First Order Logic on Pathwidth Revisited *Again*

Michael Lampis
LAMSADE

July 12th 2023 – ICALP Track B

One-Slide Summary

Theorem: Fix a positive integer p. Then, there is an algorithm that takes as input a graph G, a **path decomposition** of G of width p, and a **FO formula** ϕ, and decides if $G \models \phi$ in time $f(\phi)|G|^{O(1)}$, where f is an **elementary** function of ϕ.

One-Slide Summary

Theorem: Fix a positive integer p. Then, there is an algorithm that takes as input a graph G, a **path decomposition** of G of width p, and a **FO formula** ϕ, and decides if $G \models \phi$ in time $f(\phi)|G|^{O(1)}$, where f is an **elementary** function of ϕ.

Fun Fact: I couldn't sleep at night when I thought of this question!

One-Slide Summary

Theorem: Fix a positive integer p. Then, there is an algorithm that takes as input a graph G, a **path decomposition** of G of width p, and a **FO formula** ϕ, and decides if $G \models \phi$ in time $f(\phi)|G|^{O(1)}$, where f is an **elementary** function of ϕ.

Normal people: This question makes me sleepy!

One-Slide Summary

Theorem: Fix a positive integer p. Then, there is an algorithm that takes as input a graph G, a **path decomposition** of G of width p, and a **FO formula** ϕ, and decides if $G \models \phi$ in time $f(\phi)|G|^{O(1)}$, where f is an **elementary** function of ϕ.

Necessary Background:

- Treewidth, Pathwidth, Parameterized Complexity
- Meta-Theorems, Courcelle's Theorem, **Non-elementary** dependence
- Meta-Theorems with **elementary** dependence

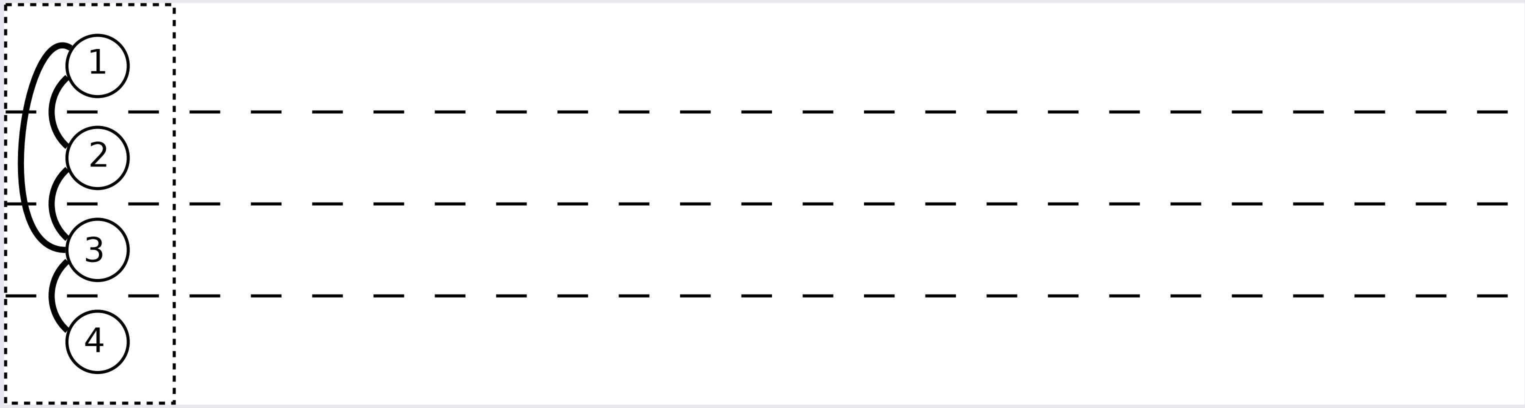
Background I:
Graph Widths
and
Parameterized Complexity

Gentle definition of pathwidth k:

- We have k stacks. Initially each contains a vertex. They are arbitrarily connected.
- At each step we add a vertex to the top of a stack. It can be connected to vertices currently on top of a stack.

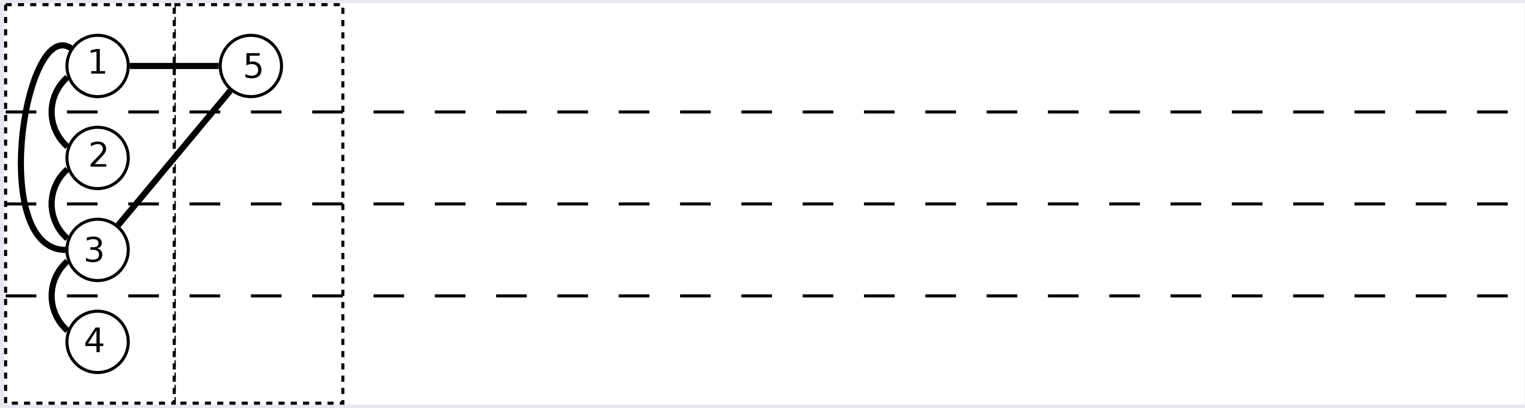
Gentle definition of pathwidth k:

- We have k stacks. Initially each contains a vertex. They are arbitrarily connected.
- At each step we add a vertex to the top of a stack. It can be connected to vertices currently on top of a stack.



Gentle definition of pathwidth k:

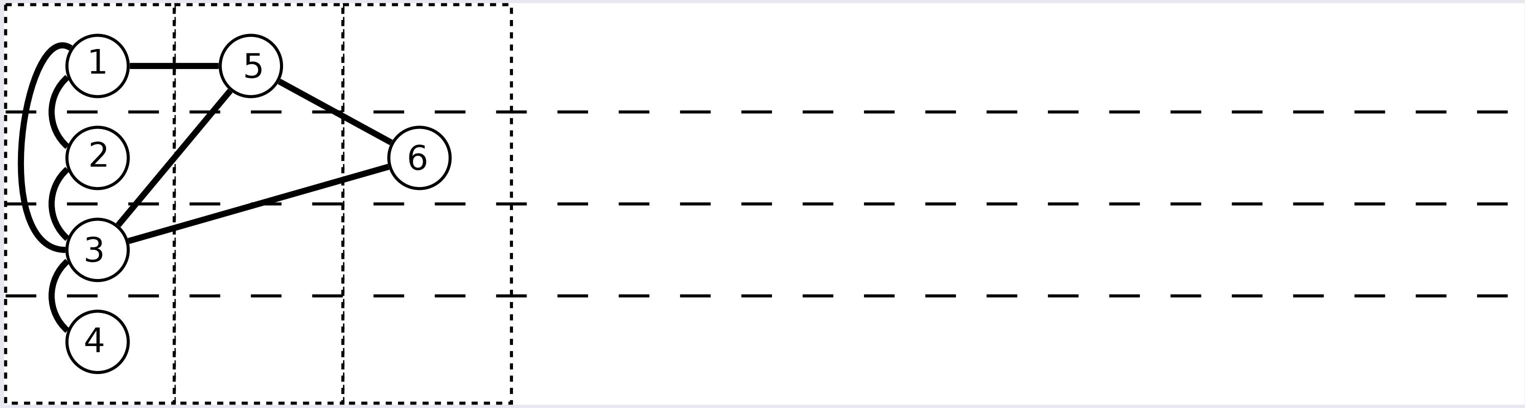
- We have k stacks. Initially each contains a vertex. They are arbitrarily connected.
- At each step we add a vertex to the top of a stack. It can be connected to vertices currently on top of a stack.



Treewidth – Pathwidth

Gentle definition of pathwidth k:

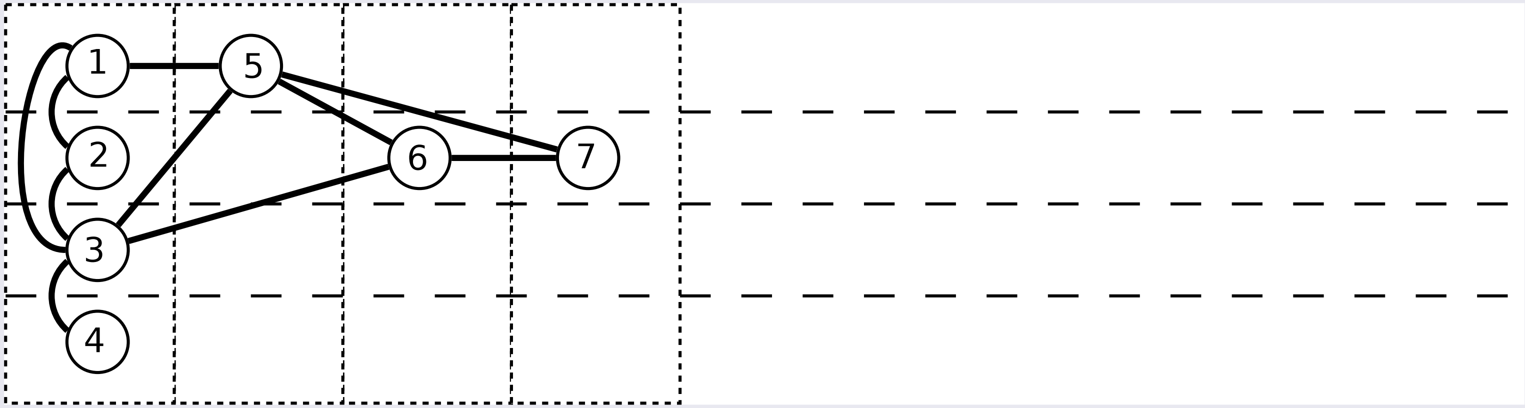
- We have k stacks. Initially each contains a vertex. They are arbitrarily connected.
- At each step we add a vertex to the top of a stack. It can be connected to vertices currently on top of a stack.



Treewidth – Pathwidth

Gentle definition of pathwidth k:

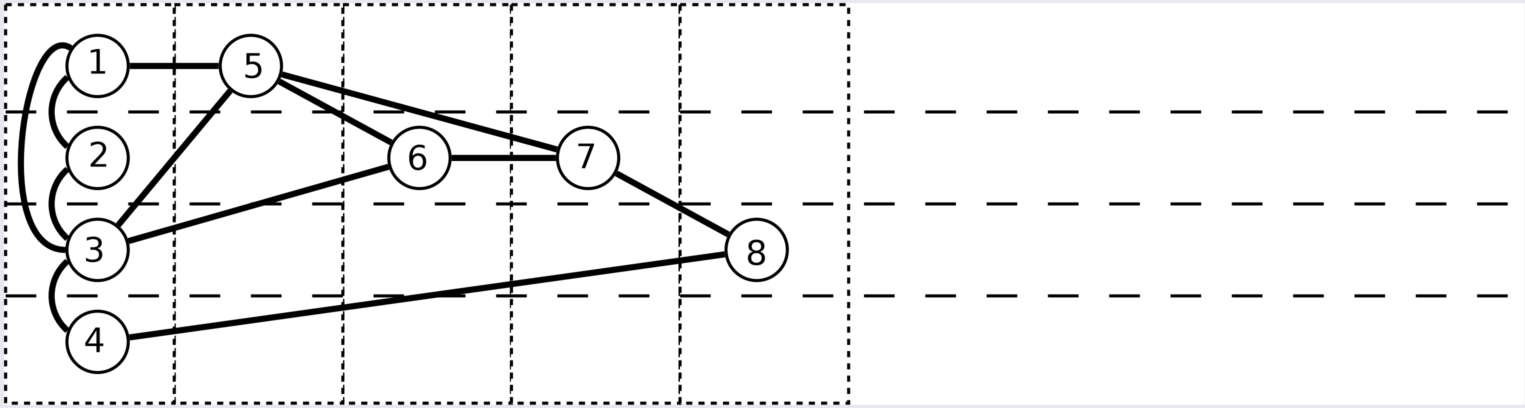
- We have k stacks. Initially each contains a vertex. They are arbitrarily connected.
- At each step we add a vertex to the top of a stack. It can be connected to vertices currently on top of a stack.



Treewidth – Pathwidth

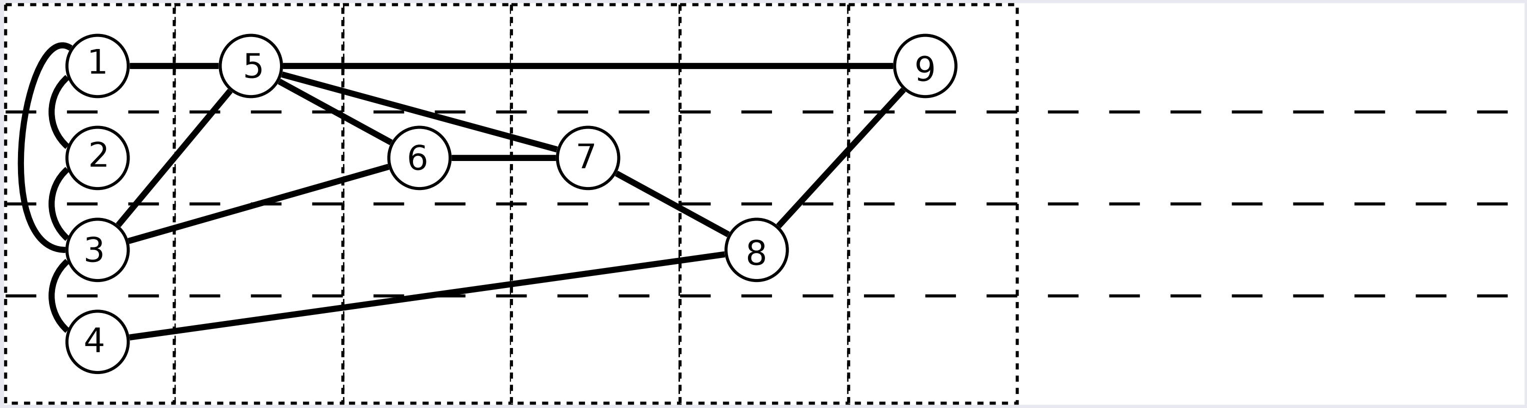
Gentle definition of pathwidth k:

- We have k stacks. Initially each contains a vertex. They are arbitrarily connected.
- At each step we add a vertex to the top of a stack. It can be connected to vertices currently on top of a stack.



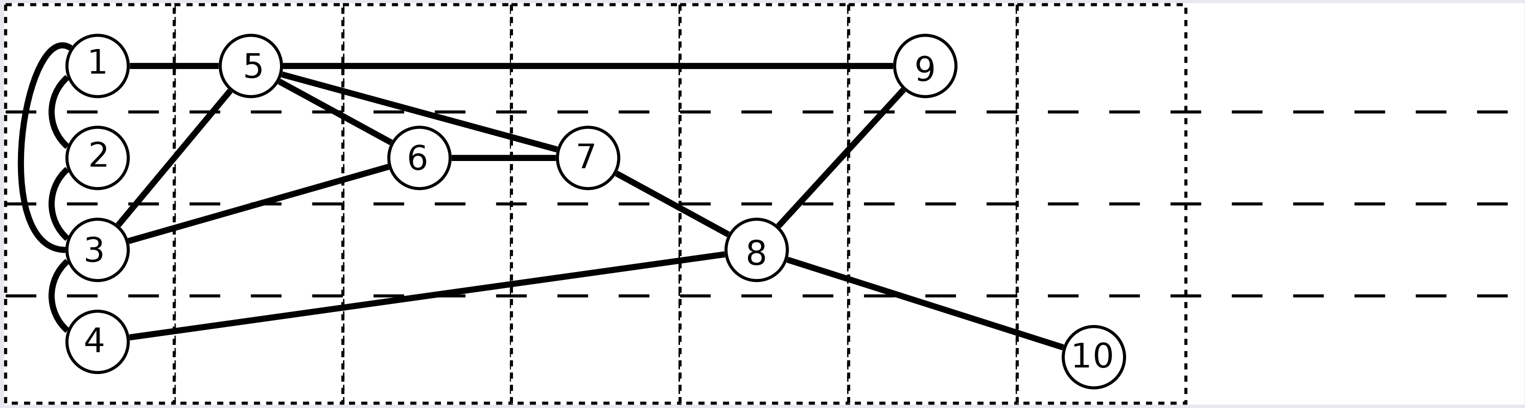
Gentle definition of pathwidth k:

- We have k stacks. Initially each contains a vertex. They are arbitrarily connected.
- At each step we add a vertex to the top of a stack. It can be connected to vertices currently on top of a stack.



