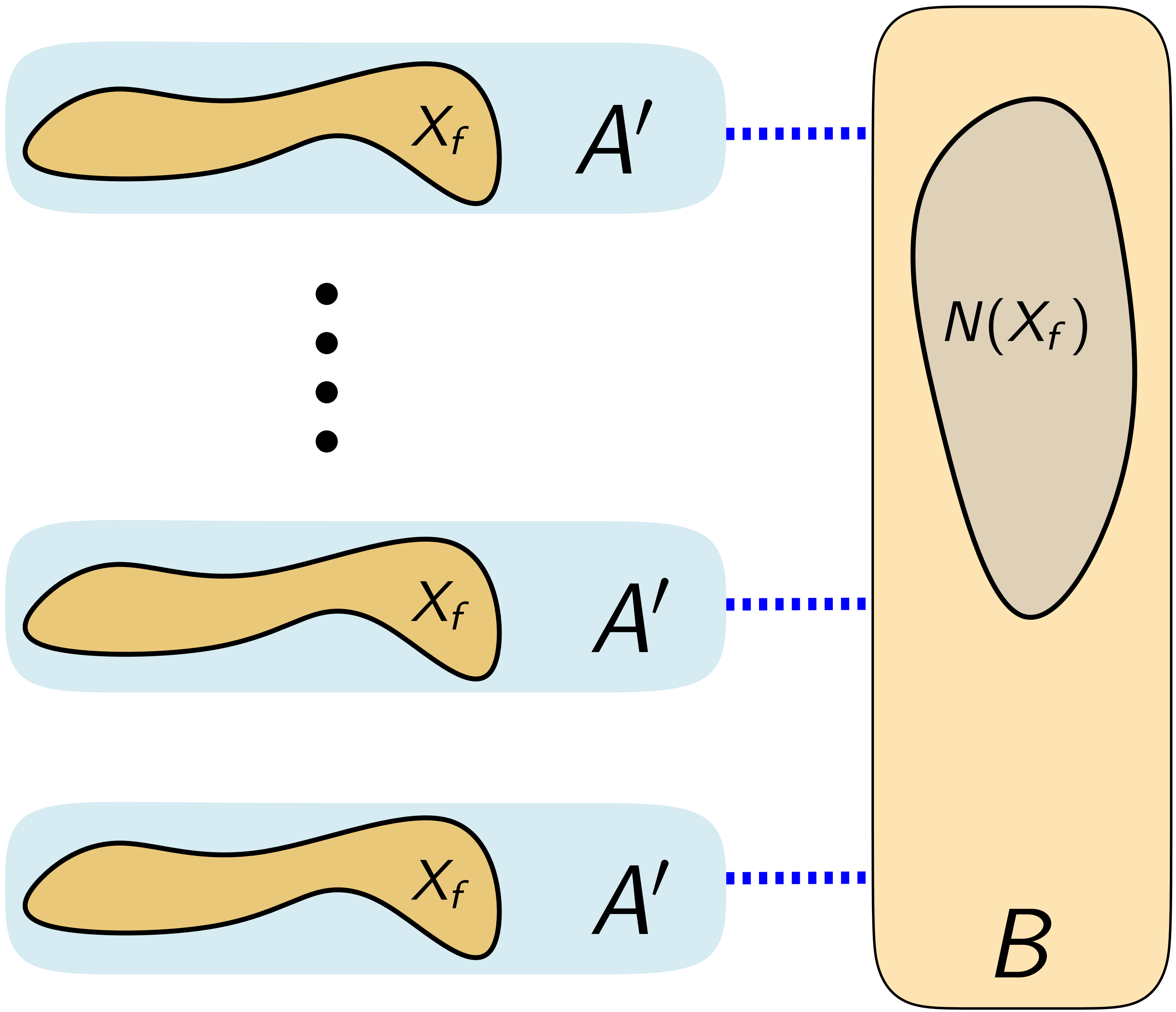
$$A' = A_1 \cup \cdots \cup A_k.$$



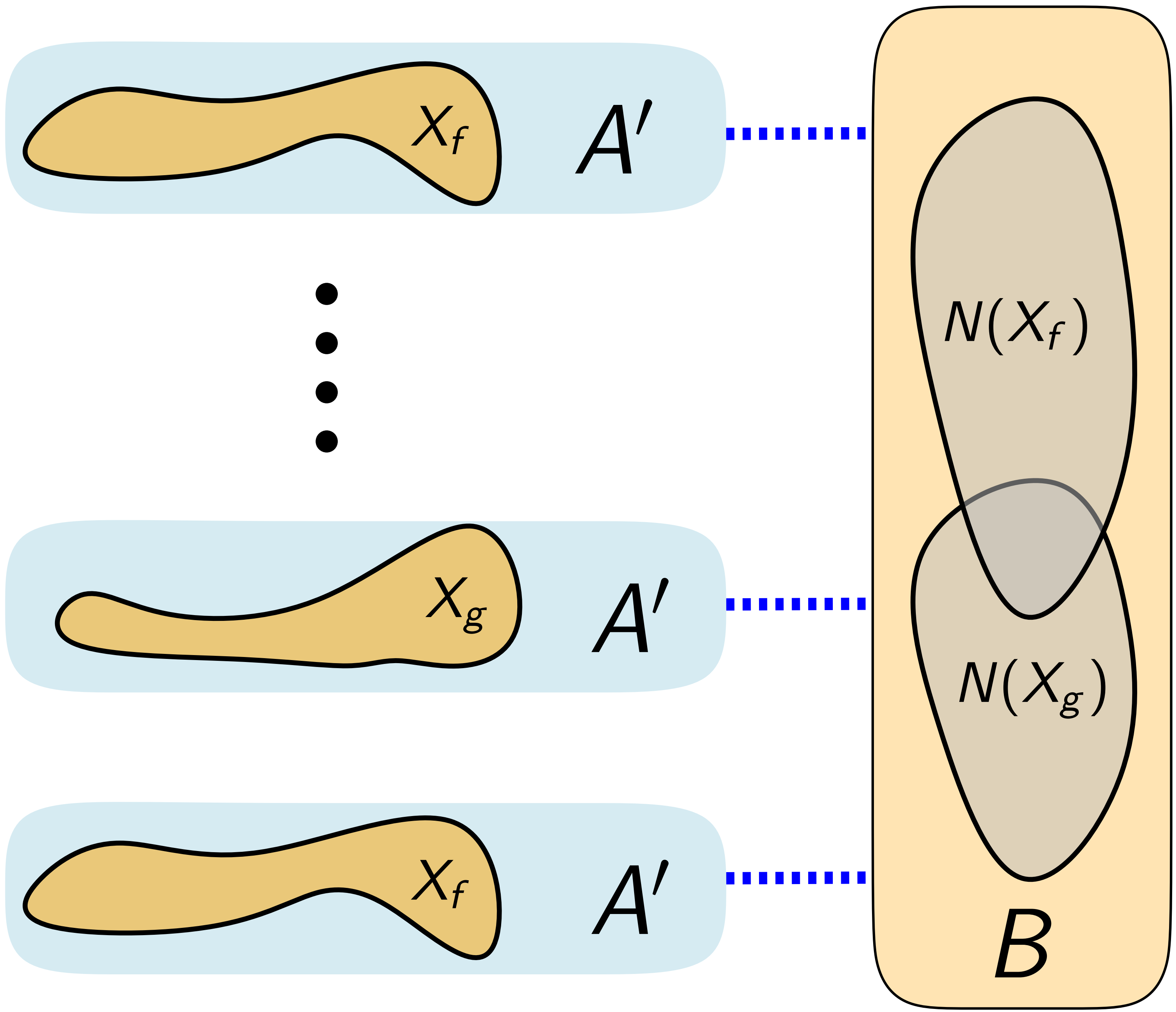
Lemma

Every maximal independent set of $G[A']$ is of the form X_f with f an **interpretation**

