Well-Quasi-Orders

and

Logic on Graphs

ZYGMUNT
ZALESKI
STICHTING

Aliaume Lopez
University of Warsaw

Les Houches
FMT'25, 2025-05-29

https://www.irif.fr/~alopez/

Well-Quasi-Orders 101

Well-quasi-order (WQO) $(X, \leq)$

$$\begin{tikzcd} x_0 & x_1 & x_2 & \dots & x_i \arrow[rr, bend right, "\leq"'] \arrow[rr, phantom, "\exists"] & \dots & x_j & \dots \end{tikzcd}$$

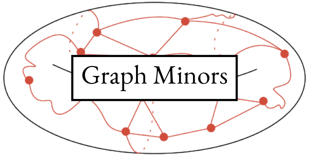
Well-Quasi-Orders 101

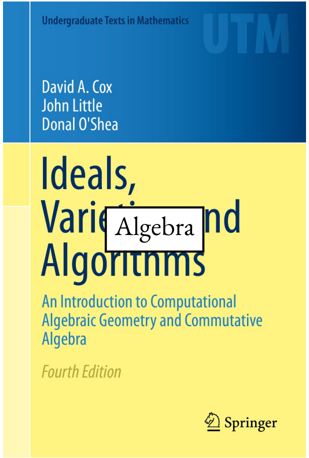
Well-quasi-order (WQO) $(X, \leq)$

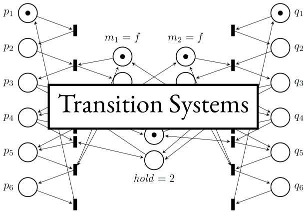
$$x_0 \qquad x_1 \qquad x_2 \qquad \ldots \qquad x_i \qquad \ldots \qquad x_j \qquad \ldots$$

$\exists$

$\leq$

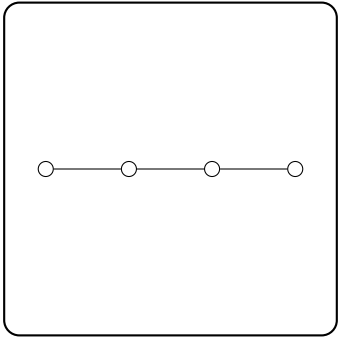






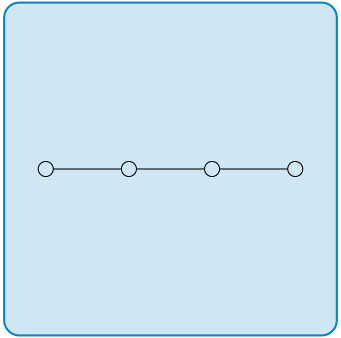
Is the following class WQO for induced subgraphs

YES NO



Is the following class WQO for induced subgraphs

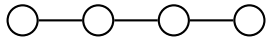
YES NO

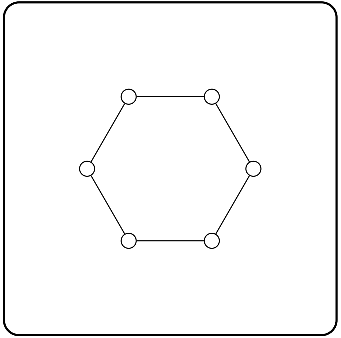


Is the following class WQO for induced subgraphs

YES

NO

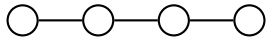


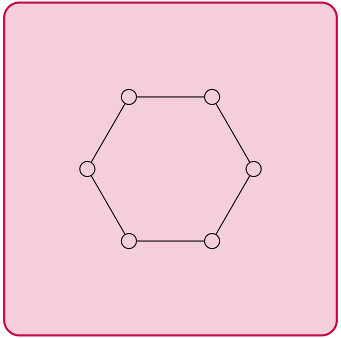


Is the following class WQO for induced subgraphs

YES

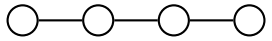
NO

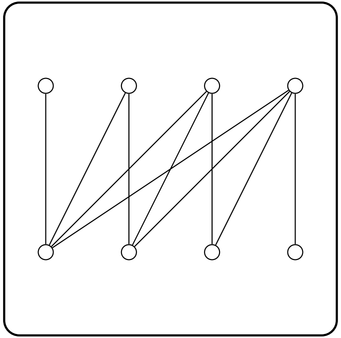




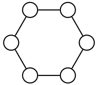
Is the following class WQO for induced subgraphs

YES





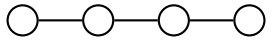
NO

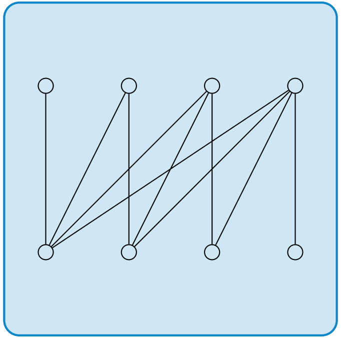


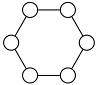
Is the following class WQO for induced subgraphs

YES

NO

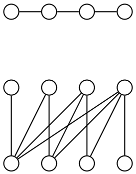


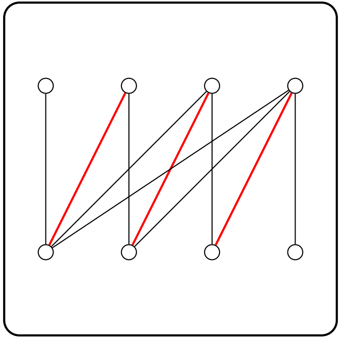




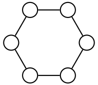
Is the following class WQO for induced subgraphs

YES



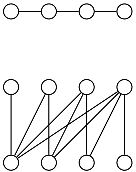


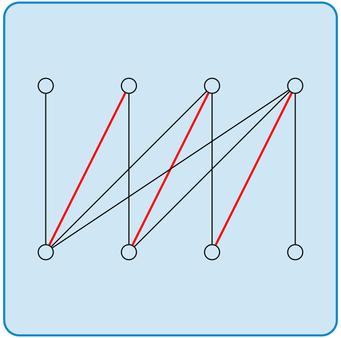
NO



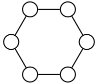
Is the following class WQO for induced subgraphs