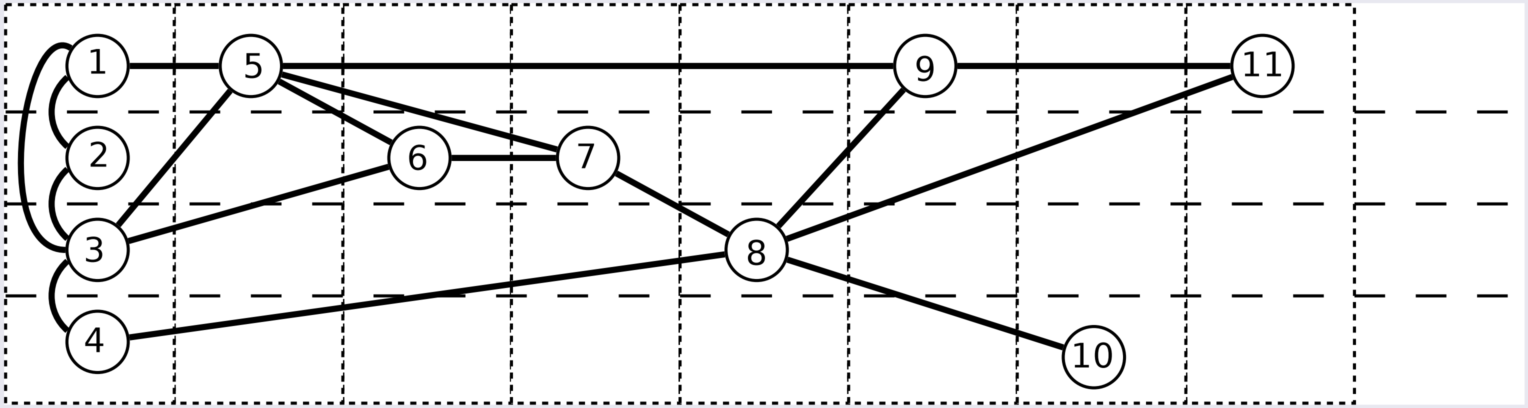
Gentle definition of pathwidth k:

- We have k stacks. Initially each contains a vertex. They are arbitrarily connected.
- At each step we add a vertex to the top of a stack. It can be connected to vertices currently on top of a stack.



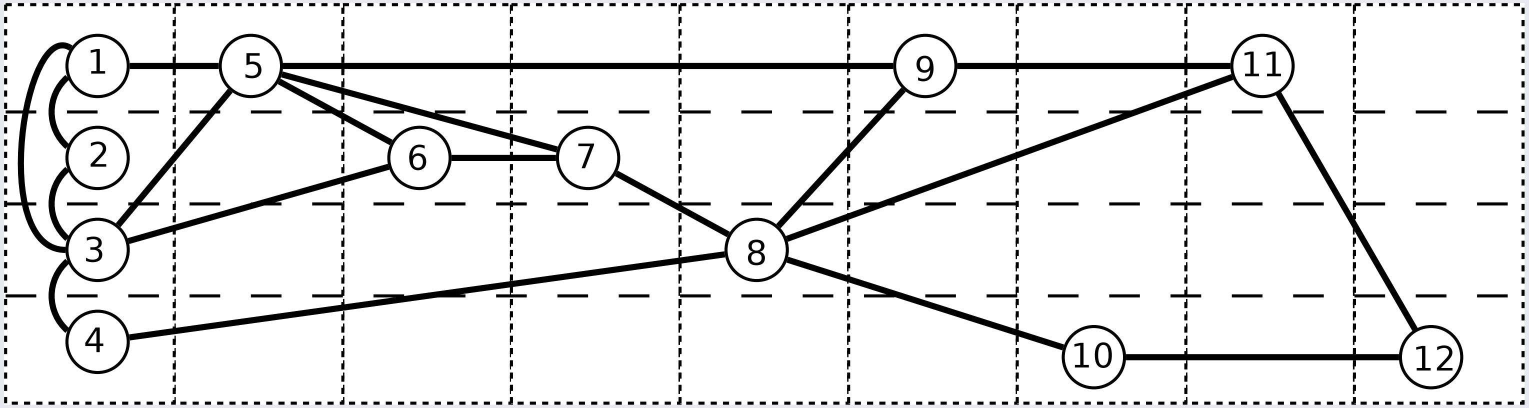
Gentle definition of pathwidth k:

- We have k stacks. Initially each contains a vertex. They are arbitrarily connected.
- At each step we add a vertex to the top of a stack. It can be connected to vertices currently on top of a stack.

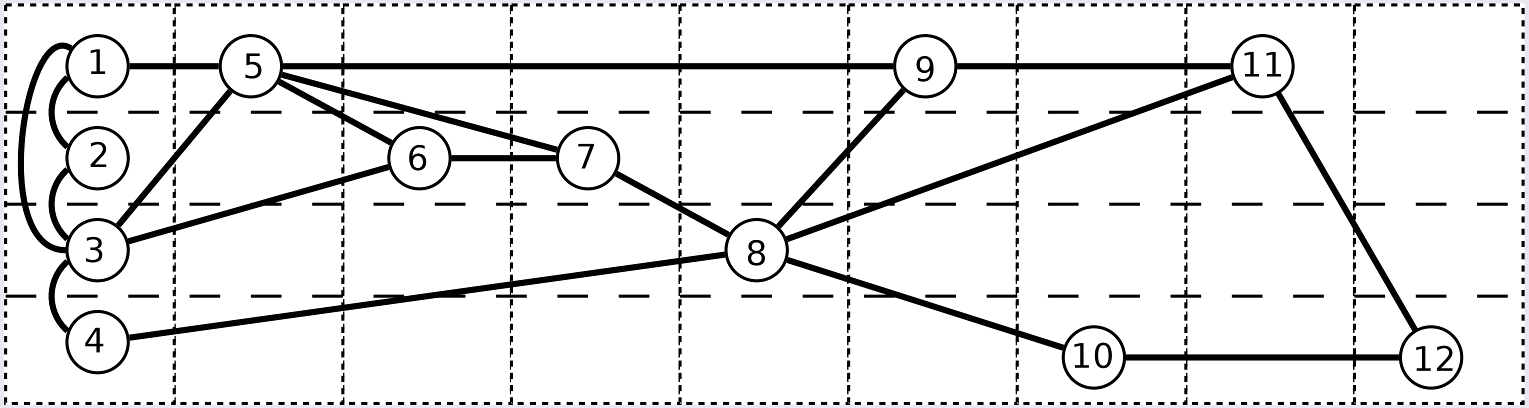


Gentle definition of pathwidth k:

- We have k stacks. Initially each contains a vertex. They are arbitrarily connected.
- At each step we add a vertex to the top of a stack. It can be connected to vertices currently on top of a stack.

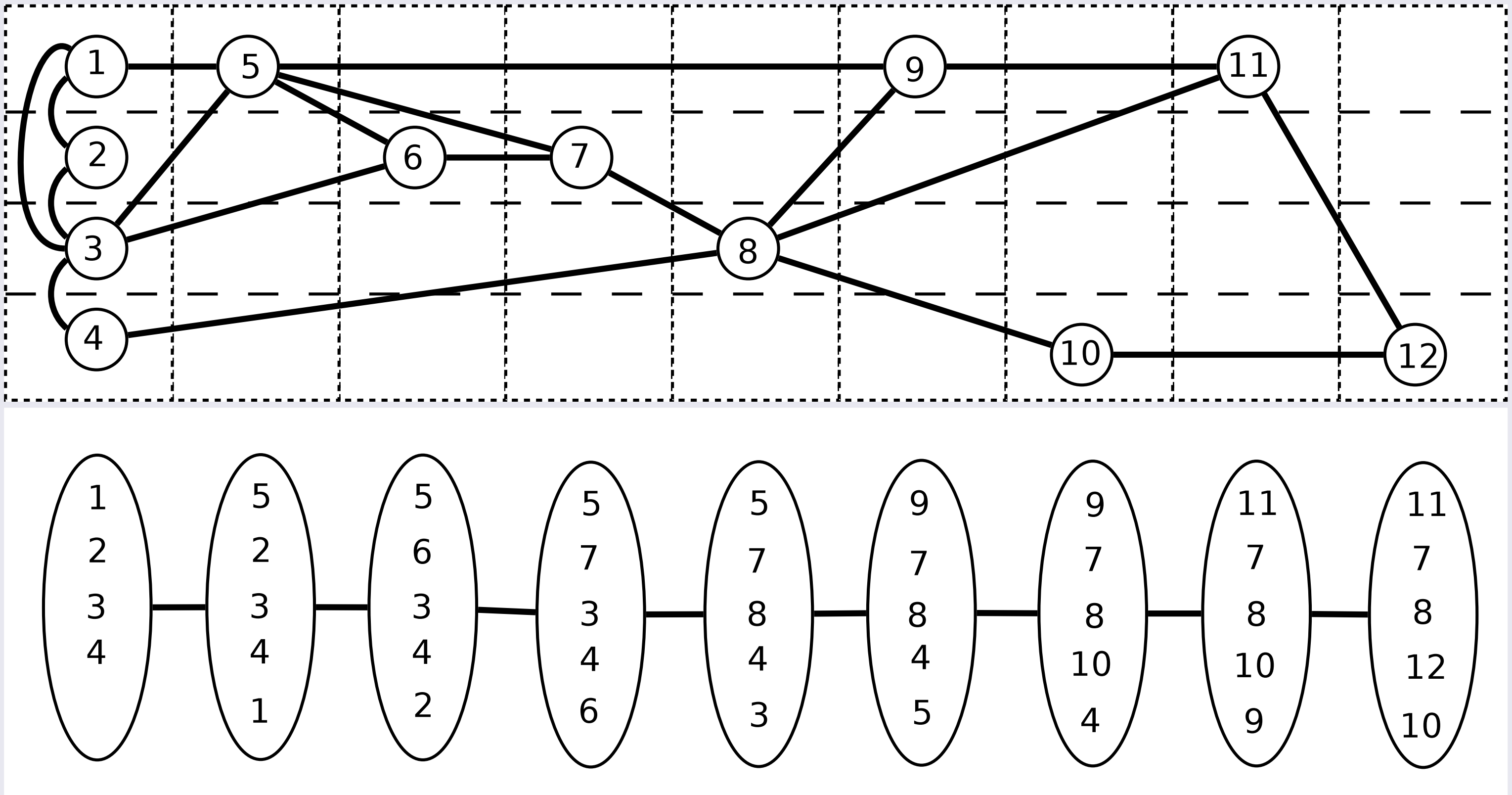


Note that this is equivalent to the standard definition of path decompositions.



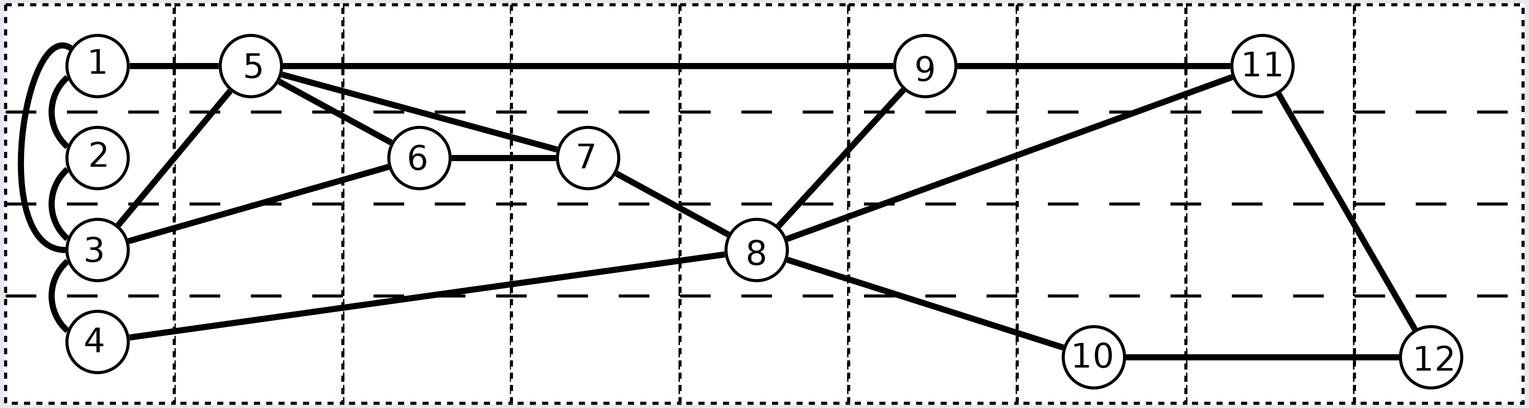
Treewidth – Pathwidth

Note that this is equivalent to the standard definition of path decompositions.

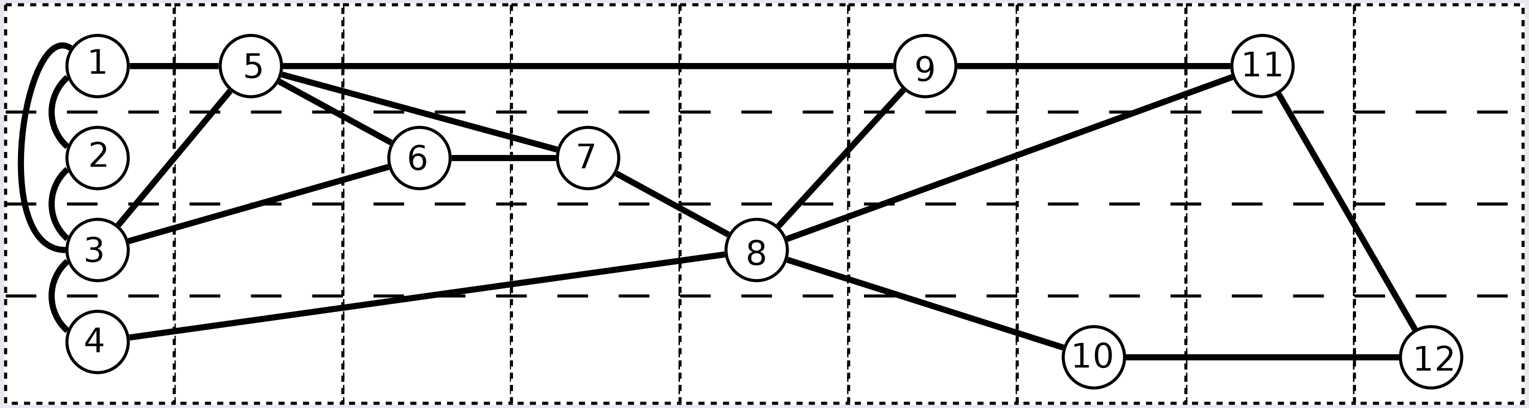


Treewidth – Pathwidth – Tree-depth

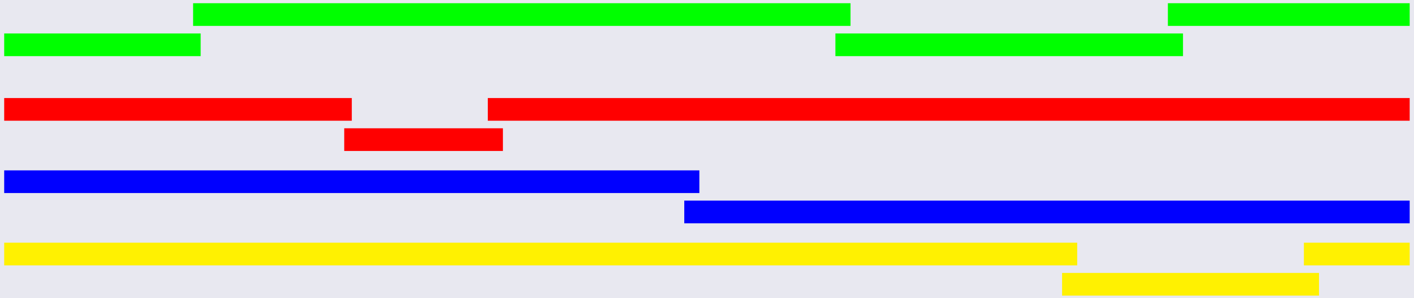
A connection to graph classes:



A connection to graph classes:

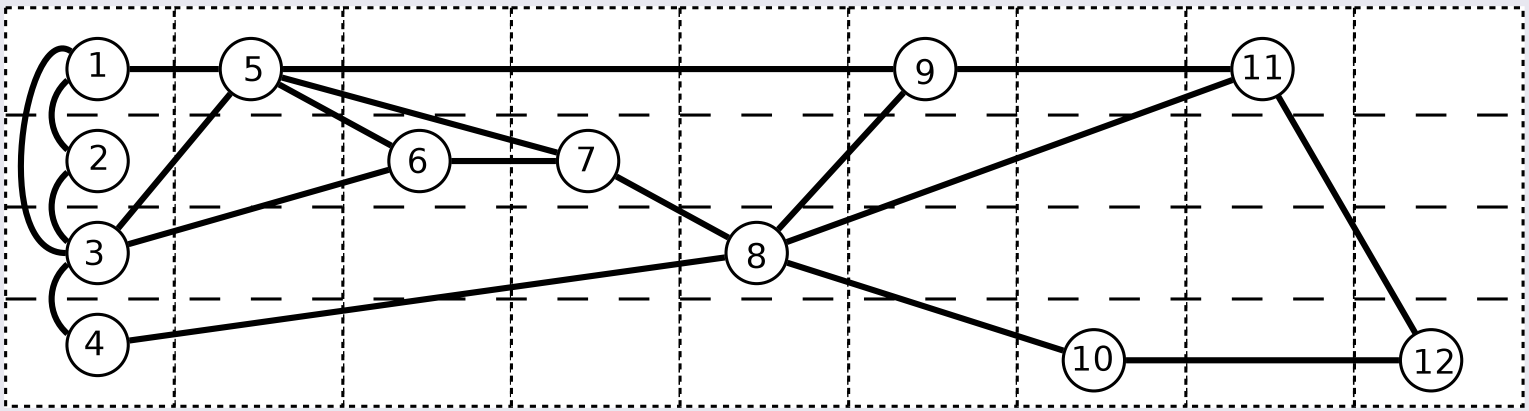


Corresponding interval graph:

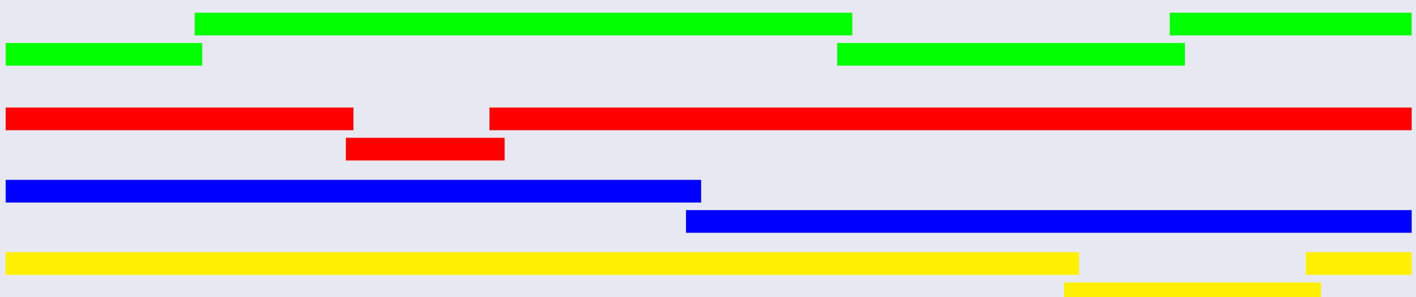


Treewidth – Pathwidth – Tree-depth

A connection to graph classes:



Corresponding interval graph:



Treewidth(G) $\min \omega(G')$ where G' is chordal supergraph of G

Pathwidth(G) $\min \omega(G')$ where G' is interval supergraph of G

Treedepth(G) $\min \omega(G')$ where G' is trivially perfect supergraph of G

Treewidth vs Pathwidth

- Treewidth k is a much wider class than Pathwidth k.
- But most problems have same complexity for both parameters!
  - IND. SET, DOM. SET, STEINER TREE, COLORING,...
  - (HAMILTONICITY?)
- In particular, **almost all** natural problems which are FPT for pathwidth, are FPT for treewidth.

Exception: GRUNDY COLORING

Theorem: GRUNDY COLORING is FPT parameterized by pathwidth but W[1]-hard parameterized by treewidth. [Belmonte, Kim, L., Mitsou, Otachi, ESA 2020 SIDMA 2022].

Background II:
Meta-Theorems

Meta-Theorems and Courcelle's Theorem

- Statements of the form:
  "Every problem in family $\mathcal{F}$ is *tractable*"
  - Family $\mathcal{F}$: often "expressible in FO/MSO or other logic"
  - Tractable: often "FPT parameterized by some parameter"

- Statements of the form:
  "Every problem in family $\mathcal{F}$ is *tractable*"

  - Family $\mathcal{F}$: often "expressible in FO/MSO or other logic"
  - Tractable: often "FPT parameterized by some parameter"

  Courcelle's famous meta-theorem:

  $\boxed{\text{All problems expressible in MSO logic are FPT parameterized by treewidth.}}$

FO logic:

- Two relations: $=$ and $\sim$ (equality, adjacency)
- (Quantified) Variables $x_1, x_2, \ldots$ represent vertices
- Standard boolean connectives $(\vee, \wedge, \neg, \rightarrow)$

Standard Example: 2-Dominating set

$$\exists x_1 \exists x_2 \forall x_3 \left(x_1 = x_3 \vee x_2 = x_3 \vee x_1 \sim x_3 \vee x_2 \sim x_3 \right)$$

FO and MSO logic reminder

FO logic:

- Two relations: $=$ and $\sim$ (equality, adjacency)
- (Quantified) Variables $x_1, x_2, \ldots$ represent vertices
- Standard boolean connectives $(\vee, \wedge, \neg, \rightarrow)$

MSO logic: FO logic plus the following

- $\in$ relation
- (Quantified) **Set** Variables $X_1, X_2, \ldots$ represent sets of vertices

Standard Examples: 3-Coloring, Connectivity

$$
\begin{aligned}
\exists X_1 \exists X_2 \exists X_3 \quad \Big(\forall x_1 \quad & (x_1 \in X_1 \vee x_1 \in X_2 \vee x_1 \in X_3) \wedge \\
\forall x_2 \quad & (x_1 \sim x_2 \rightarrow (\neg(x_1 \in X_1 \wedge x_2 \in X_1)) \wedge \\
& (\neg(x_1 \in X_2 \wedge x_2 \in X_2)) \wedge \\
& (\neg(x_1 \in X_3 \wedge x_2 \in X_3))) \Big)
\end{aligned}
$$

A Closer Look

- Courcelle: If G has treewidth tw, we can check if it satisfies an MSO property ϕ in time
$$f(\mathrm{tw}, \phi) \cdot |G|$$

A Closer Look

- Courcelle: If G has treewidth tw, we can check if it satisfies an MSO property ϕ in time
$$f(\mathrm{tw}, \phi) \cdot |G|$$
- Problem: f is approximately $2^{2^{2^{\cdot^{\cdot^{\cdot^{2^{\mathrm{tw}}}}}}}}$, where the height of the tower is upper-bounded by the number of **quantifier alternations** in ϕ.

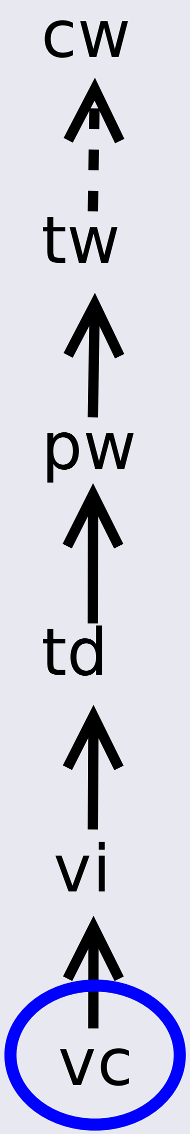
A Closer Look

- Courcelle: If G has treewidth tw, we can check if it satisfies an MSO property ϕ in time
$$f(\mathrm{tw},\phi)\cdot|G|$$
- Problem: f is approximately $2^{2^{2^{\cdot^{\cdot^{\cdot^{2^{\mathrm{tw}}}}}}}}$, where the height of the tower is upper-bounded by the number of **quantifier alternations** in ϕ.
- **Serious Problem**: This tower of exponentials cannot be avoided[1] even for **FO logic on trees**!
  - "The complexity of first-order and monadic second-order logic **revisited**", [Frick and Grohe, APAL 2004].
- **Question**: Does f become nicer if we consider more restricted parameters?

[1] Assuming P≠NP or FPT≠W[1].

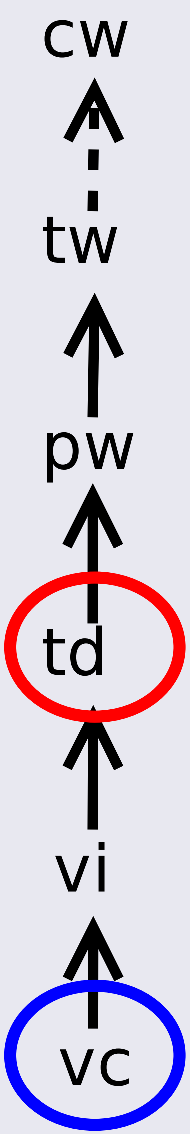
Known Fine-Grained Meta-Theorems

- Vertex Cover
  - MSO with q quantifiers can be decided in $2^{2^{O(\mathrm{vc}+q)}}$
  - FO with q quantifiers can be decided in $2^{O(\mathrm{vc}\cdot q)}q^{O(q)}$
  - These are **optimal under ETH**.
    - There exists fixed MSO formula which cannot be decided in $2^{2^{o(\mathrm{vc})}}$.
- "Algorithmic Meta-Theorems for Restrictions of Treewidth", [L. ESA 2010, Algorithmica 2012].



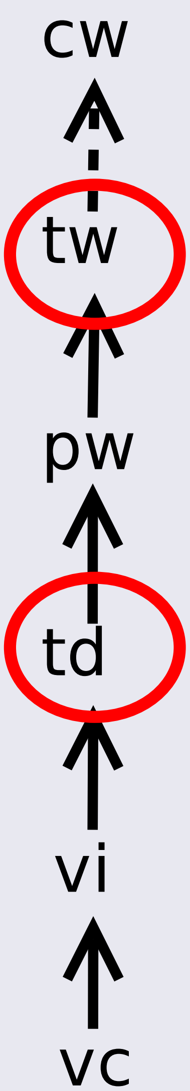
Known Fine-Grained Meta-Theorems (cont'd)

- Tree-depth
  - MSO/FO with q quantifiers can be decided by an algorithm running in time $2^{2^{\cdot^{\cdot^{\cdot^{2^{\mathrm{td}+q}}}}}}$
  - ... where height of tower is at most td (even for large q)
  - This is **optimal under ETH**.
- "Kernelizing MSO Properties of Trees of Fixed Height, and Some Consequences", [Gajarsky and Hlineny, MFCS 2012, LMCS 2015].
- "Model-Checking Lower Bounds for Simple Graphs", [L. ICALP 2013, LMCS 2014].



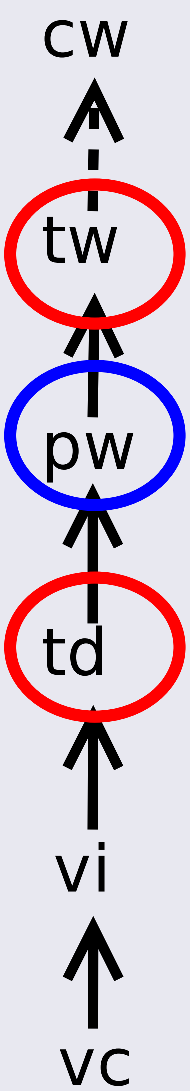
Summary

- For treewidth we can solve MSO in $f(\mathrm{tw}, \phi) \cdot n$
  - But f is **non-elementary**!
  - Inevitable even for $\mathrm{tw} = 1$ and FO logic!
- For tree-depth we can solve MSO in $f(\mathrm{td}, \phi) \cdot n$
  - For each **fixed** value of td, f is an **elementary** function of ϕ.
- Can the same be done for pathwidth?
- (For MSO logic $\rightarrow$ **No** [Frick and Grohe, APAL 2004])



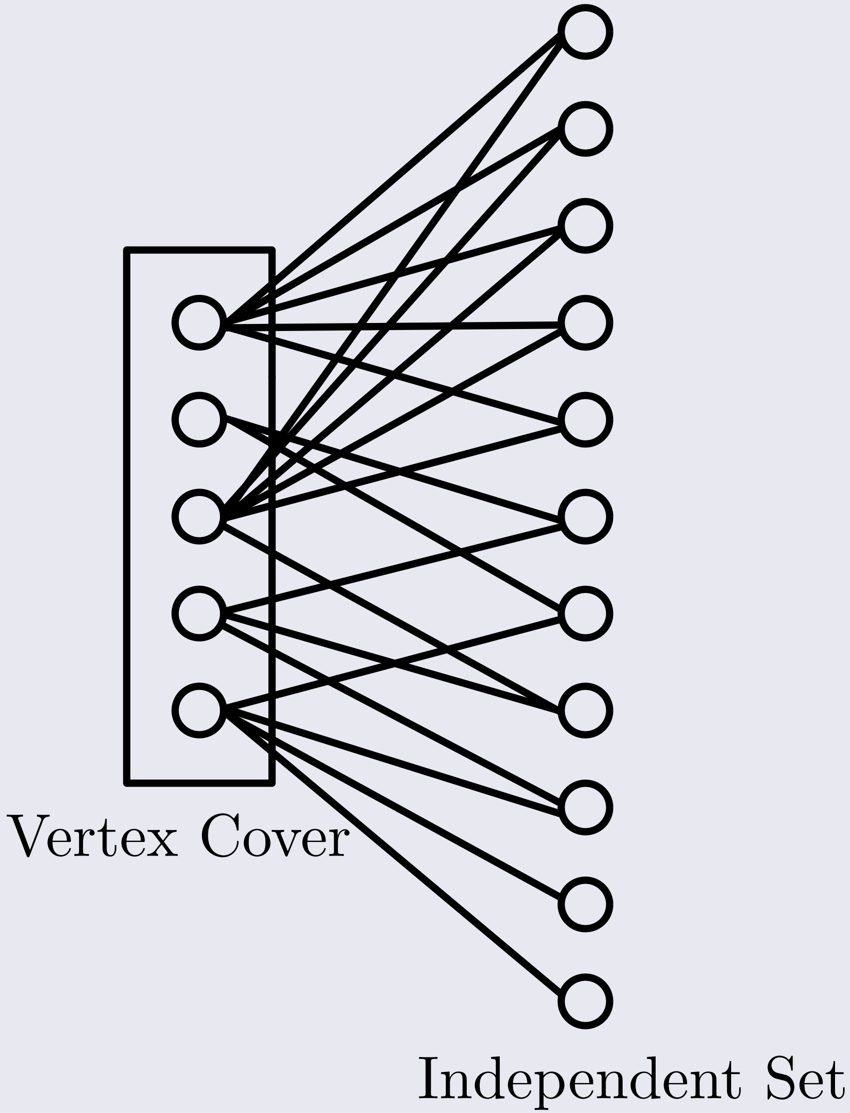
Summary

- For treewidth we can solve MSO in $f(\mathrm{tw}, \phi) \cdot n$
  - But f is **non-elementary**!
  - Inevitable even for $\mathrm{tw} = 1$ and FO logic!
- For tree-depth we can solve MSO in $f(\mathrm{td}, \phi) \cdot n$
  - For each **fixed** value of td, f is an **elementary** function of ϕ.
- Can the same be done for pathwidth?
- (For MSO logic $\rightarrow$ **No** [Frick and Grohe, APAL 2004])