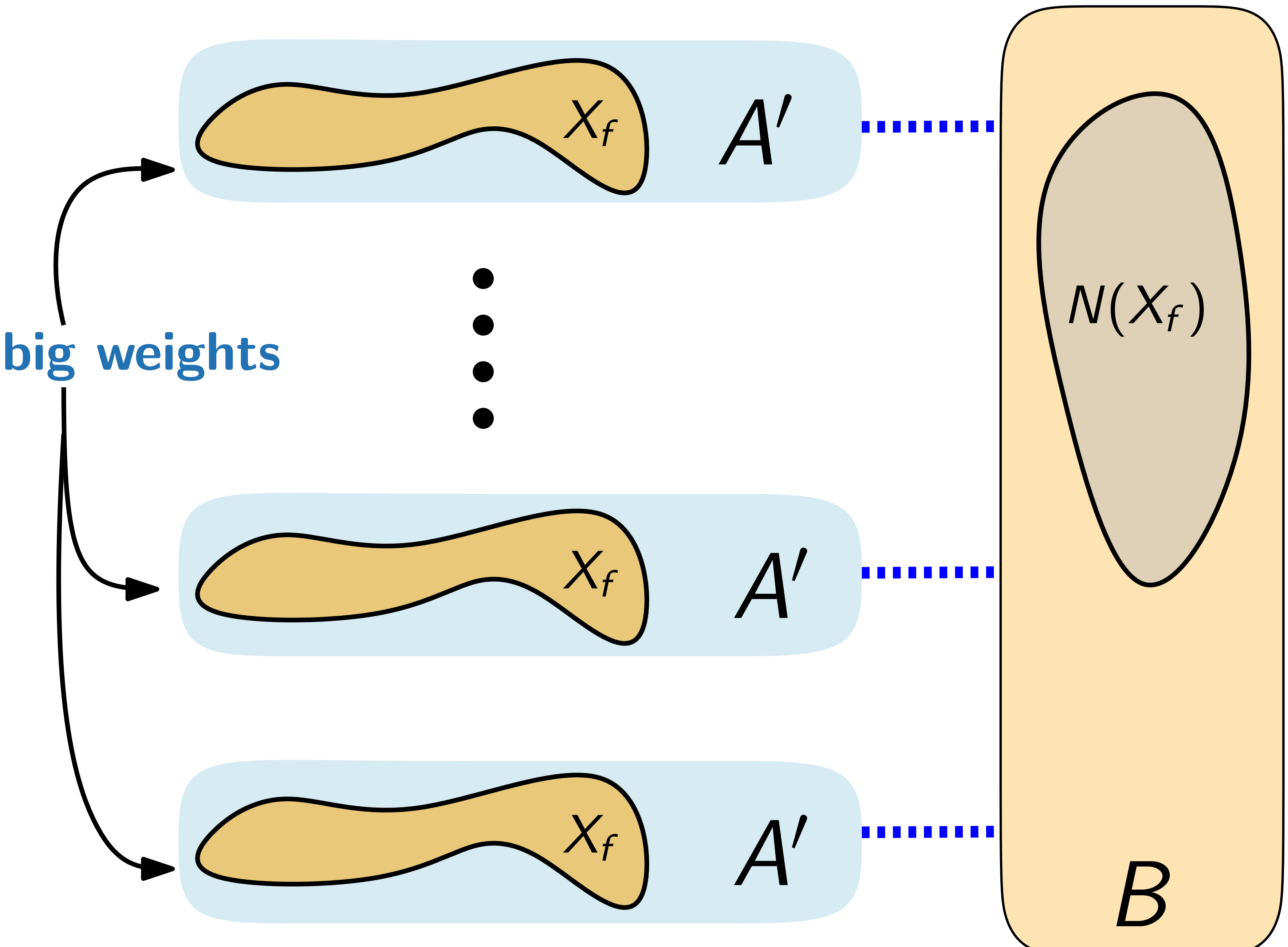
Selection Gadget



Selection Gadget

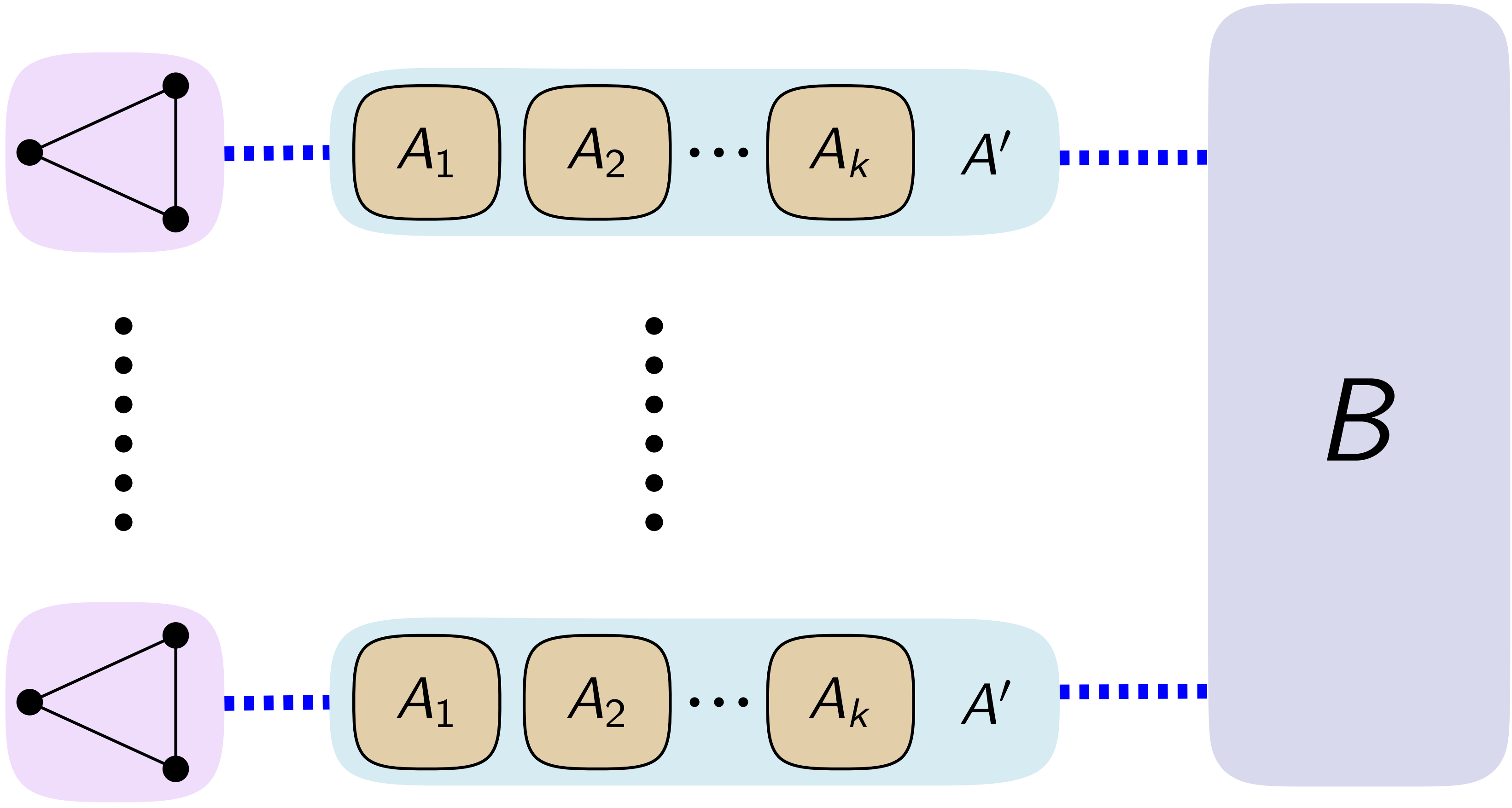