YES



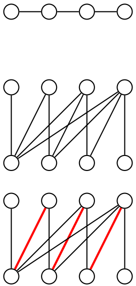


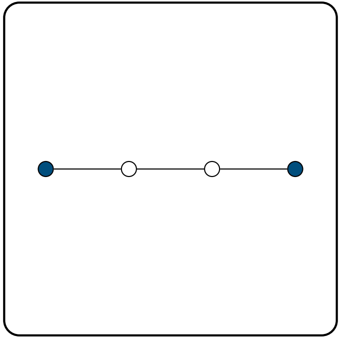
NO



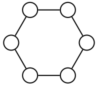
Is the following class WQO for induced subgraphs

YES



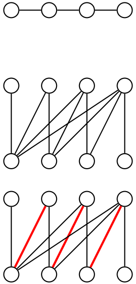


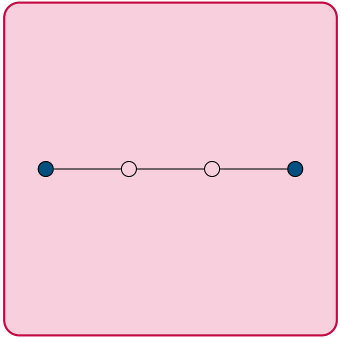
NO

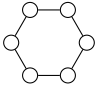


Is the following class WQO for induced subgraphs

YES **NO**

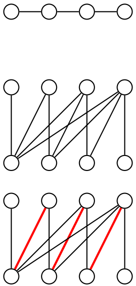


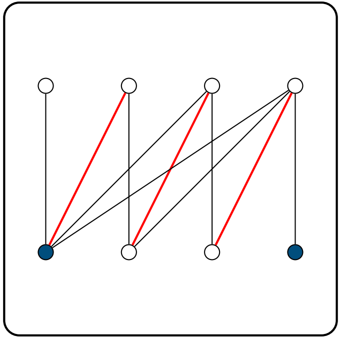




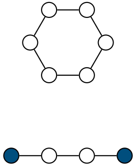
Is the following class WQO for induced subgraphs

YES



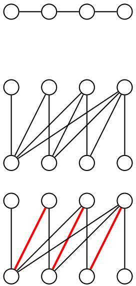


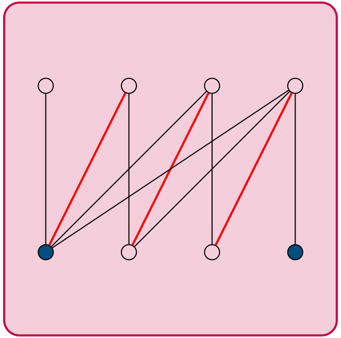
NO



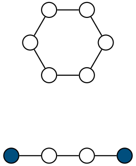
Is the following class WQO for induced subgraphs

YES



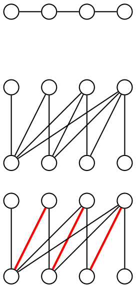


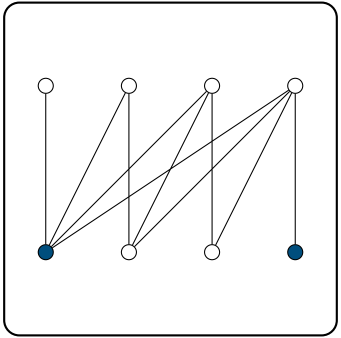
NO



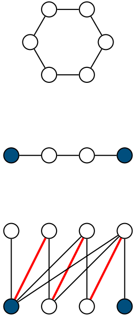
Is the following class WQO for induced subgraphs

YES





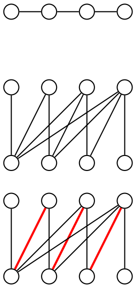
NO

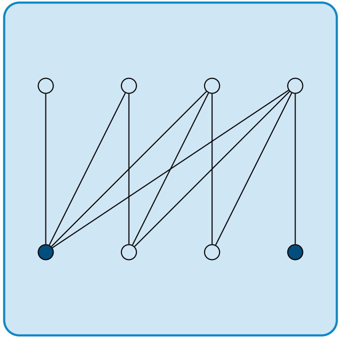


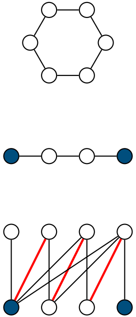
Is the following class WQO for induced subgraphs

YES

NO







What is known for WQO?

What is known for WQO ?

Theorems [Ding'92, folklore]

- Bounded tree-depth $\Longrightarrow$ WQO
- Bounded shrub-depth $\Longrightarrow$ WQO
- m-partite cograph $\Longrightarrow$ WQO

What is known for WQO ?

Theorems [Ding'92, folklore]

— Bounded tree-depth $\implies$ WQO
— Bounded shrub-depth $\implies$ WQO
— m-partite cograph $\implies$ WQO

Remark [folklore] "Every" order $(X, \leq)$ is represented as a class of finite graphs $(\mathcal{C}, \subseteq_i)$.
$\rightsquigarrow$ Not easy to characterise WQOs inside

What is known for WQO ?

Theorems [Ding'92, folklore]

- Bounded tree-depth $\implies$ WQO
- Bounded shrub-depth $\implies$ WQO
- m-partite cograph $\implies$ WQO

Remark [folklore] "Every" order $(X, \leq)$ is represented as a class of finite graphs $(\mathcal{C}, \subseteq_i)$.
$\rightsquigarrow$ Not easy to characterise WQOs inside

For **colored paths** (i.e. finite words), WQO is decidable

- for *regular languages* [Atminas, Lozin, Moshkov, '17]
- for languages recognized by *amalgamation systems* (CFG, VASS, etc) and infixes of *morphic words* [Lhote, L, Schütze, arxiv 2025]

What is known for WQO ?