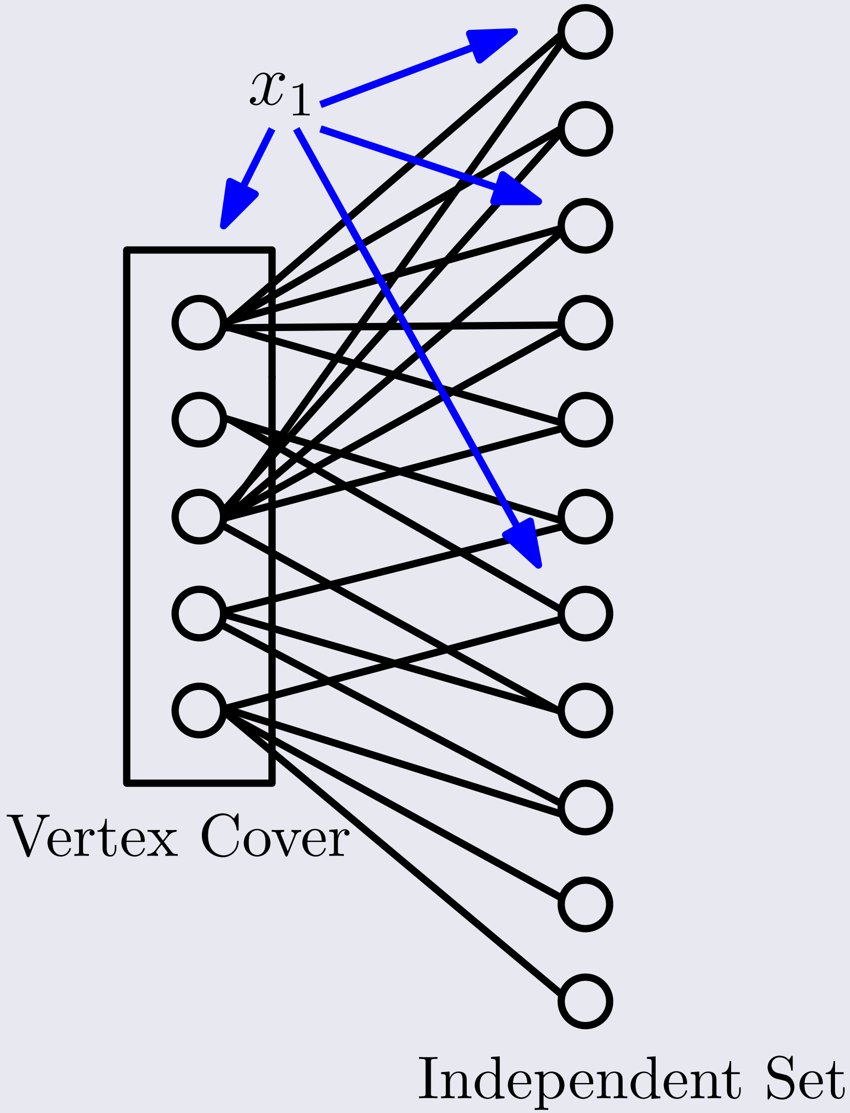
Background III: Techniques

Vertex Cover Meta-Theorem – Reminder



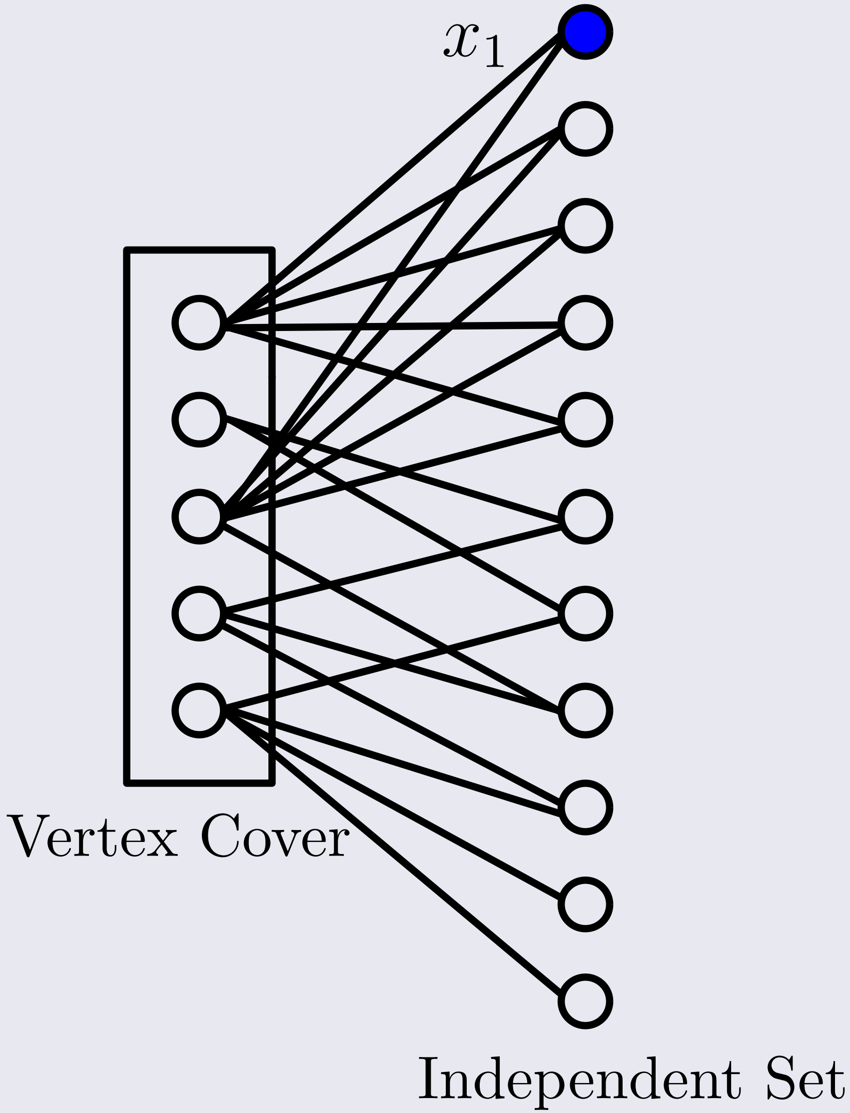
- Given a graph with vertex cover $\mathrm{vc} = 5$
- we want to check an FO property ϕ with $q = 3$ variables.

Vertex Cover Meta-Theorem – Reminder



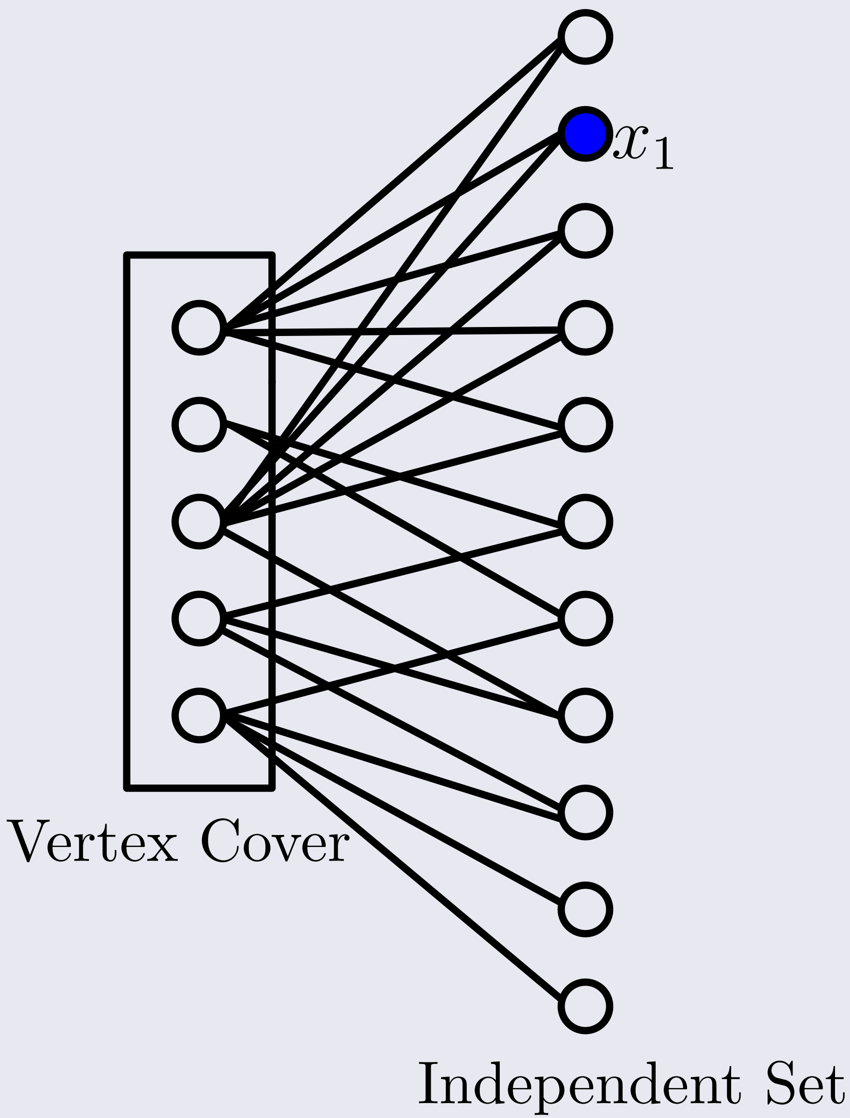
- Sentence has form $\exists x_1 \psi(x_1)$
- We must "place" x_1 somewhere in the graph
- If we try all cases we get n^q running time.

Vertex Cover Meta-Theorem – Reminder



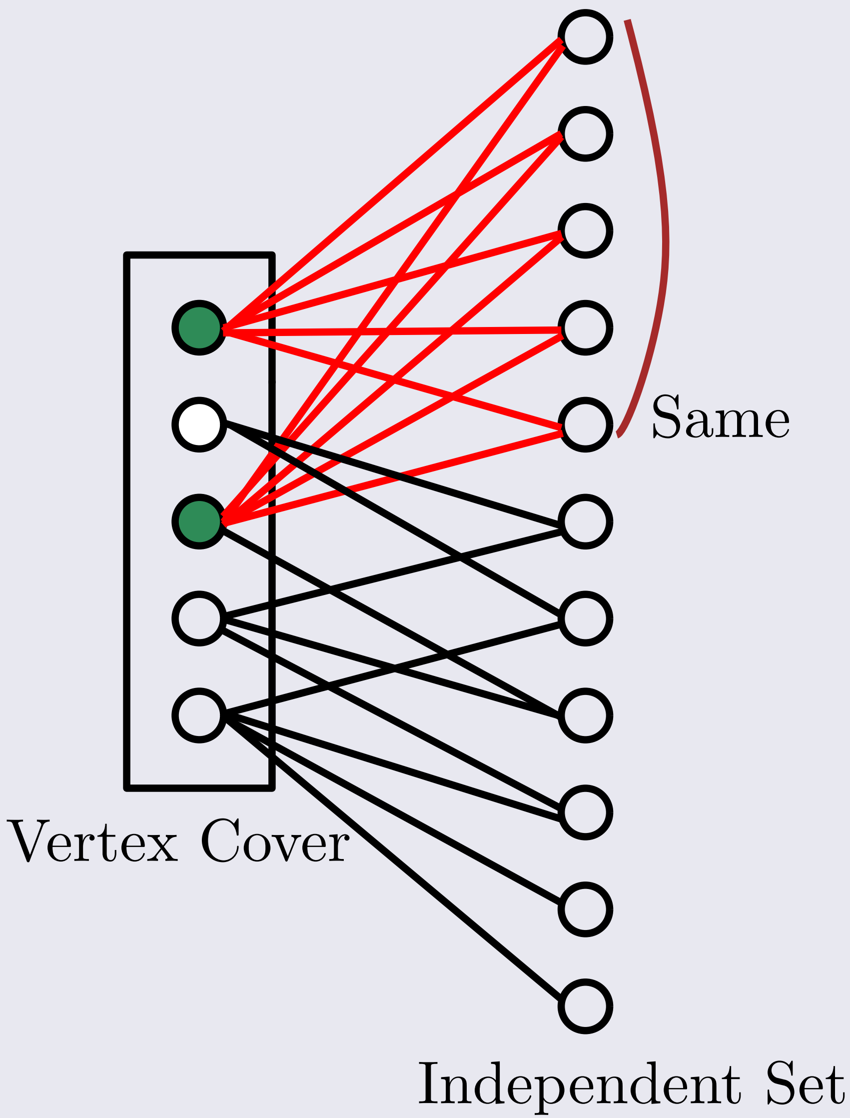
- Sentence has form $\exists x_1 \psi(x_1)$
- We must "place" x_1 somewhere in the graph
- If we try all cases we get n^q running time.

Vertex Cover Meta-Theorem – Reminder



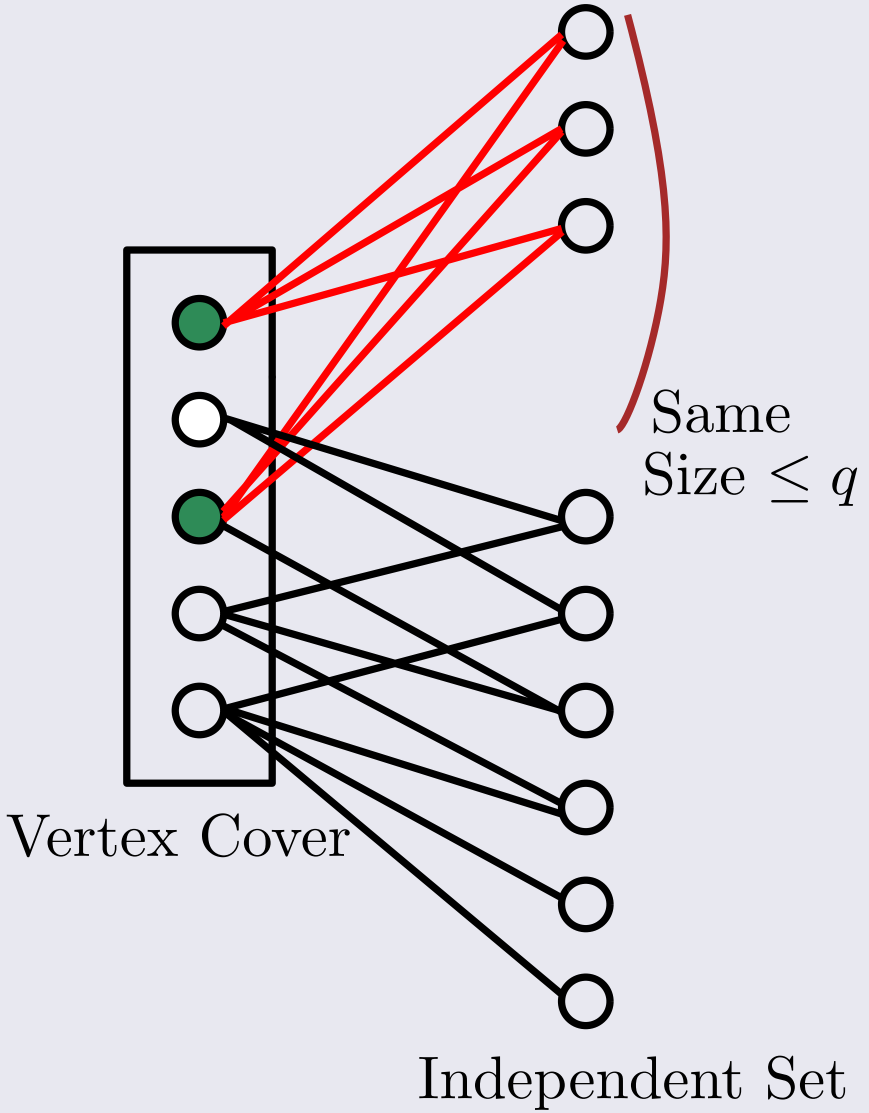
- Sentence has form $\exists x_1 \psi(x_1)$
- We must "place" x_1 somewhere in the graph
- If we try all cases we get n^q running time.

Vertex Cover Meta-Theorem – Reminder



- We observe that some vertices of the independent set have the same neighbors.
- These vertices should be equivalent.