Selection Gadget



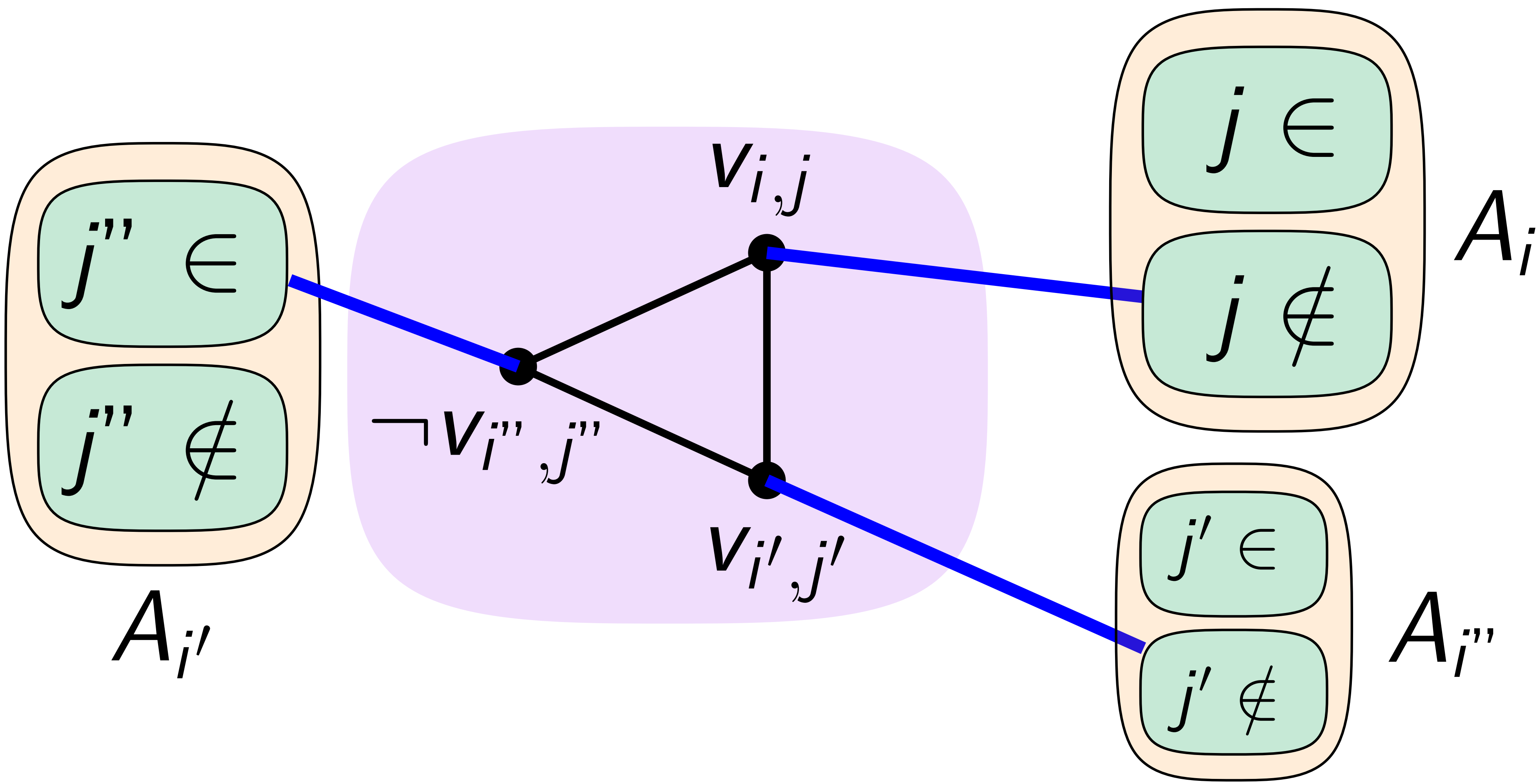
Clause Gadget

Clause gadget $=$ a **triangle** and some edges with a copy of A'