Theorems [Ding'92, folklore]
- Bounded tree-depth $\implies$ WQO
- Bounded shrub-depth $\implies$ WQO
- m-partite cograph $\implies$ WQO

Remark [folklore] "Every" order $(X, \leq)$ is represented as a class of finite graphs $(\mathcal{C}, \subseteq_i)$.
$\rightsquigarrow$ Not easy to characterise WQOs inside

Instead : consider *hereditary classes* and *freely colour* them

For **colored paths** (i.e. finite w
- for *regular languages* [At
- for languages recognized by *amalgamation systems* (CFG, VASS, etc) and infixes of *morphic words* [Lhote, L, Schütze, arxiv 2025]

Labelling Graphs

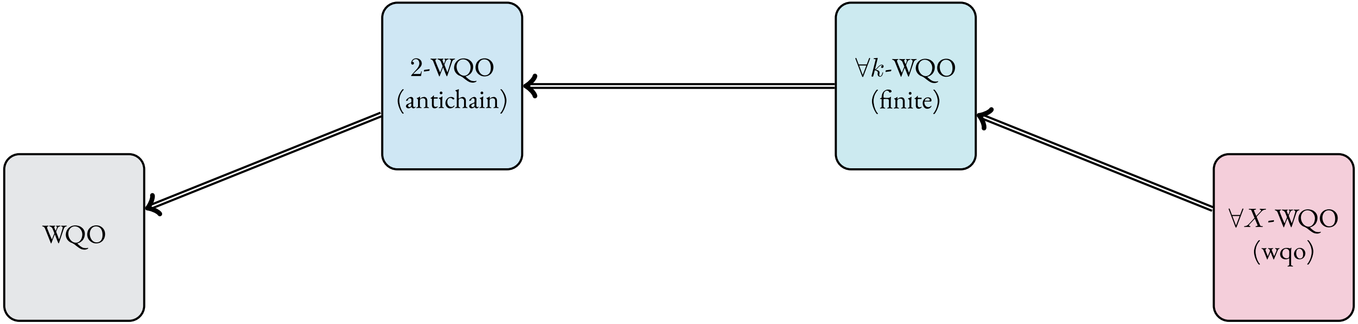
$\mathsf{Label}_X(\mathcal{C})$ "X-labelled elements of $\mathcal{C}$"

$\mathcal{C}$ is $(X, \leq)$-WQO $\iff$ $\mathsf{Label}_X(\mathcal{C})$ is WQO

Labelling Graphs

$\mathsf{Label}_X(\mathcal{C})$ “X-labelled elements of $\mathcal{C}$”

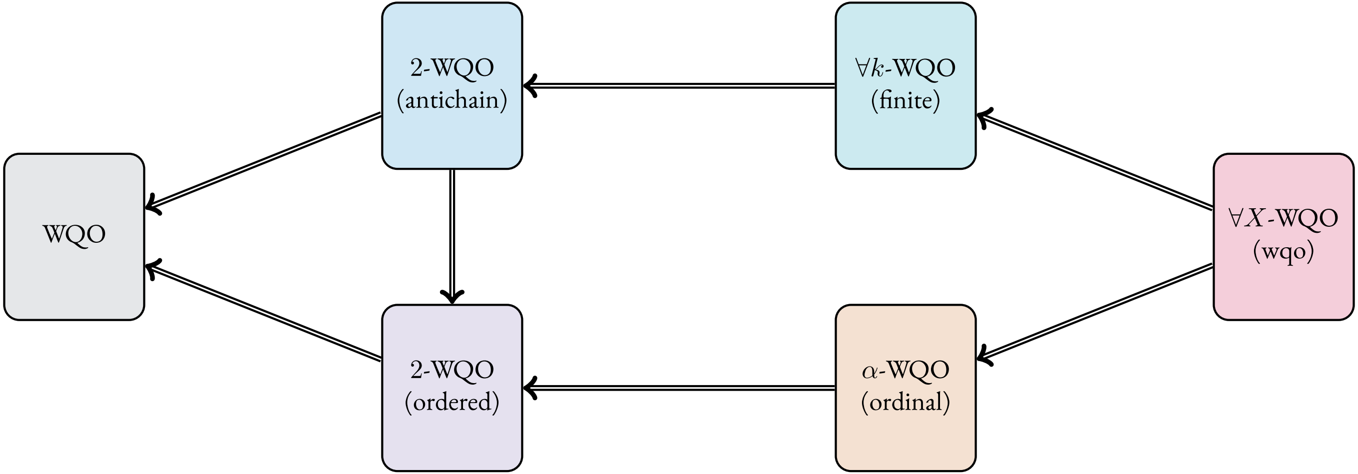
$\mathcal{C}$ is $(X, \leq)$-WQO $\iff$ $\mathsf{Label}_X(\mathcal{C})$ is WQO



Labelling Graphs

$\mathsf{Label}_X(\mathcal{C})$ "X-labelled elements of $\mathcal{C}$"

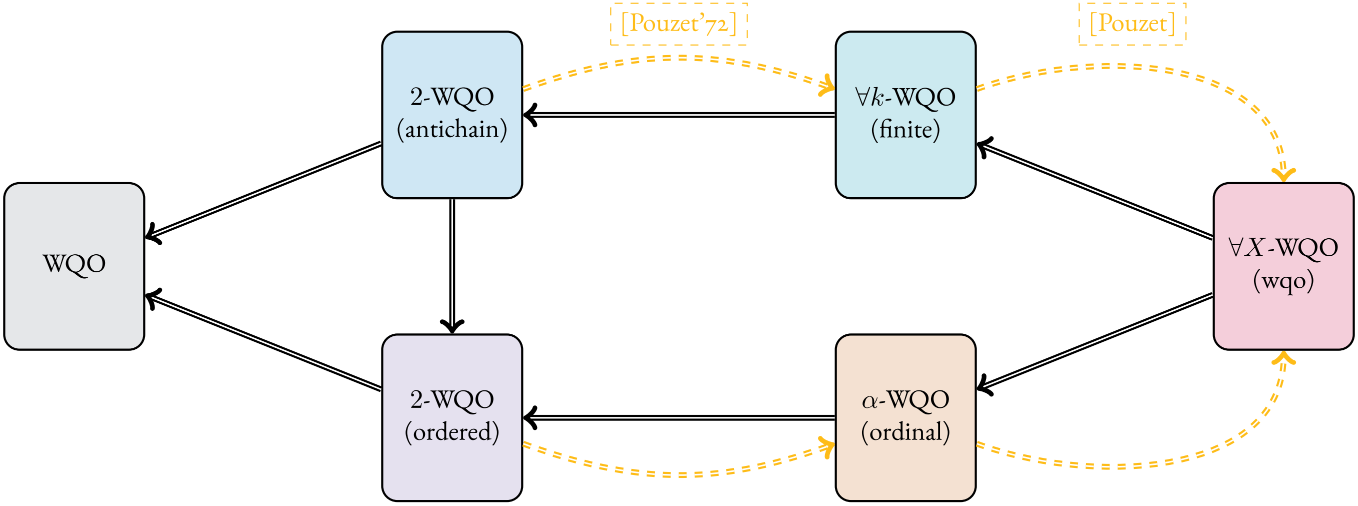
$\mathcal{C}$ is $(X, \leq)$-WQO $\iff$ $\mathsf{Label}_X(\mathcal{C})$ is WQO