Vertex Cover Meta-Theorem – Reminder

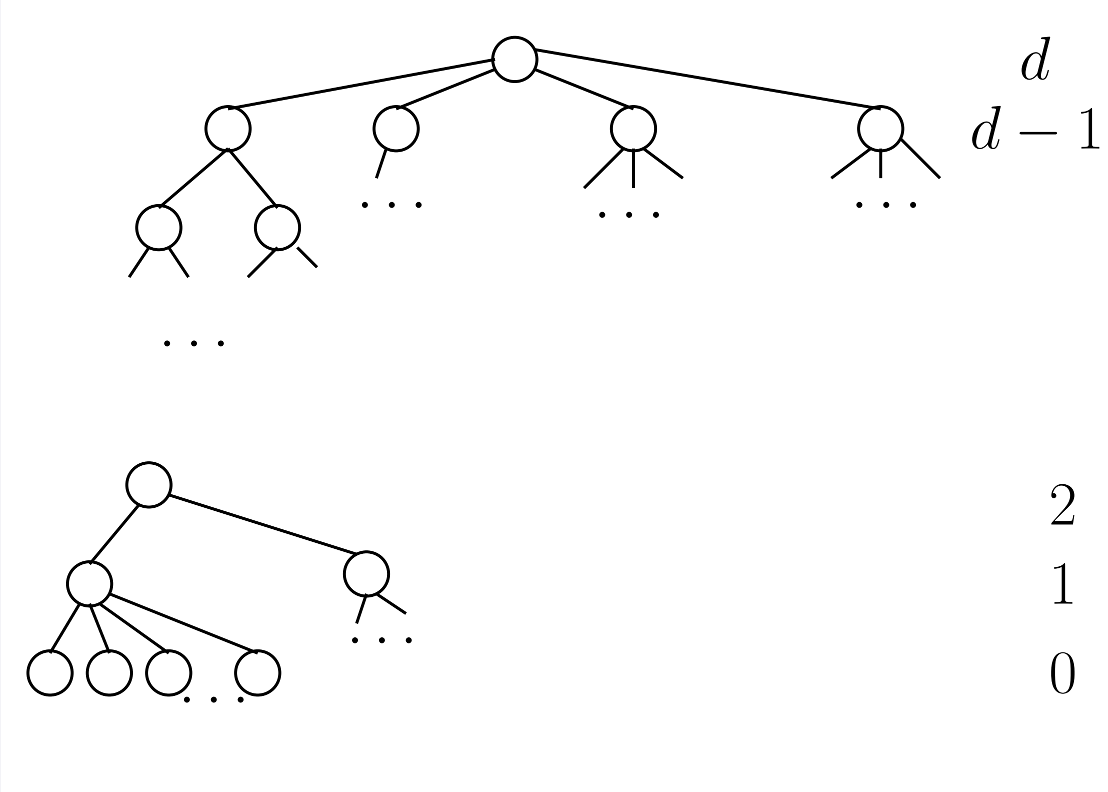


- We observe that some vertices of the independent set have the same neighbors.
- These vertices should be equivalent.
- Key idea: if a group has $> q$ vertices, we can simply remove one!

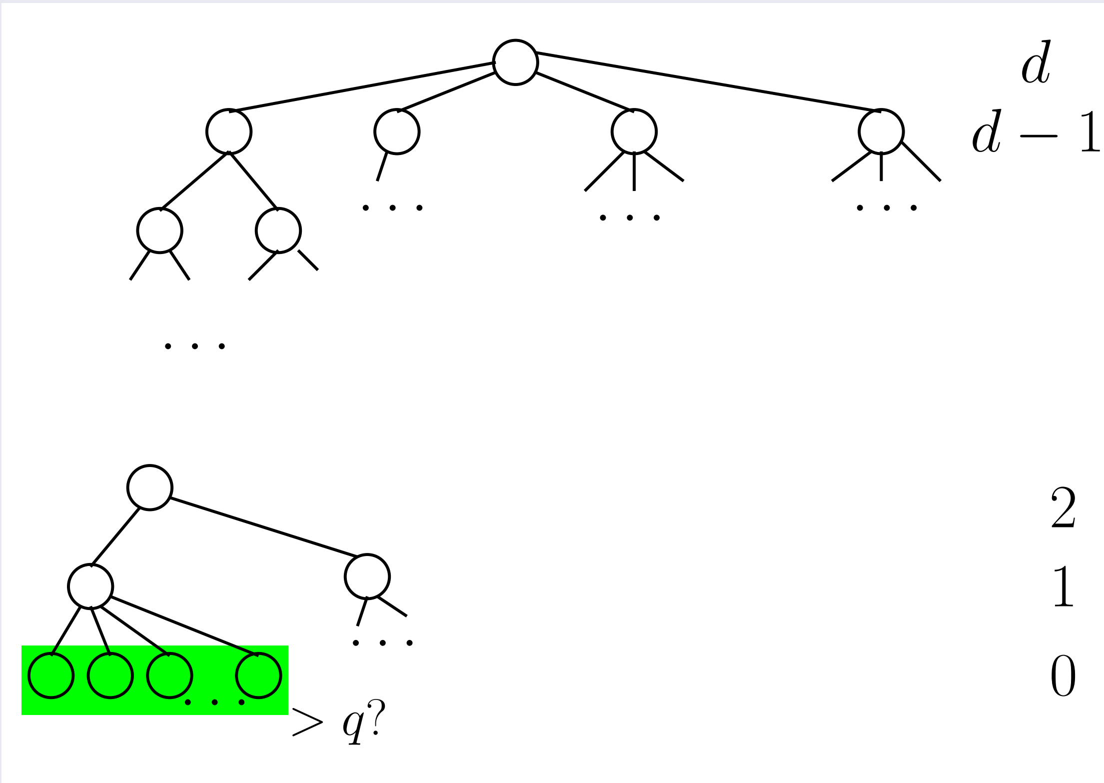
Tree-depth Meta-theorem – Reminder

Tree-depth Meta-theorem – Reminder

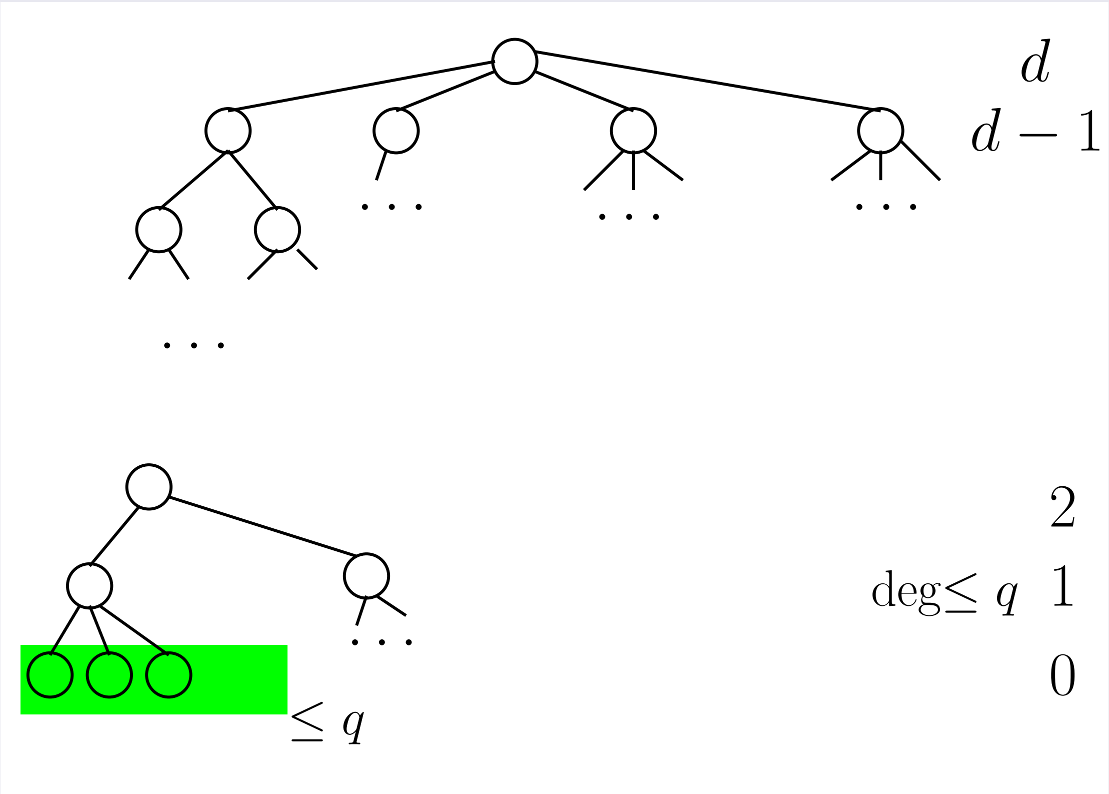
Tree-depth Meta-theorem – Reminder



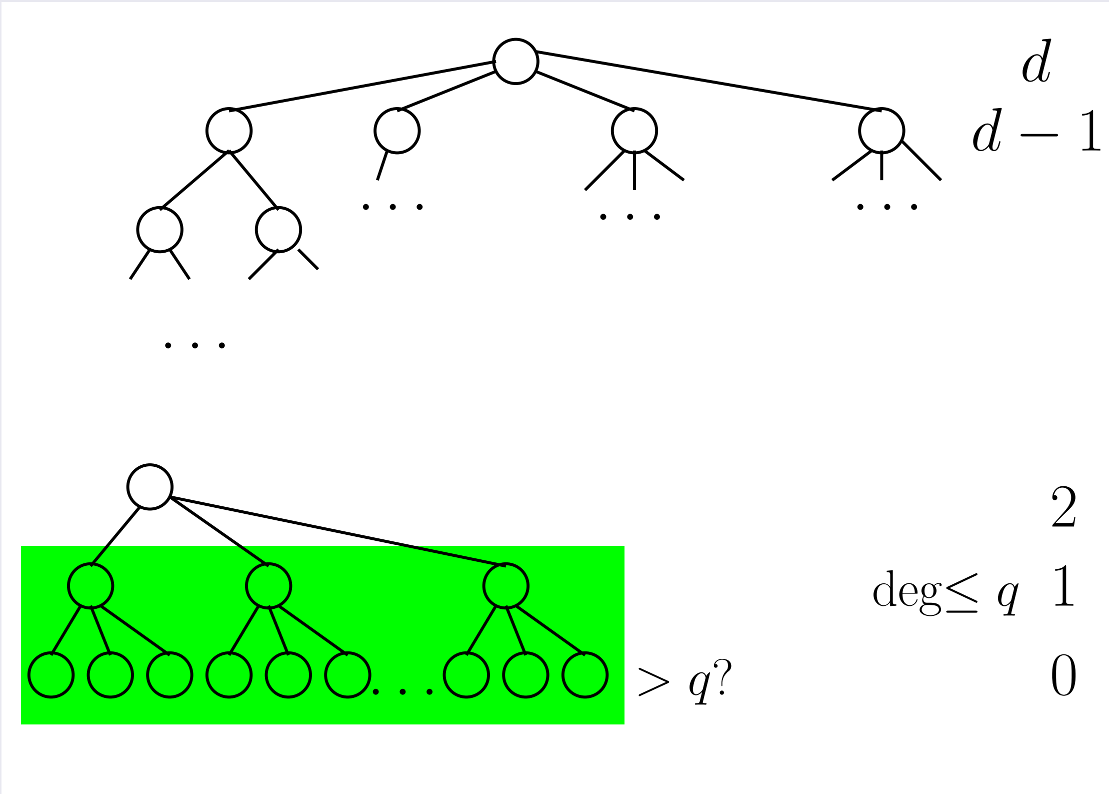
We have a rooted tree with d layers (d fixed)



Apply the previous argument to the bottom layer (leaves)

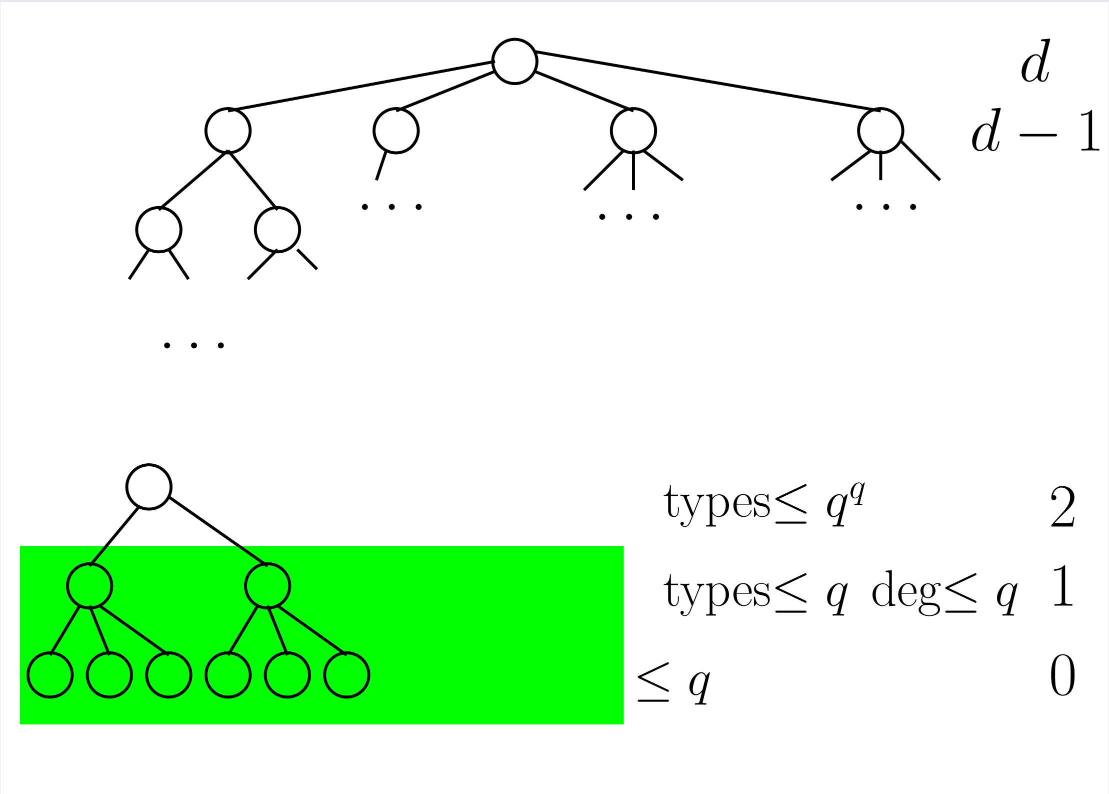


Apply the previous argument to the bottom layer (leaves)

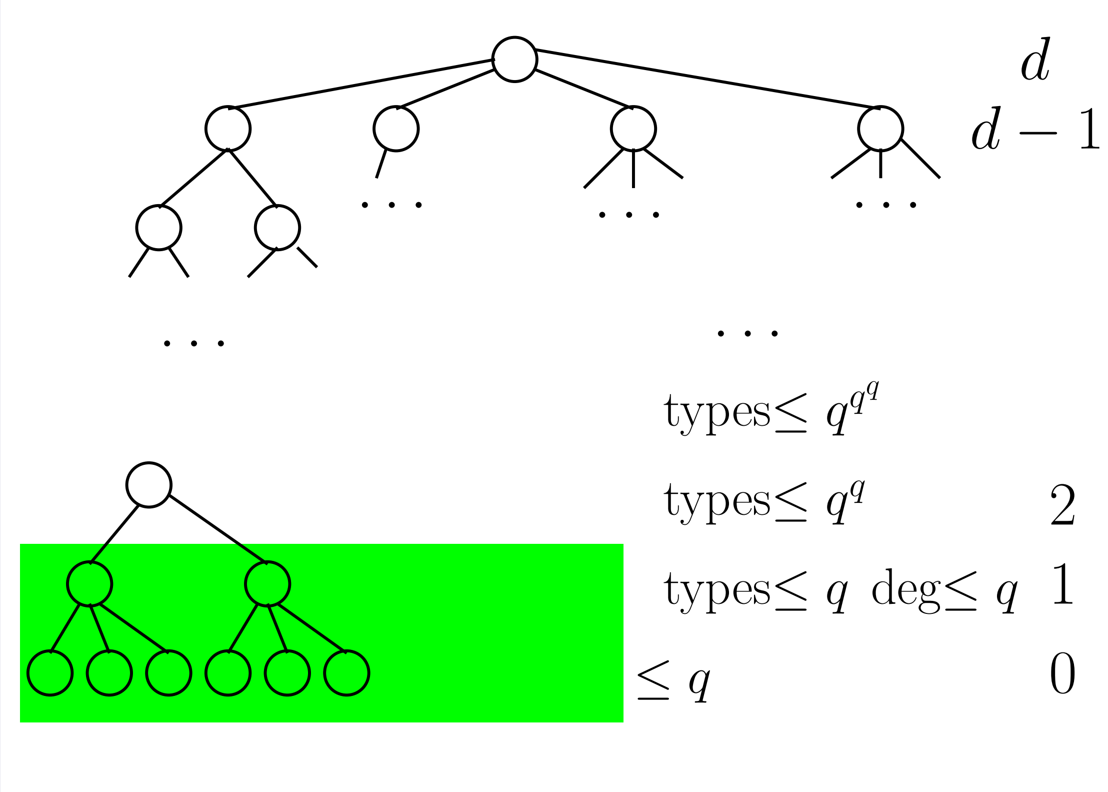


Key intuition: same argument can be applied to level 2, deleting identical sub-trees.