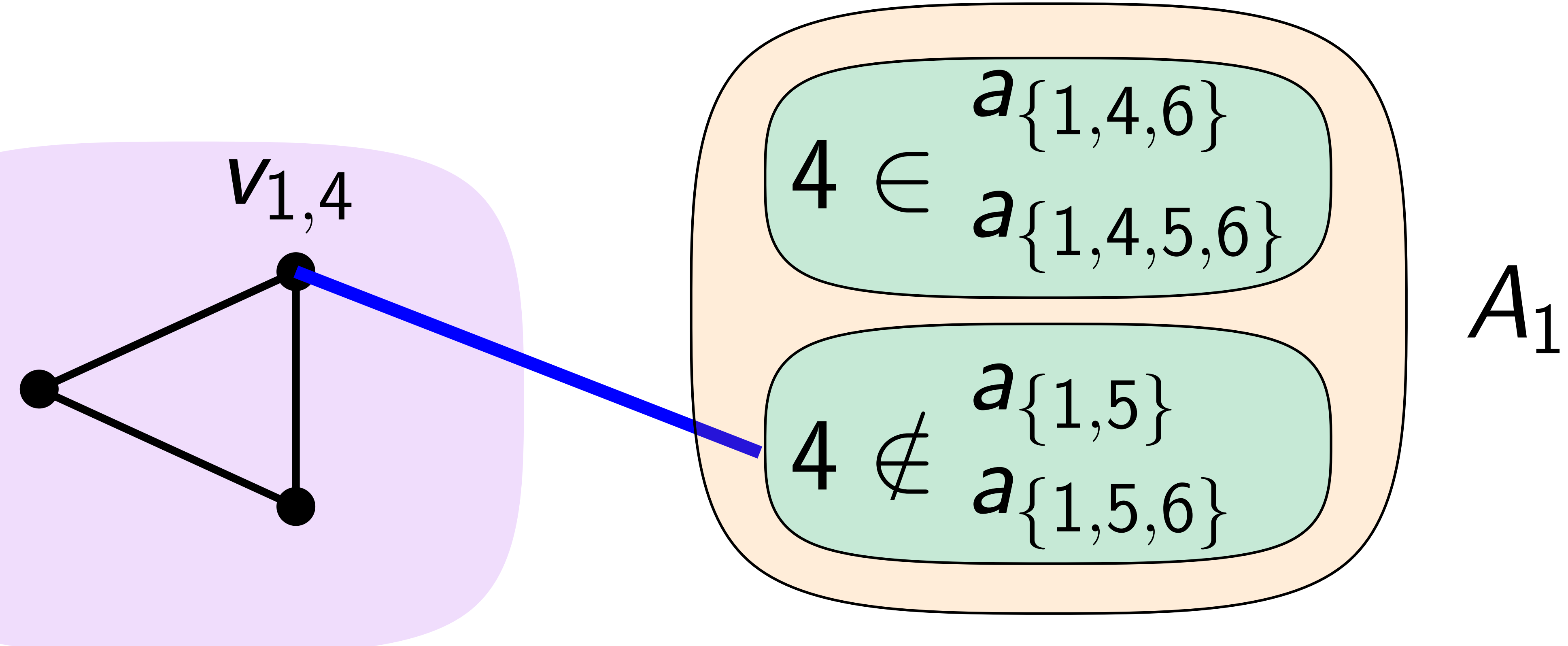
Clause Gadget

Clause gadget = a **triangle** and some edges with a copy of A'