Labelling Graphs

$\mathsf{Label}_X(\mathcal{C})$ "X-labelled elements of $\mathcal{C}$"

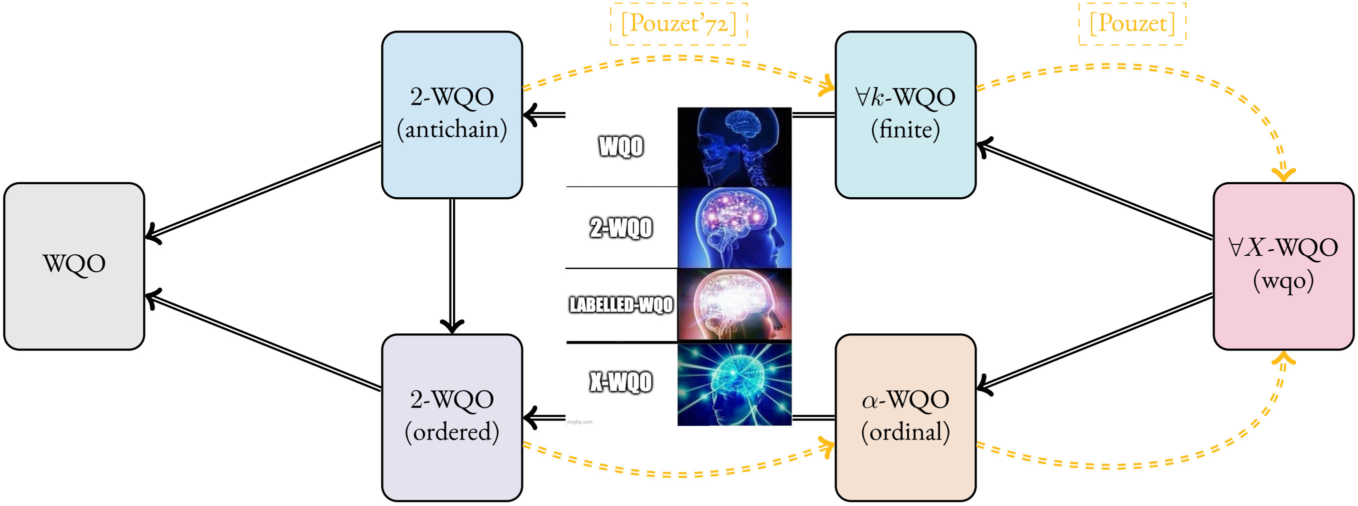
$\mathcal{C}$ is $(X, \leq)$-WQO $\iff$ $\mathsf{Label}_X(\mathcal{C})$ is WQO



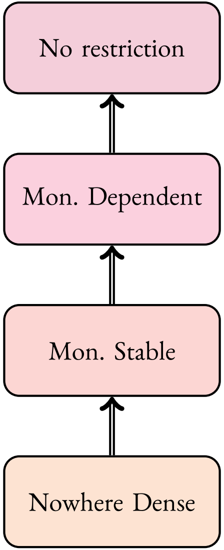
Labelling Graphs

$\mathsf{Label}_X(\mathcal{C})$ "X-labelled elements of $\mathcal{C}$"

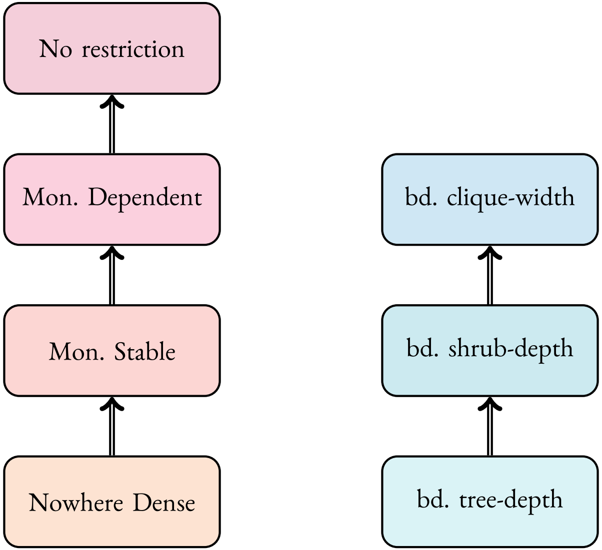
$\mathcal{C}$ is $(X, \leq)$-WQO $\iff$ $\mathsf{Label}_X(\mathcal{C})$ is WQO