Tree-depth Meta-theorem – Reminder



Key intuition: same argument can be applied to level 2, deleting identical sub-trees.

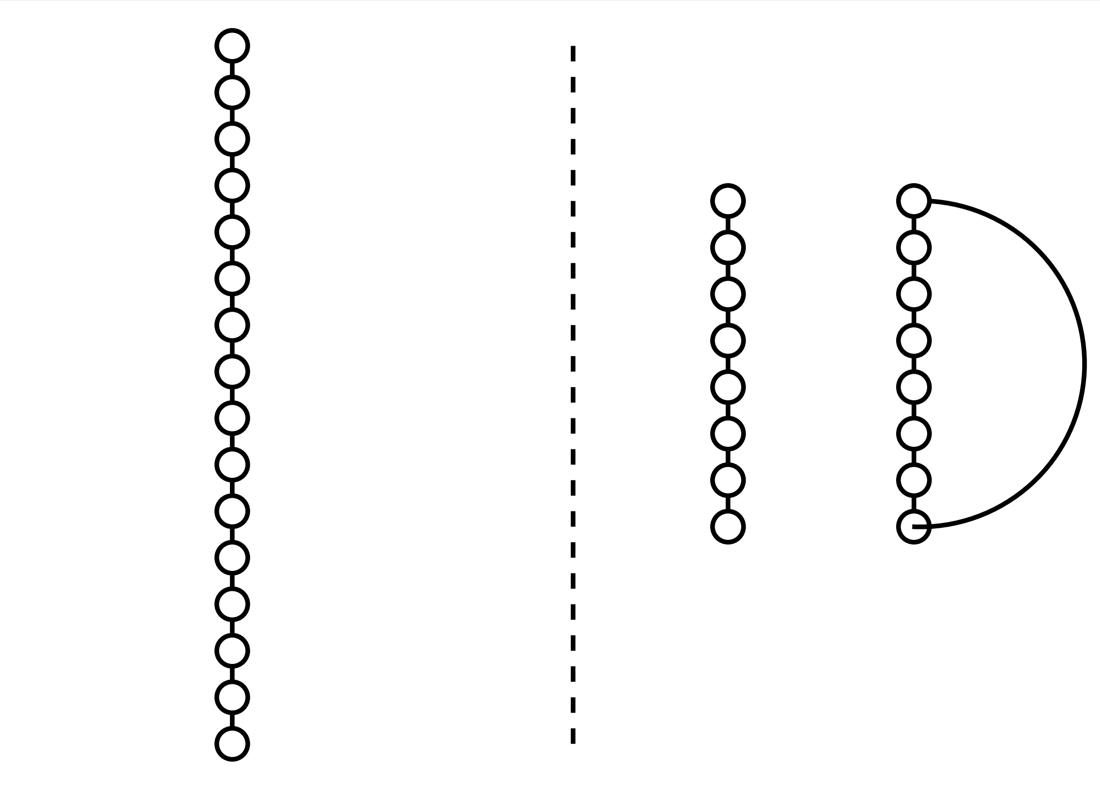


There are q^q different "types" of vertices at level 2. Applying the same argument to level 3, there are q^{q^q} types of vertices of level 3. . . .

In the end graph has bounded size![2]

[2]bounded by a tower of exponentials of height d.

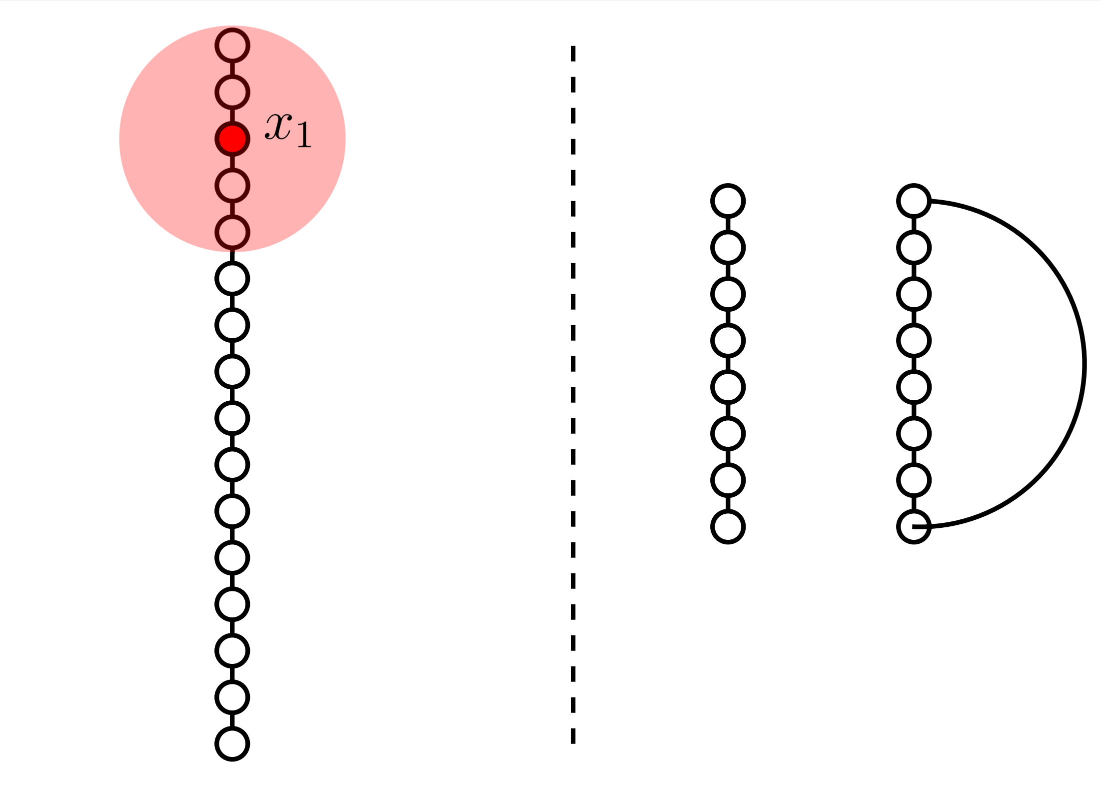
FO logic is local



Classical Example:
FO logic with q quantifiers cannot distinguish a long (say 4^q) path, and a union of a path and a cycle.

CONNECTIVITY cannot be expressed in FO logic!

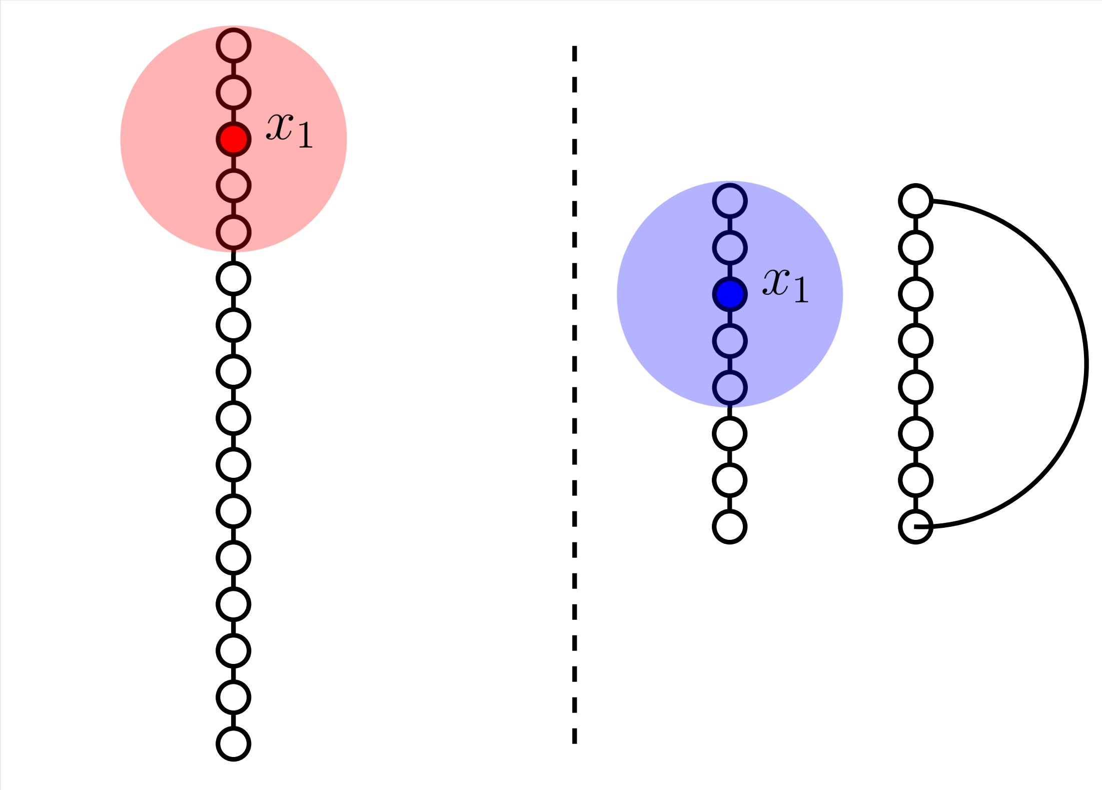
FO logic is local



Classical Example:
FO logic with q quantifiers cannot distinguish a long (say 4^q) path, and a union of a path and a cycle.

CONNECTIVITY cannot be expressed in FO logic!

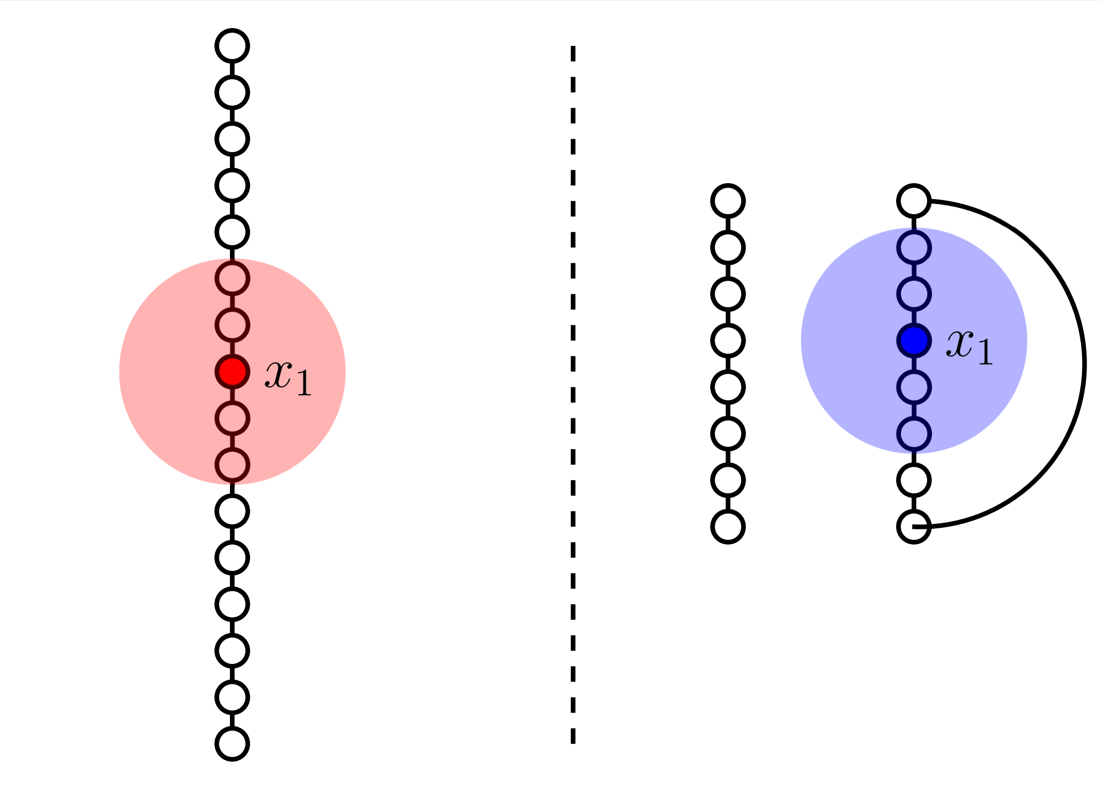
FO logic is local



Classical Example:
FO logic with q quantifiers cannot distinguish a long (say 4^q) path, and a union of a path and a cycle.

CONNECTIVITY cannot be expressed in FO logic!

FO logic is local



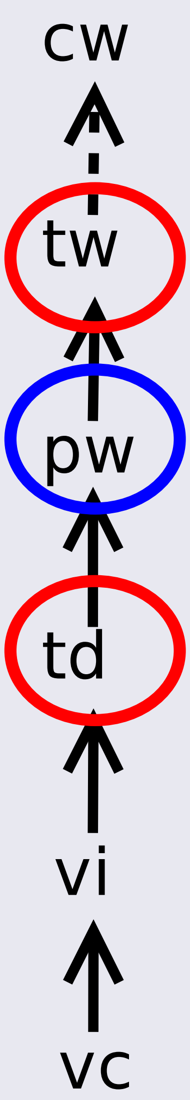
Classical Example:
FO logic with q quantifiers cannot distinguish a long (say 4^q) path, and a union of a path and a cycle.

CONNECTIVITY cannot be expressed in FO logic!

The Algorithm

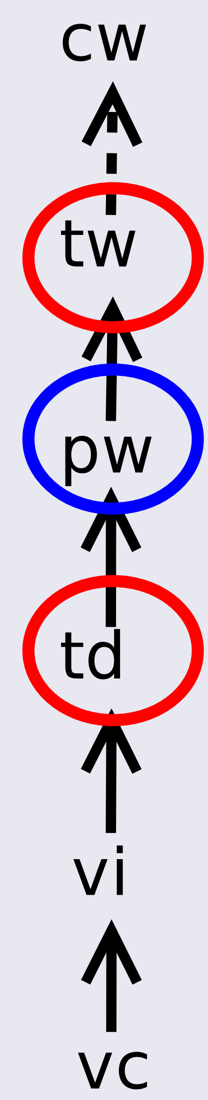
Where we are

Parameter	FO	MSO
Treewidth	Non-elementary on Trees [FrickG04]	Non-elementary on Trees [FrickG04]
Pathwidth		Non-elementary on Caterpillars [FrickG04]
Tree-depth	Elementary [GajarskyH15]	Elementary [GajarskyH15]



Where we are

Parameter	FO	MSO
Treewidth	Non-elementary on Trees [FrickG04]	Non-elementary on Trees [FrickG04]
Pathwidth	Elementary	Non-elementary on Caterpillars [FrickG04]
Tree-depth	Elementary [GajarskyH15]	Elementary [GajarskyH15]

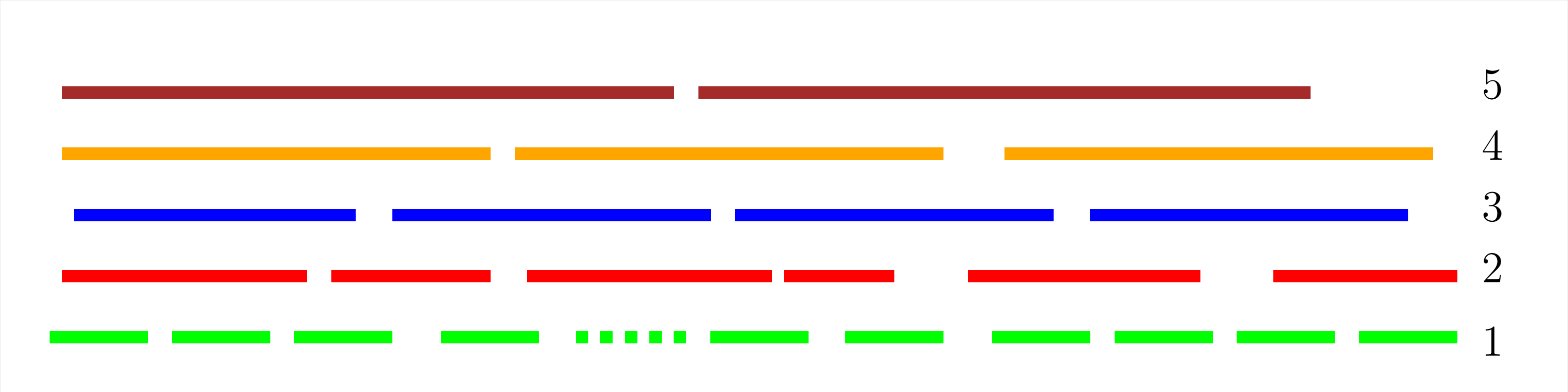


- Last missing case where it was not known if dependence is elementary.
- Complexity different for pathwidth/treewidth (!!)
- Complexity different for FO/MSO (cf. tree-depth)

To obtain algorithm will use:

- A **ranked** version of path decompositions that will make graph hierarchical (like tree-depth).
- A generalized version of the "delete identical parts" argument.
- To find identical parts: a **surgical** operation that relies on the locality of FO logic.