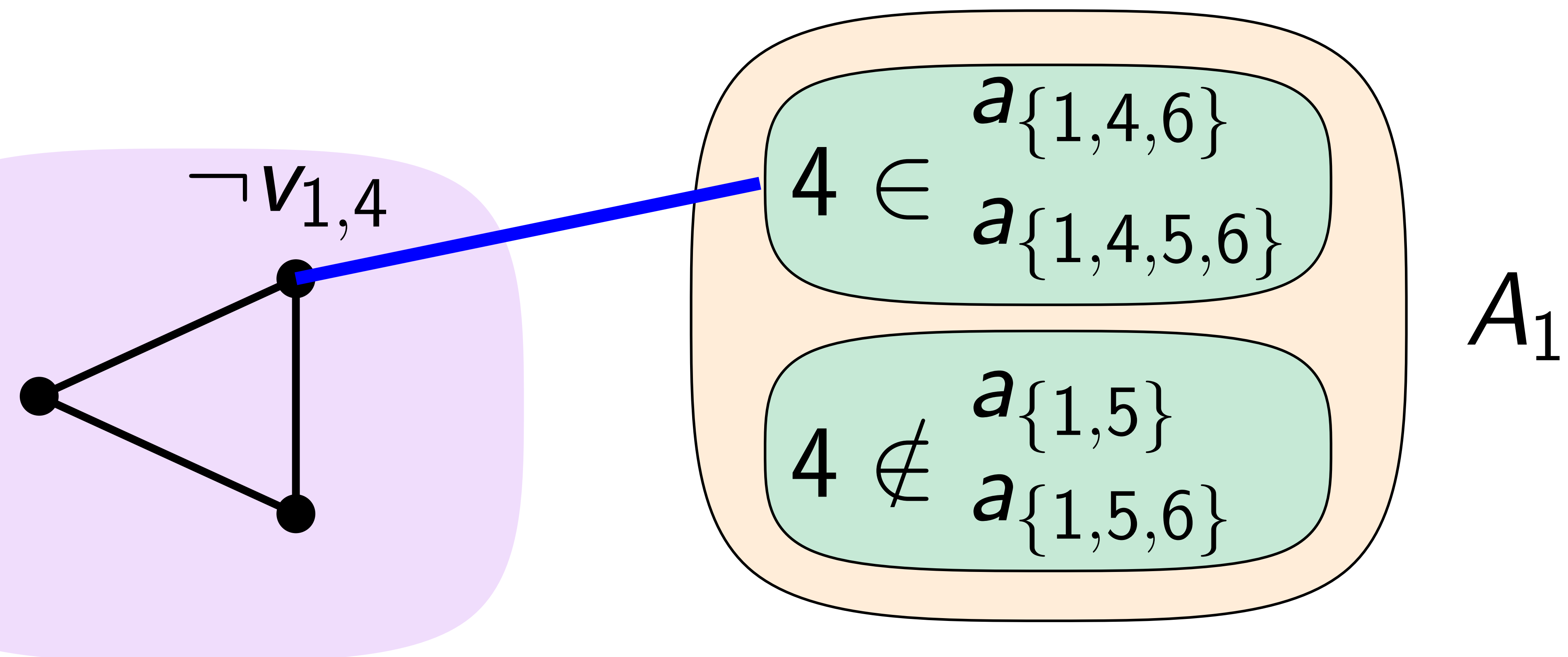
Clause Gadget

Clause gadget = a **triangle** and some edges with a copy of A'



Clause Gadget

Clause gadget = a **triangle** and some edges with a copy of A'