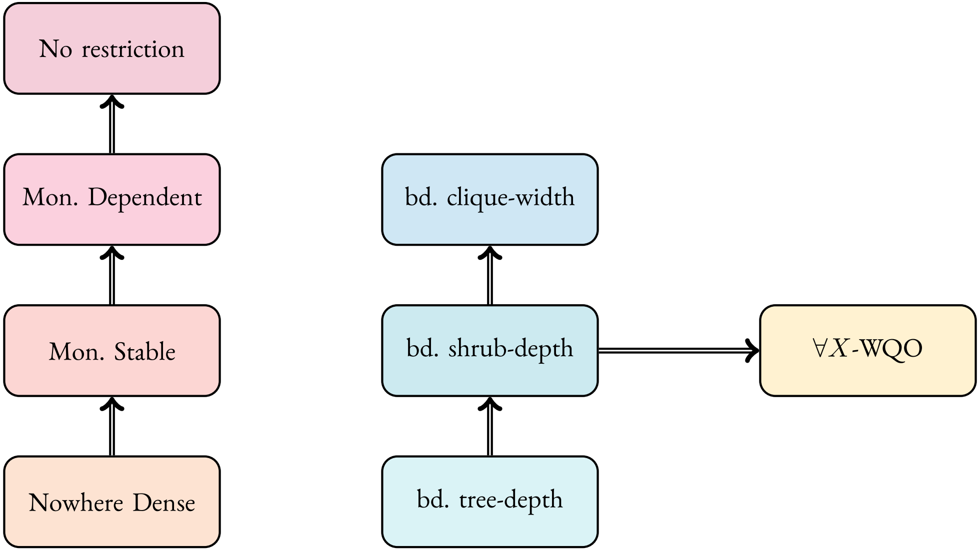
Relationship with structural graph theory



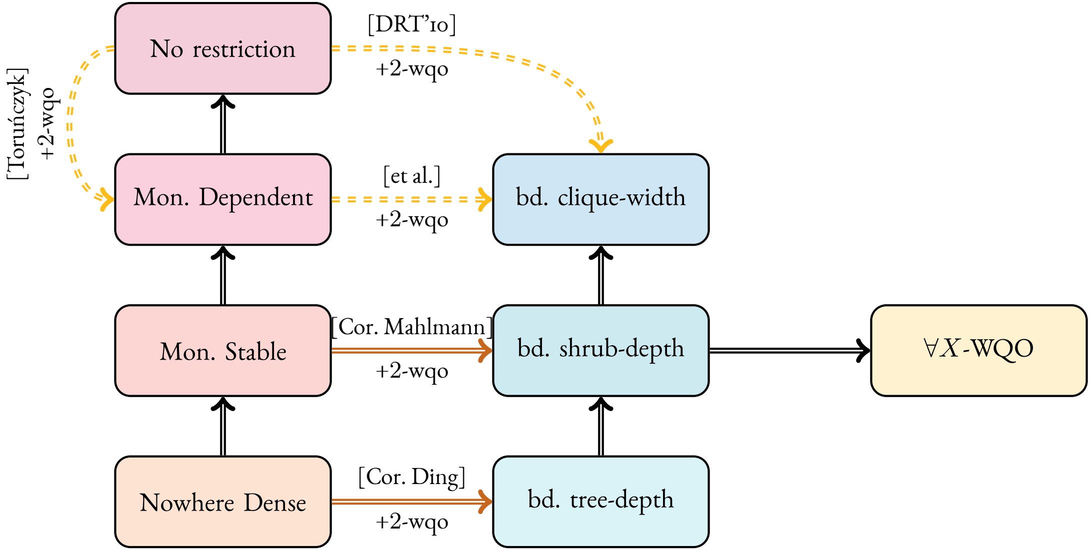
Relationship with structural graph theory



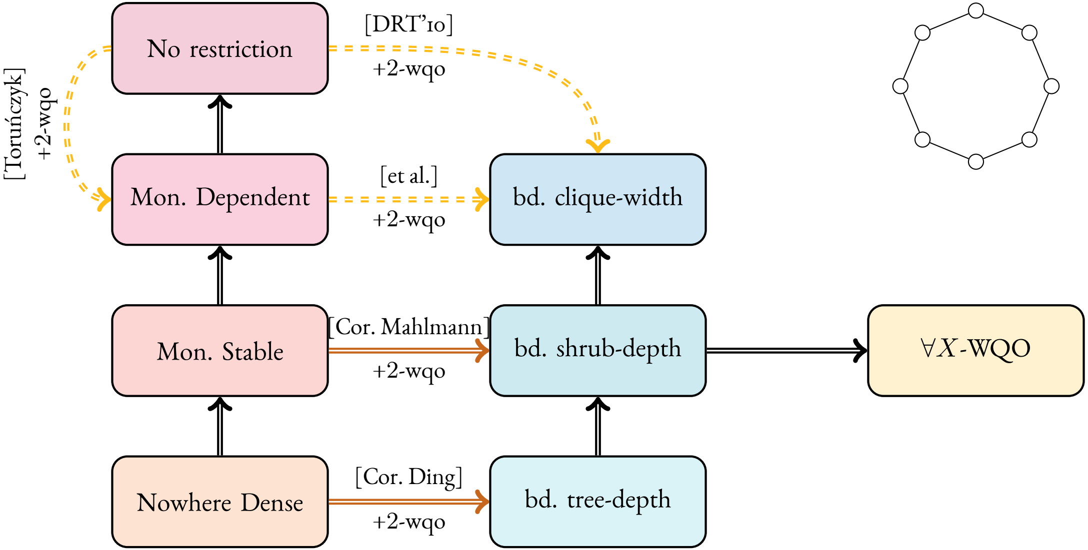
Relationship with structural graph theory



Relationship with structural graph theory



Relationship with structural graph theory



Relabel Functions / Clique-Width / NLC_ℓ

Finite Set : $Q = \{a, b, c\}$

Relabels : $\mathcal{F} = \{\mathrm{id}_Q, \rho, \rho^2\}$

$$\rho : \begin{cases} a & \mapsto b \\ b & \mapsto c \\ c & \mapsto c \end{cases}$$

Relabel Functions / Clique-Width / NLC_ℓ

Finite Set : $Q = \{a, b, c\}$

Relabels : $\mathcal{F} = \{\mathrm{id}_Q, \rho, \rho^2\}$

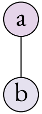
$$\rho \colon \begin{cases} a & \mapsto b \\ b & \mapsto c \\ c & \mapsto c \end{cases}$$