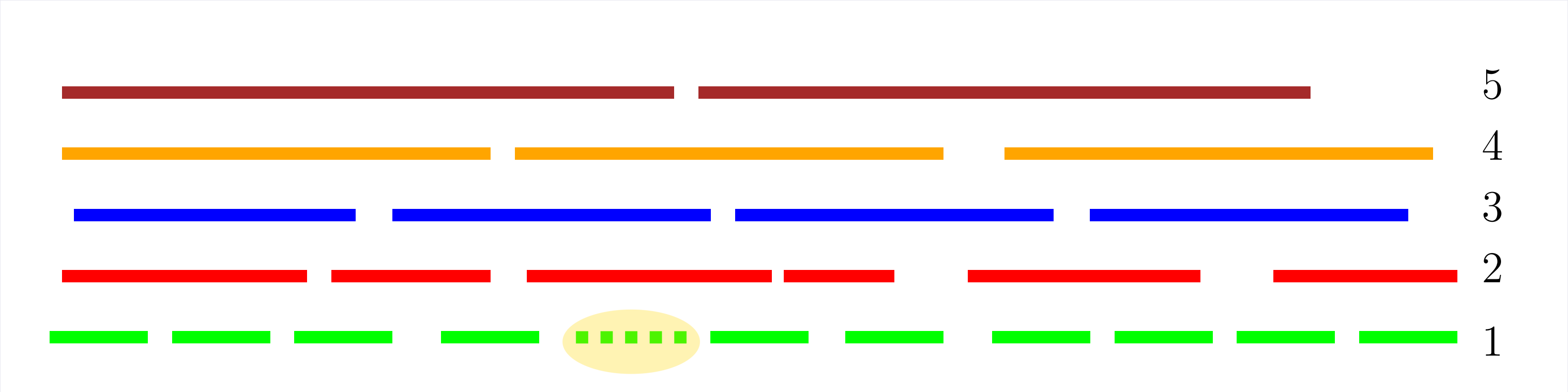
A Surgical Operation – Motivation



- Intuition: try to delete identical parts on lower levels.
- Works well for level 1, there are only $2^{O(\mathrm{pw})}$ types.

- Strategy breaks down at level 2.
- No twins are guaranteed to exist.
- Deleting something makes locally detectable changes to graph.
- Must carefully **cut out** parts to make sure formula validity is not affected.

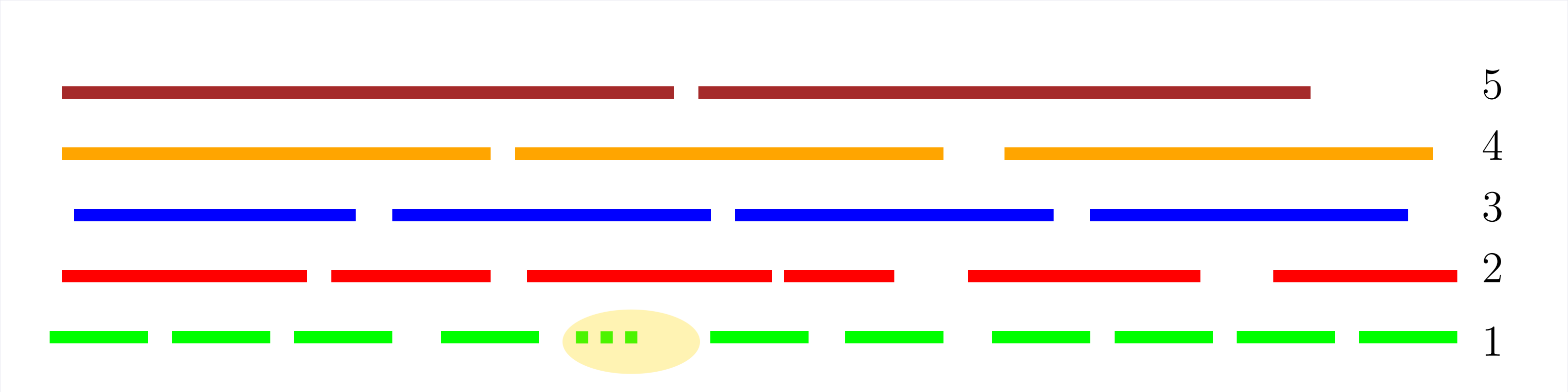
A Surgical Operation – Motivation



- Intuition: try to delete identical parts on lower levels.
- Works well for level 1, there are only $2^{O(\mathrm{pw})}$ types.

- Strategy breaks down at level 2.
- No twins are guaranteed to exist.
- Deleting something makes locally detectable changes to graph.
- Must carefully **cut out** parts to make sure formula validity is not affected.

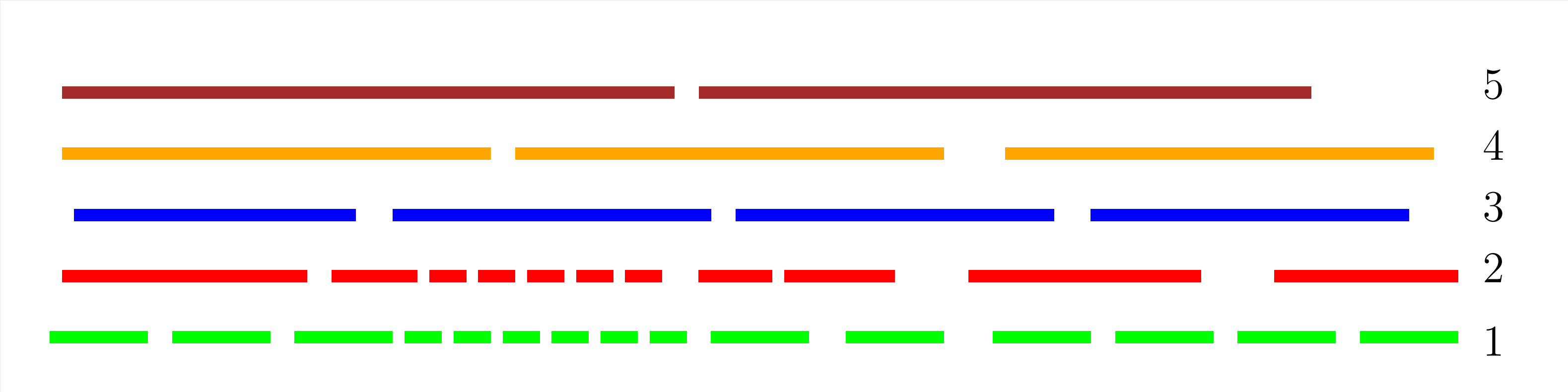
A Surgical Operation – Motivation



- Intuition: try to delete identical parts on lower levels.
- Works well for level 1, there are only $2^{O(\text{pw})}$ types.

- Strategy breaks down at level 2.
- No twins are guaranteed to exist.
- Deleting something makes locally detectable changes to graph.
- Must carefully **cut out** parts to make sure formula validity is not affected.

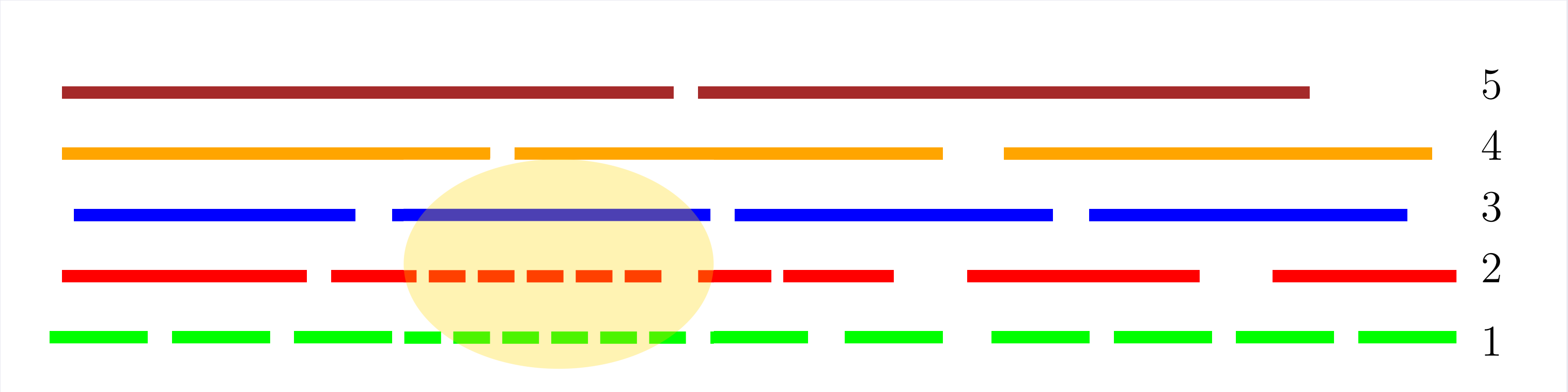
A Surgical Operation – Motivation



- Intuition: try to delete identical parts on lower levels.
- Works well for level 1, there are only $2^{O(\mathrm{pw})}$ types.

- Strategy breaks down at level 2.
- No twins are guaranteed to exist.
- Deleting something makes locally detectable changes to graph.
- Must carefully **cut out** parts to make sure formula validity is not affected.

A Surgical Operation – Motivation



- Intuition: try to delete identical parts on lower levels.
- Works well for level 1, there are only $2^{O(\mathrm{pw})}$ types.

- Strategy breaks down at level 2.
- No twins are guaranteed to exist.
- Deleting something makes locally detectable changes to graph.
- Must carefully **cut out** parts to make sure formula validity is not affected.

A Surgical Operation – Motivation

- Intuition: try to delete identical parts on lower levels.
- Works well for level 1, there are only $2^{O(\mathrm{pw})}$ types.

- Strategy breaks down at level 2.
- No twins are guaranteed to exist.
- Deleting something makes locally detectable changes to graph.
- Must carefully **cut out** parts to make sure formula validity is not affected.

A surgical operation

Idea: Path $\leftrightarrow$ Path+Ring transformation.

- Identify two areas where for a large radius things are similar.
- Cut graph in middle of each area.
- Paste into a main path and a ring.
- Appropriately chosen radius $\rightarrow$ area around each vertex the same $\rightarrow$ FO-equivalent graphs.

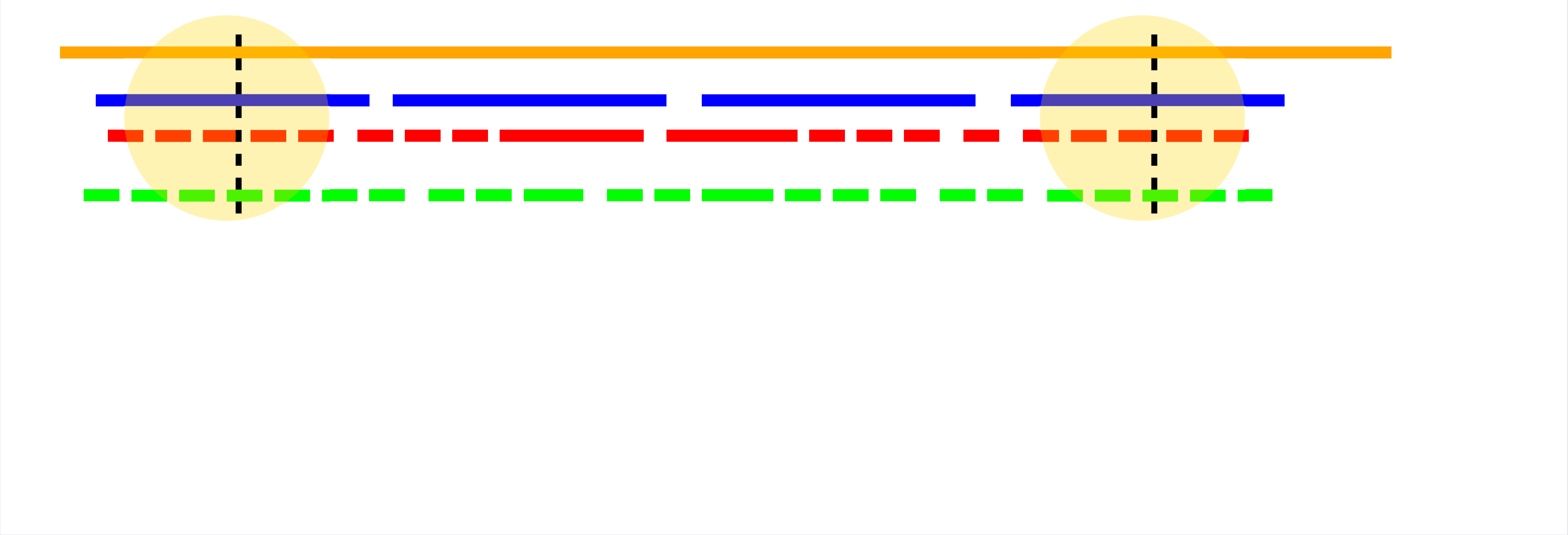
A surgical operation

Idea: Path $\leftrightarrow$ Path+Ring transformation.

- Identify two areas where for a large radius things are similar.
- Cut graph in middle of each area.
- Paste into a main path and a ring.
- Appropriately chosen radius $\rightarrow$ area around each vertex the same $\rightarrow$ FO-equivalent graphs.

A surgical operation

Idea: Path $\leftrightarrow$ Path+Ring transformation.



- Identify two areas where for a large radius things are similar.
- Cut graph in middle of each area.
- Paste into a main path and a ring.
- Appropriately chosen radius $\rightarrow$ area around each vertex the same $\rightarrow$ FO-equivalent graphs.

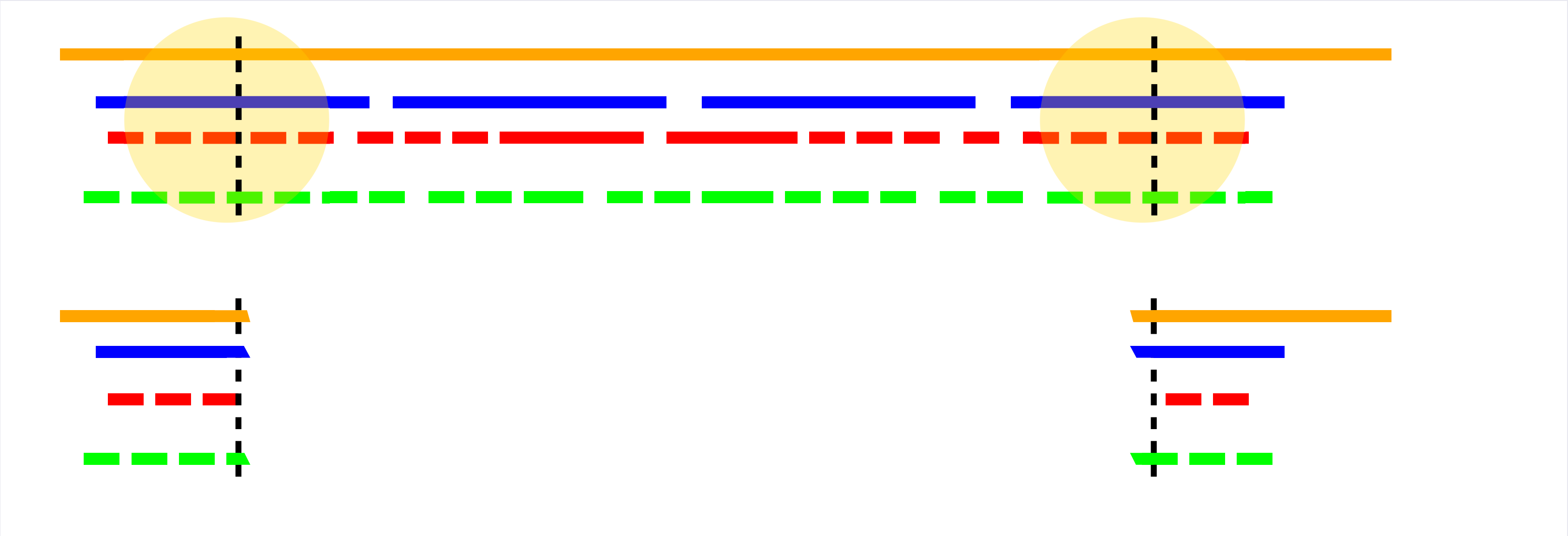
A surgical operation

Idea: Path $\leftrightarrow$ Path+Ring transformation.