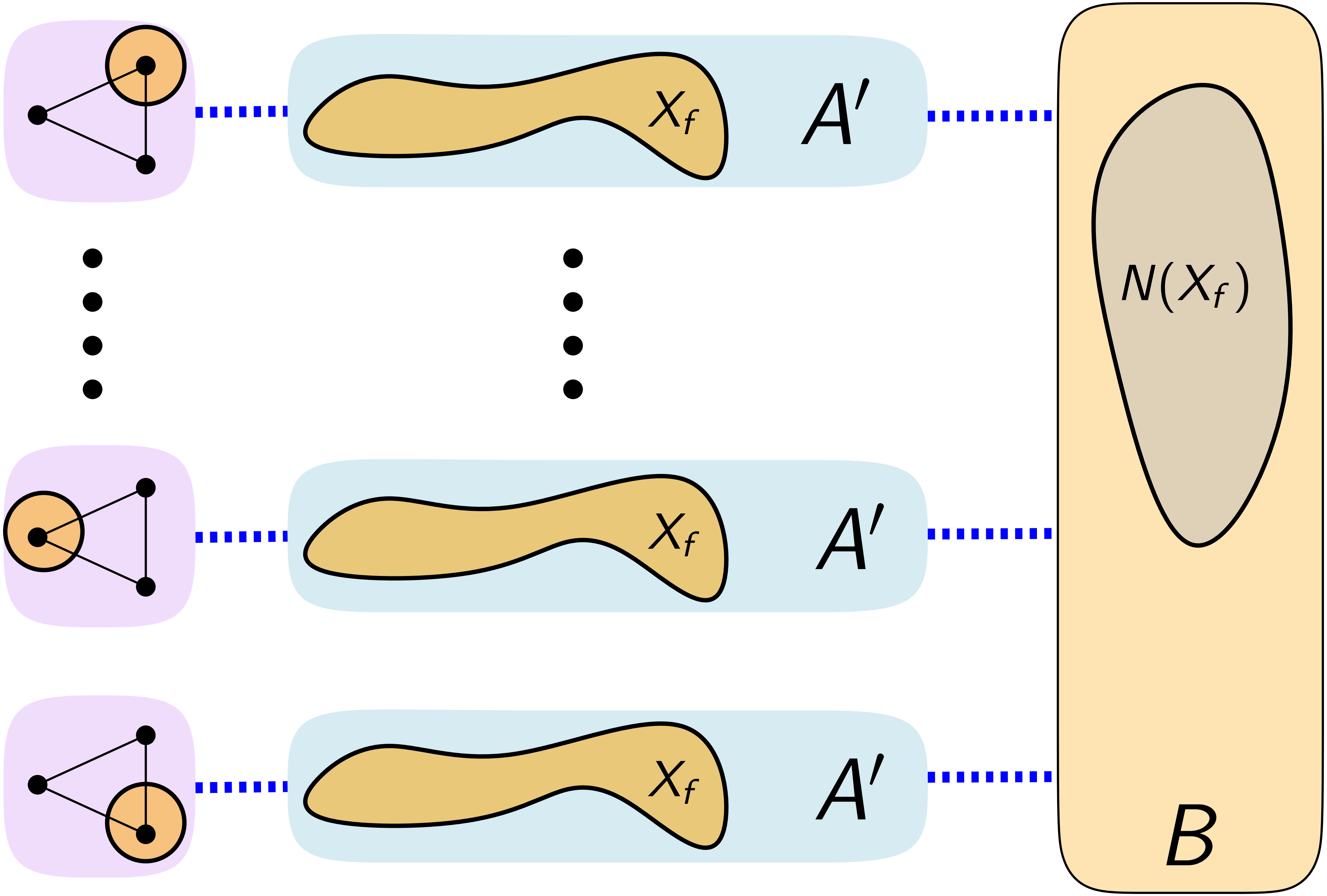


Clause Gadget

Clause gadget = a **triangle** and some edges with a copy of A'

Lemma
f satisfies C **iff** X_f can be completed with one vertex from the clause gadget

φ is satisfiable **iff** there exists an independent set of weight W



Our result

- The **linear rank-width** of this graph is at most $2k+4$.
- This graph has $2^{O(k)}m$ vertices and can be constructed in $2^{O(k)}m$ time.

Theorem [B., Korhonen and Nederlof, 2022+]
Under ETH, there are no $2^{o(\mathrm{rw}^2)}n^{O(1)}$ time algorithms for Independent Set