Relabel Functions / Clique-Width / NLC_ℓ

Finite Set : $Q = \{a, b, c\}$

Relabels : $\mathcal{F} = \{\mathrm{id}_Q, \rho, \rho^2\}$

$$\rho : \begin{cases} a & \mapsto b \\ b & \mapsto c \\ c & \mapsto c \end{cases}$$



Connect : ('a', 'b')

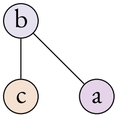
Relabel using ρ

Relabel Functions / Clique-Width / NLC_ℓ

Finite Set : $Q = \{a, b, c\}$

Relabels : $\mathcal{F} = \{\mathrm{id}_Q, \rho, \rho^2\}$

$$\rho : \begin{cases} a & \mapsto b \\ b & \mapsto c \\ c & \mapsto c \end{cases}$$



Connect : ('a', 'b'), ('a', 'c')

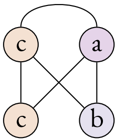
Relabel using ρ

Relabel Functions / Clique-Width / NLC_ℓ

Finite Set : $Q = \{a, b, c\}$

Relabels : $\mathcal{F} = \{\mathrm{id}_Q, \rho, \rho^2\}$

$$\rho : \begin{cases} a & \mapsto b \\ b & \mapsto c \\ c & \mapsto c \end{cases}$$



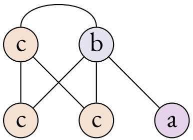
Connect : ('a', 'b')
Relabel using ρ

Relabel Functions / Clique-Width / NLC_ℓ

Finite Set : $Q = \{a, b, c\}$

Relabels : $\mathcal{F} = \{\mathrm{id}_Q, \rho, \rho^2\}$

$$\rho \colon \begin{cases} a & \mapsto b \\ b & \mapsto c \\ c & \mapsto c \end{cases}$$



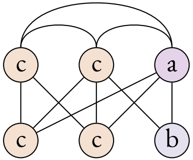
Connect : ('a', 'b'), ('a', 'c')
Relabel using ρ

Relabel Functions / Clique-Width / NLC_ℓ

Finite Set : $Q = \{a, b, c\}$

Relabels : $\mathcal{F} = \{\mathrm{id}_Q, \rho, \rho^2\}$

$$\rho : \begin{cases} a & \mapsto b \\ b & \mapsto c \\ c & \mapsto c \end{cases}$$



Connect : ('a', 'b')

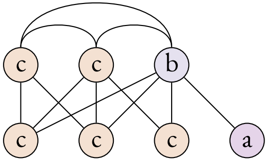
Relabel using ρ

Relabel Functions / Clique-Width / NLC_ℓ

Finite Set : $Q = \{a, b, c\}$

Relabels : $\mathcal{F} = \{\mathrm{id}_Q, \rho, \rho^2\}$

$$\rho\colon \begin{cases} a & \mapsto b \\ b & \mapsto c \\ c & \mapsto c \end{cases}$$



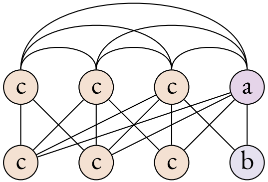
Connect : ('a', 'b'), ('a', 'c')
Relabel using ρ

Relabel Functions / Clique-Width / NLC_ℓ

Finite Set : $Q = \{a, b, c\}$

Relabels : $\mathcal{F} = \{\mathrm{id}_Q, \rho, \rho^2\}$

$$\rho : \begin{cases} a & \mapsto b \\ b & \mapsto c \\ c & \mapsto c \end{cases}$$



Connect : ('a', 'b')

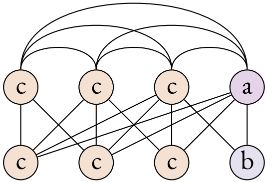
Relabel using ρ

Relabel Functions / Clique-Width / NLC_ℓ

Finite Set : $Q = \{a, b, c\}$

Relabels : $\mathcal{F} = \{\mathrm{id}_Q, \rho, \rho^2\}$

$$\rho : \begin{cases} a & \mapsto b \\ b & \mapsto c \\ c & \mapsto c \end{cases}$$



Connect : ('a', 'b')

…elabel using ρ

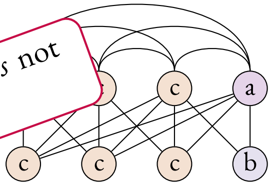
Theorem [DRT'10] One can decide whether $\mathsf{Relab}(\mathcal{F})$ is 2-wqo.
2-wqo $\iff$ $\forall X$-WQO holds on these classes.

Relabel Functions / Clique-Width / NLC_ℓ

Finite Set : $Q = \{a, b, c\}$

Relabels : $\mathcal{F} = \{\mathrm{id}_Q, \rho, \rho^2\}$

$$\rho : \begin{cases} a & \mapsto b \\ b & \mapsto c \\ c & \mapsto c \end{cases}$$



Connect : ('a', 'b')

Relabel using ρ

Problem : this is a theorem on *semigroups* not *languages.*

[DRT'10] One can decide whether $\mathrm{Relab}(\mathcal{F})$ is 2-wqo.
2-wqo $\iff$ $\forall X$-WQO holds on these classes.

Clique-width (second attempt)

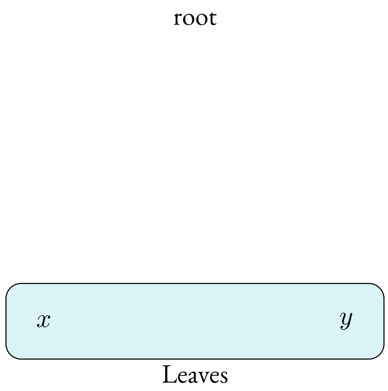
Finite **monoid** : M

Accepting condition : $P \subseteq M^3$

Clique-width (second attempt)

Finite **monoid** : M

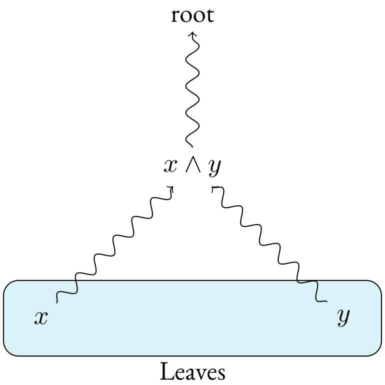
Accepting condition : $P \subseteq M^3$



Clique-width (second attempt)