- Identify two areas where for a large radius things are similar.
- Cut graph in middle of each area.
- Paste into a main path and a ring.
- Appropriately chosen radius $\rightarrow$ area around each vertex the same $\rightarrow$ FO-equivalent graphs.

A surgical operation

Idea: Path $\leftrightarrow$ Path+Ring transformation.



- Identify two areas where for a large radius things are similar.
- Cut graph in middle of each area.
- Paste into a main path and a ring.
- Appropriately chosen radius $\rightarrow$ area around each vertex the same $\rightarrow$ FO-equivalent graphs.

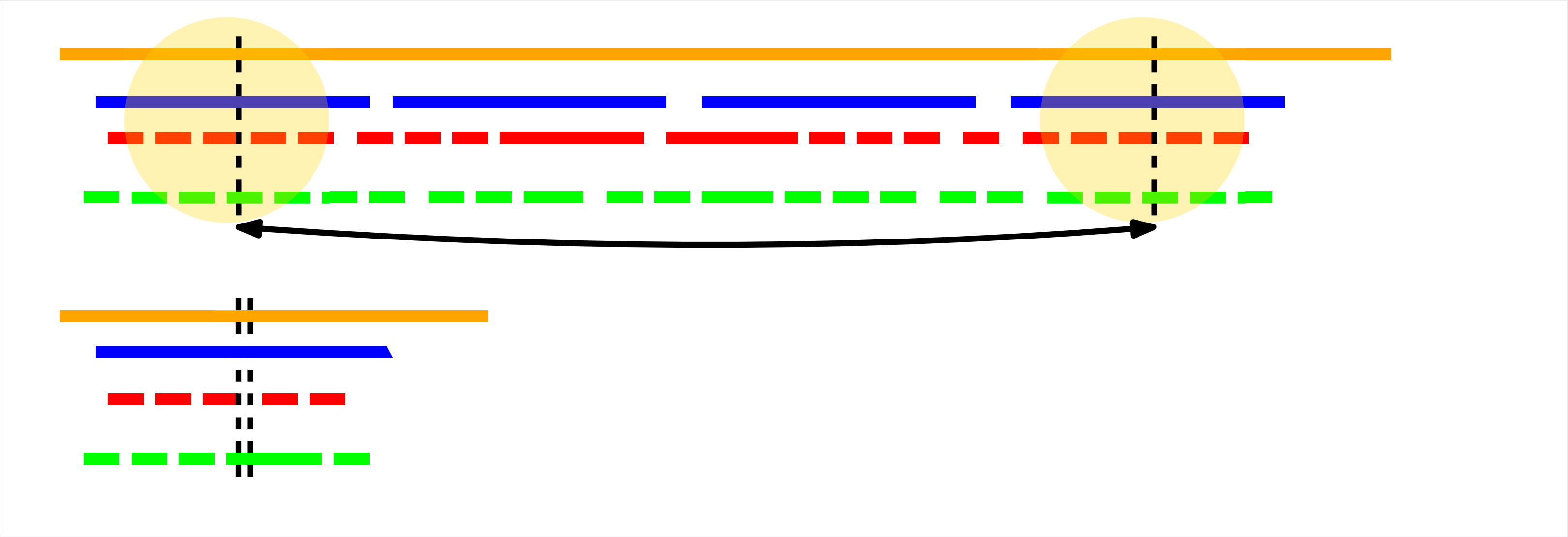
A surgical operation

Idea: Path $\leftrightarrow$ Path+Ring transformation.

- Identify two areas where for a large radius things are similar.
- Cut graph in middle of each area.
- Paste into a main path and a ring.
- Appropriately chosen radius $\rightarrow$ area around each vertex the same $\rightarrow$ FO-equivalent graphs.

A surgical operation

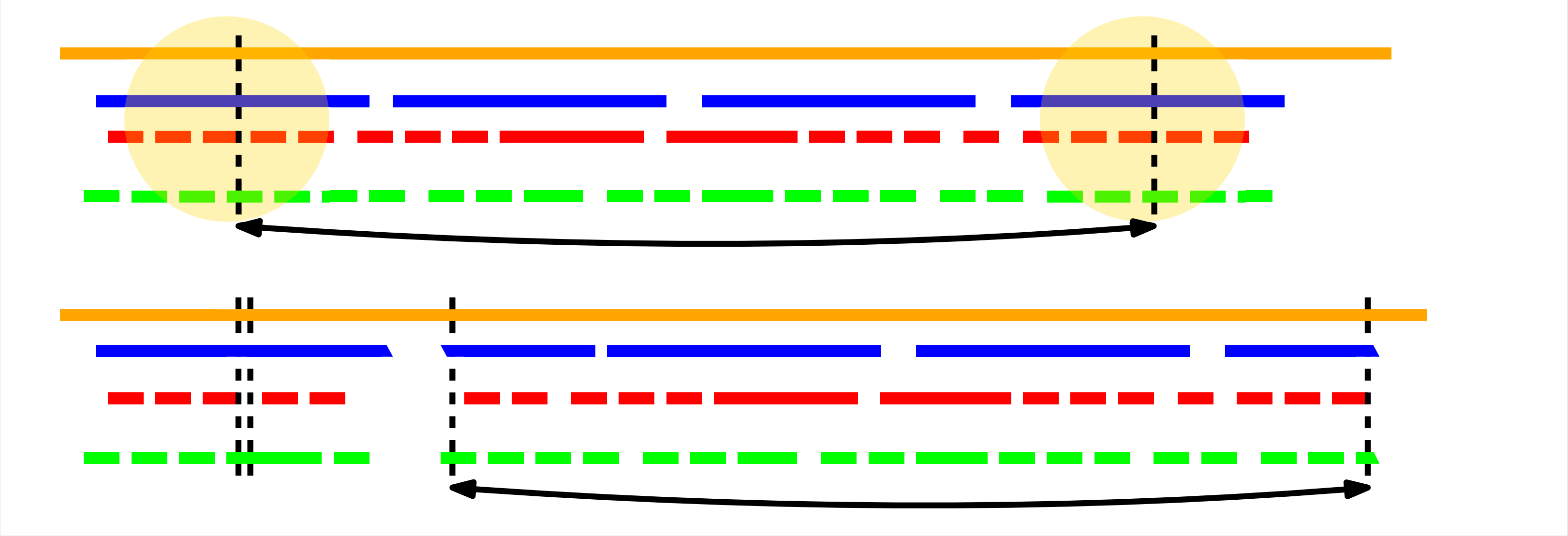
Idea: Path $\leftrightarrow$ Path+Ring transformation.



- Identify two areas where for a large radius things are similar.
- Cut graph in middle of each area.
- Paste into a main path and a ring.
- Appropriately chosen radius $\rightarrow$ area around each vertex the same $\rightarrow$ FO-equivalent graphs.

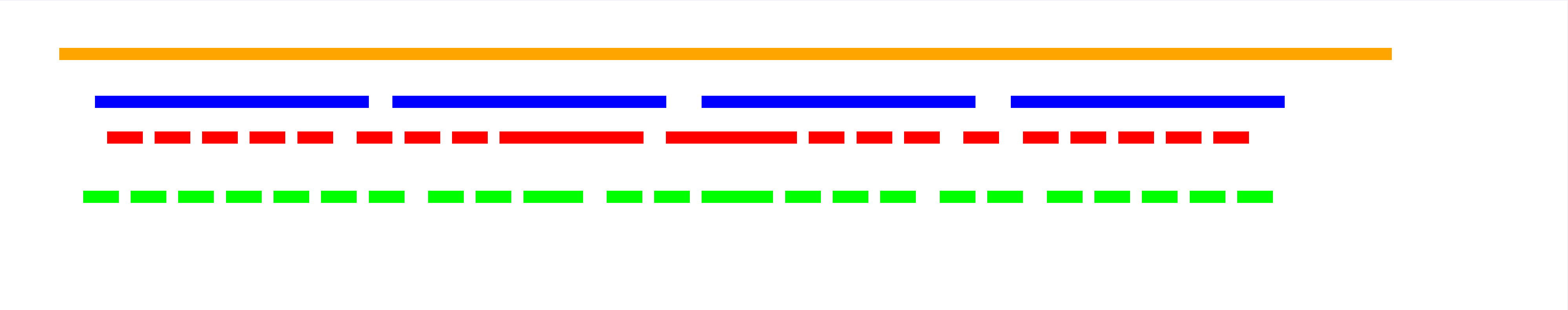
A surgical operation

Idea: Path $\leftrightarrow$ Path+Ring transformation.



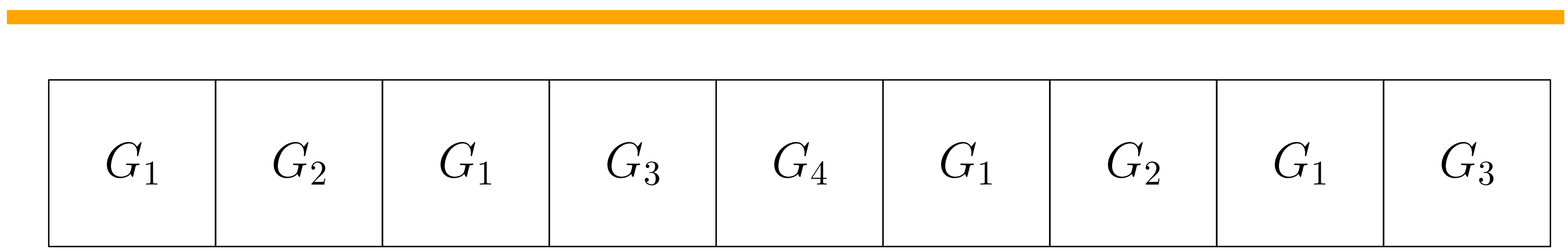
- Identify two areas where for a large radius things are similar.
- Cut graph in middle of each area.
- Paste into a main path and a ring.
- Appropriately chosen radius $\rightarrow$ area around each vertex the same $\rightarrow$ FO-equivalent graphs.

Putting it all together



- Inductive Hypothesis: intervals of color $\leq i \rightarrow$ length at most $f(i,q)$.
- Process color $i+1$:
  - Find $q+1$ identical blocks where surgical operation applies
  - Argue that one can be shortened.
  - $\rightarrow$ interval has length $\leq f(i+1,q)$, (which is $> 2^{f(i,q)}$).
- End result: bounded-degree graph.

Putting it all together



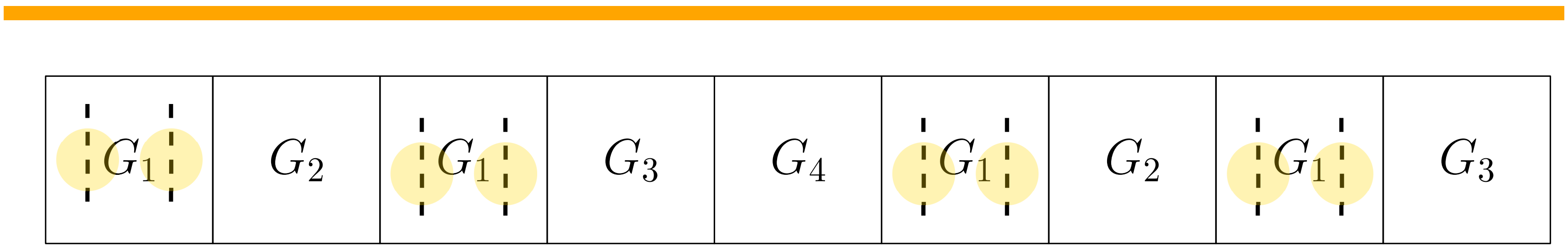
- Inductive Hypothesis: intervals of color $\leq i \rightarrow$ length at most $f(i, q)$.
- Process color $i + 1$:
  - Find $q + 1$ identical blocks where surgical operation applies
  - Argue that one can be shortened.
  - $\rightarrow$ interval has length $\leq f(i + 1, q)$, (which is $> 2^{f(i,q)}$).
- End result: bounded-degree graph.

Putting it all together

G_1	G_2	G_1	G_3	G_4	G_1	G_2	G_1	G_3

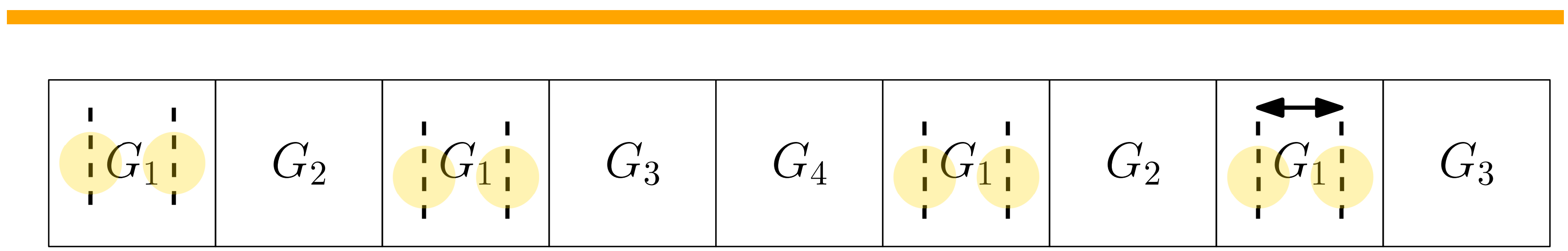
- Inductive Hypothesis: intervals of color $\leq i \rightarrow$ length at most $f(i,q)$.
- Process color $i+1$:
  - Find $q+1$ identical blocks where surgical operation applies
  - Argue that one can be shortened.
  - $\rightarrow$ interval has length $\leq f(i+1,q)$, (which is $> 2^{f(i,q)}$).
- End result: bounded-degree graph.

Putting it all together



- Inductive Hypothesis: intervals of color $\leq i \to$ length at most $f(i, q)$.
- Process color $i + 1$:
  - Find $q + 1$ identical blocks where surgical operation applies
  - Argue that one can be shortened.
  - $\to$ interval has length $\leq f(i + 1, q)$, (which is $> 2^{f(i,q)}$).
- End result: bounded-degree graph.

Putting it all together



- Inductive Hypothesis: intervals of color $\leq i \rightarrow$ length at most $f(i,q)$.
- Process color $i+1$:
  - Find $q+1$ identical blocks where surgical operation applies
  - Argue that one can be shortened.
  - $\rightarrow$ interval has length $\leq f(i+1,q)$, (which is $> 2^{f(i,q)}$).
- End result: bounded-degree graph.

Conclusions

Conclusions

- FO model-checking is **elementary** for graphs of bounded pathwidth.
  - Surprising because tw/pw are usually similar.
  - Surprising that this was not known!

Open problems:

- Extension to dense graphs?
  - Extension to linear clique-width impossible due to hardness for threshold graphs.
- Other graph classes with elementary model-checking?
- **Realistic** meta-theorems?

Conclusions

- FO model-checking is **elementary** for graphs of bounded pathwidth.
  - Surprising because tw/pw are usually similar.
  - Surprising that this was not known!

Open problems:

- Extension to dense graphs?
  - Extension to linear clique-width impossible due to hardness for threshold graphs.
- Other graph classes with elementary model-checking?
- **Realistic** meta-theorems?

Thank you!