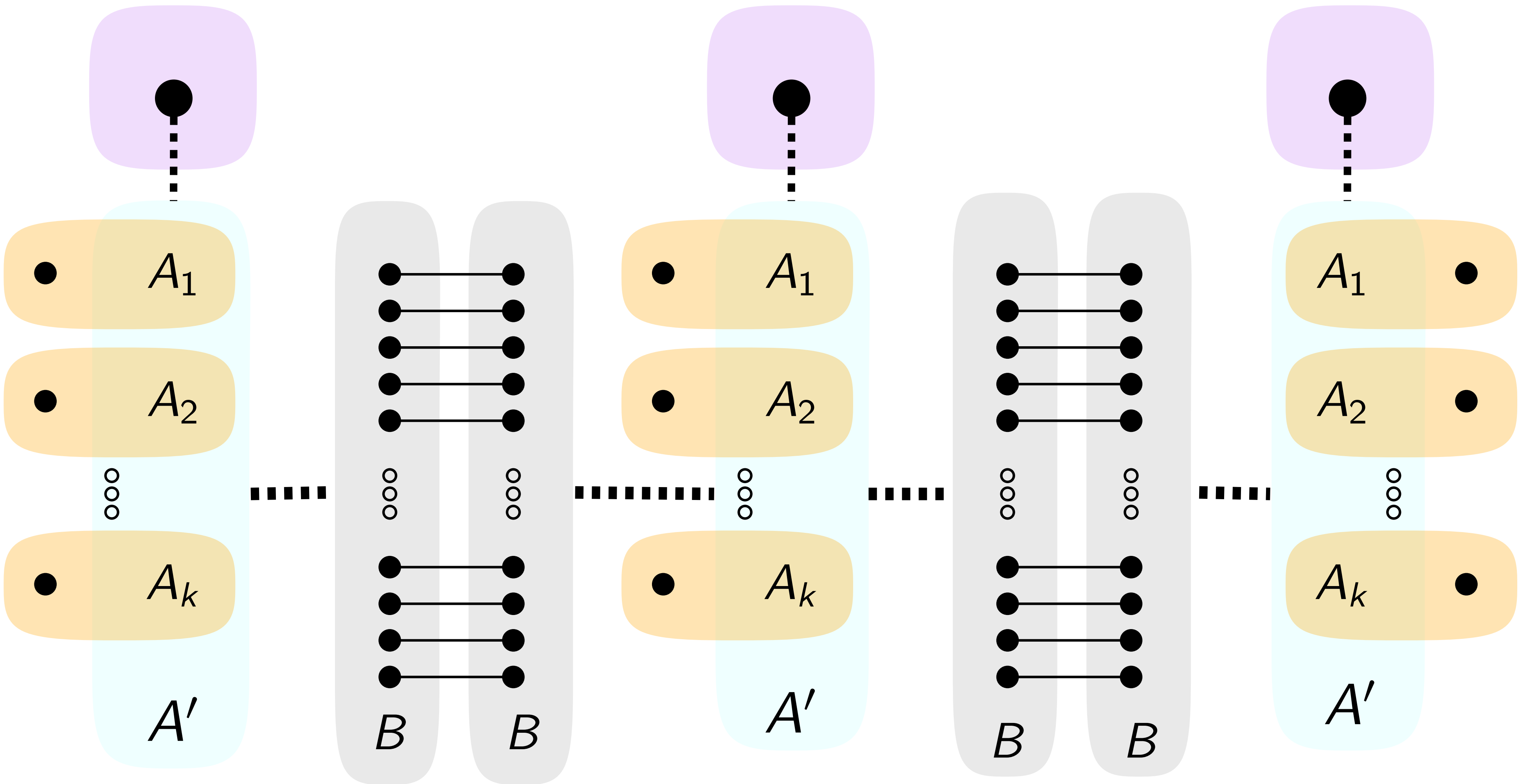
Other problems

Given a graph G, we can construct G' such that $\mathrm{rw}(G') \leqslant \mathrm{rw}(G) + 1$ and the following are equivalent:

- G has an **independent set** of size k
- G' has an **induced matching** of size k
- G' has an **induced forest** of size $2k$

Theorem [B., Korhonen and Nederlof, 2022+]
Under ETH, there are no $2^{o(\mathrm{rw}^2)} n^{O(1)}$ time algorithms for **Max. Induced Matching** and **FVS**.

Dominating Set



Theorem [B., Korhonen and Nederlof, 2022+]
Under ETH, there are no $2^{o(\mathrm{rw}^2)} n^{O(1)}$ time algorithms for **Weighted** Dominating Set.

Boolean-width [Bui-Xuan, Telle, and Vatshelle, 2011]
Defined from $\mathrm{boolw}(A) := \log_2 |\{N(X) \cap \overline{A} \mid X \subseteq A\}|$.

Equivalent to clique-width and rank-width!

Theorem [Bui-Xuan, Telle, and Vatshelle, 2011]
For all graph G, we have $\log_2 \mathrm{rw} \leq \mathrm{boolw} \leq O(\mathrm{rw}^2)$.

Theorem [B., Korhonen and Nederlof, 2022+]
There are graphs with rank-width k and boolean-width $\Omega(k^2)$ for arbitrary large k.

Conclusion

First non-trivial ETH lower bounds for **rank-width**

Using $|\{N(X) \cap \overline{A} \mid X \subseteq A\}|$ leads to optimal algorithms for many **width measures** (tree-width, clique-width and rank-width) for several **problems**!

Theorem [Belmonte and Sau, 2021]
Some problems based on **parity** can be solved in time $2^{O(\mathrm{rw})} n^{O(1)}$.

Open questions

What about:

- **Unweighted** Dominating Set? $(2^{O(\mathrm{rw}^2)} n^{O(1)})$
- q-**Coloring**? $(2^{O(q\mathrm{rw}^2)} n^{O(1)})$
- Chromatic Number? $(n^{2^{O(\mathrm{rw}^2)}})$

We need tight lower bounds for **mim-width** and Independent Set!

$n^{O(k)}$	$n^{o(k/\log k)}$
[Bui-Xuan et al., 2013]	[Bakkane and Jaffke, 2022+]

Thank you!