Finite **monoid** : M

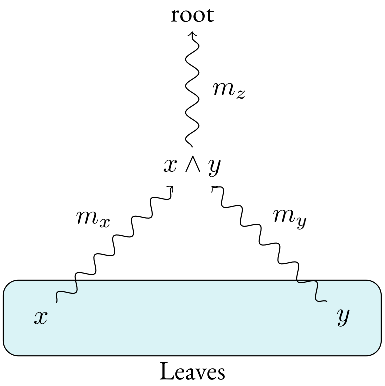
Accepting condition : $P \subseteq M^3$



Clique-width (second attempt)

Finite **monoid** : M

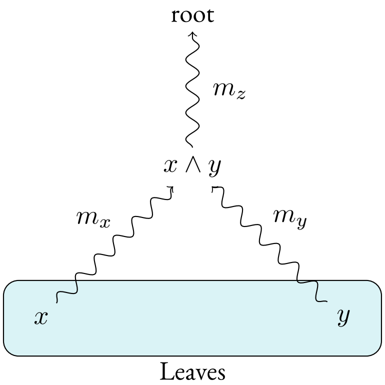
Accepting condition : $P \subseteq M^3$



Clique-width (second attempt)

Finite **monoid** : M

Accepting condition : $P \subseteq M^3$



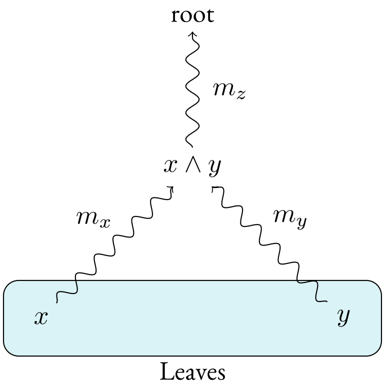
$$\boxed{E(x,y) \iff (m_x, m_z, m_y) \in P}$$

Clique-width (second attempt)

Finite **monoid** : M

Accepting condition : $P \subseteq M^3$

Theorem (new) Given M, P, one can decide if $\mathsf{Relabel}(M, P)$ is $\forall k$-WQO.
For these classes
$f(|M|)$-WQO $\iff$ $\forall k$-WQO $\iff$ $\forall X$-WQO.



$$E(x, y) \iff (m_x, m_z, m_y) \in P$$

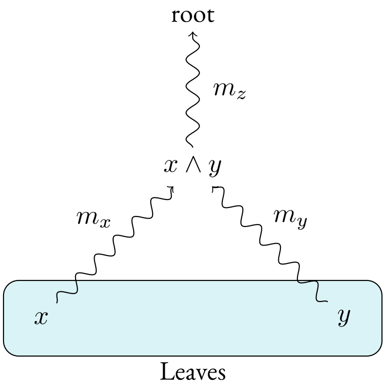
Clique-width (second attempt)

Finite **monoid** : M

Accepting condition : $P \subseteq M^3$

Theorem (new) Given M, P, one can decide if $\mathsf{Relabel}(M, P)$ is $\forall k$-WQO.
For these classes
$f(|M|)$-WQO $\iff$ $\forall k$-WQO $\iff$ $\forall X$-WQO.

Corollary For classes bounded clique-width
$\forall k$-WQO $\iff$ $\forall X$-WQO.



$$E(x, y) \iff (m_x, m_z, m_y) \in P$$

Clique-width (second attempt)

Finite **monoid** : M

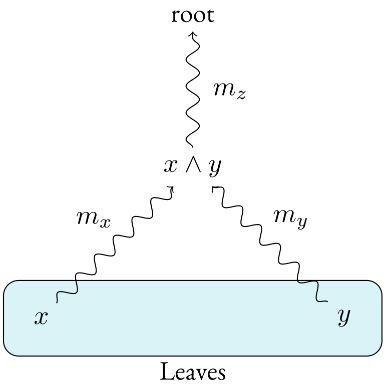
Accepting condition : $P \subseteq M^3$

Theorem (new) Given M, P, one can decide if $\mathsf{Relabel}(M, P)$ is $\forall k$-WQO.
For these classes
$f(|M|)$-WQO $\iff$ $\forall k$-WQO $\iff$ $\forall X$-WQO.

Corollary For classes bounded clique-width
$\forall k$-WQO $\iff$ $\forall X$-WQO.

Bonus

- reduces to classes of bounded *linear* clique width
- existential transductions of finite paths



$$E(x, y) \iff (m_x, m_z, m_y) \in P$$

How does one prove such statements?

How does one prove such statements?

Part 1 : Order tree-decompositions s.t. $I : \mathsf{Trees} \to \mathsf{Graphs}$ is *order preserving*

"product preserving" tree embeddings

Part 2 : Order tree-decompositions using a WQO.

usual tree embeddings (Kruskal)

Goal do (1) and (2) simultaneously.

How does one prove such statements?

Part 1 : Order tree-decompositions s.t. $I : \mathsf{Trees} \to \mathsf{Graphs}$ is *order preserving*

"product preserving" tree embeddings

Forward Ramseyan Splits [Colcombet'07]

Part 2 : Order tree-decompositions using a WQO.

usual tree embeddings (Kruskal)

Gap Embedding Relation [Dershowitz and Tzameret'03]

Goal do (1) and (2) simultaneously.

Both label nodes with elements from $\{1, \dots, n\}$

How does one prove such statements?

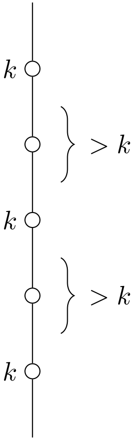
Part 1 : Order tree-decompositions s.t. I : Trees $\to$ Graphs is *order preserving*

"product preserving" tree embeddings

Part 2 : Order tree-decompositions using a WQO.

usual tree embeddings (Kruskal)

Goal do (1) and (2) simultaneously.



Forward Ramseyan Splits [Colcombet'07]

Gap Embedding Relation [Dershowitz and Tzameret'03]

Both label nodes with elements from $\{1, \ldots, n\}$

How does one prove such statements?

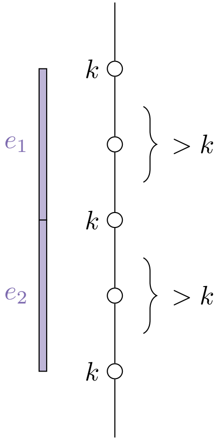
Part 1 : Order tree-decompositions s.t. I : Trees $\to$ Graphs is *order preserving*

"product preserving" tree embeddings

Part 2 : Order tree-decompositions using a WQO.

usual tree embeddings (Kruskal)

Goal do (1) and (2) simultaneously.



$e_1 e_2 = e_1$

Forward Ramseyan Splits [Colcombet'07]

Gap Embedding Relation [Dershowitz and Tzameret'03]

Both label nodes with elements from $\{1, \ldots, n\}$

How does one prove such statements ?

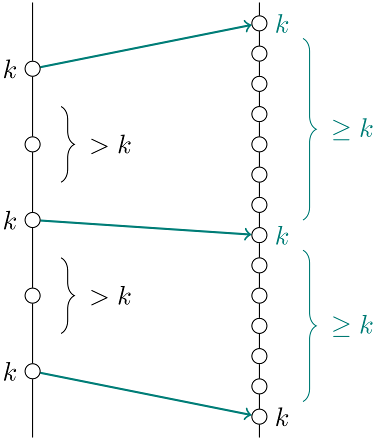
Part 1 : Order tree-decompositions s.t. I : Trees → Graphs is *order preserving*

"product preserving" tree embeddings

Part 2 : Order tree-decompositions using a WQO.

usual tree embeddings (Kruskal)

Goal do (1) and (2) simultaneously.



Forward Ramseyan Splits [Colcombet'07]

Gap Embedding Relation [Dershowitz and Tzameret'03]

Both label nodes with elements from $\{1, \ldots, n\}$

How does one prove such statements?

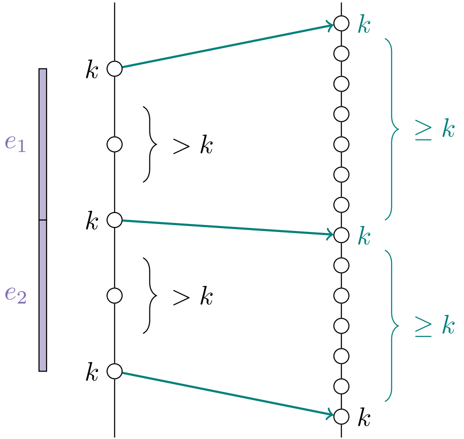
Part 1 : Order tree-decompositions s.t. I : Trees $\to$ Graphs is *order preserving*

"product preserving" tree embeddings

Part 2 : Order tree-decompositions using a WQO.

usual tree embeddings (Kruskal)

Goal do (1) and (2) simultaneously.



$e_1 e_2 = e_1$

Forward Ramseyan Splits [Colcombet'07]

Gap Embedding Relation [Dershowitz and Tzameret'03]

Both label nodes with elements from $\{1, \ldots, n\}$

How does one prove such statements ?

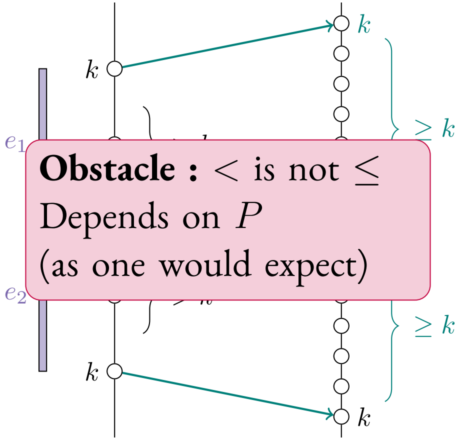
Part 1 : Order tree-decompositions s.t. I : Trees $\to$ Graphs is *order preserving*

"product preserving" tree embeddings

Part 2 : Order tree-decompositions using a WQO.

usual tree embeddings (Kruskal)

Goal do (1) and (2) simultaneously.



Obstacle : $<$ is not $\leq$
Depends on P
(as one would expect)

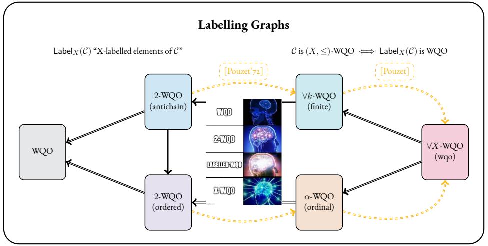
$e_1 e_2 = e_1$

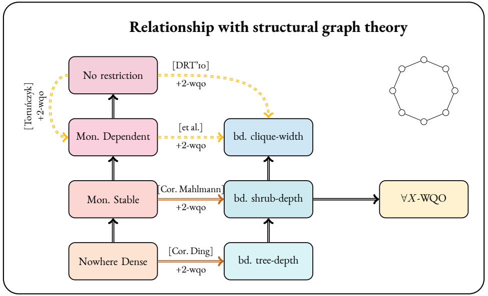
Forward Ramseyan Splits [Colcombet'07]

Gap Embedding Relation [Dershowitz and Tzameret'03]

Both label nodes with elements from $\{1, \ldots, n\}$

What's next ?





Theorem Given M, P, one can decide if $\text{Relabel}(M, P)$ is $\forall k$-WQO.
For these classes
$f(|M|)$-WQO $\iff$ $\forall k$-WQO $\iff$ $\forall X$-WQO.

2-**WQO** :
- 2-WQO (antichain) $\implies$ $\forall k$-WQO
- 2-WQO (order) $\implies$ $\forall k$-WQO
- Relationship to monadically dependent classes
- "Successor-free graphs"

WQO :
- Decide WQO for relabel functions
- Decide WQO given excluded patterns