Shrub-Depth

a successful depth measure for dense graphs

Petr Hliněný

Faculty of Informatics, Masaryk University

Brno, Czech Republic

Shrub-Depth

a successful depth measure for dense graphs

Petr Hliněný

Faculty of Informatics, Masaryk University

Brno, Czech Republic

Ingredients: joint results with

J. Gajarský, R. Ganian, O. Kwon, J. Nešetřil, J. Obdržálek, S. Ordyniak, P. Ossona de Mendez

Measuring Width or Depth?

- **Being close to a TREE – "•-width"**

SPARSE

tree-width / branch-width
– showing a *structure*

DENSE

clique-width / rank-width
– showing a *construction*

Measuring Width or Depth?

- **Being close to a TREE – "•-width"**

SPARSE

tree-width / branch-width
– showing a *structure*

DENSE

clique-width / rank-width
– showing a *construction*

- **Being close to a STAR – "•-depth"**

SPARSE

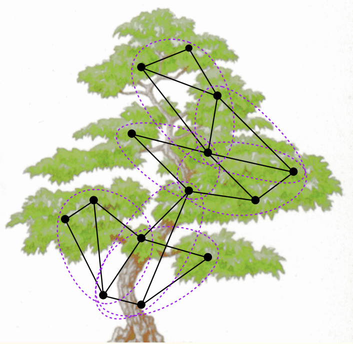
tree-depth
– containment in a *structure*

DENSE

???
(will show)

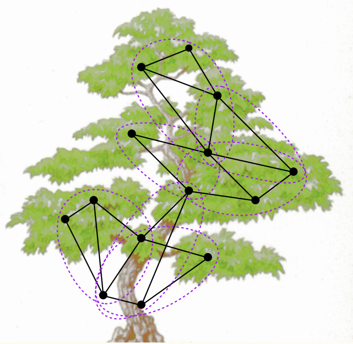
1 Recall: Width Measures

Tree-width $\boldsymbol{tw(G) \leq k}$ if whole G can be covered by bags of size $\leq k+1$, arranged in a "tree-like fashion".



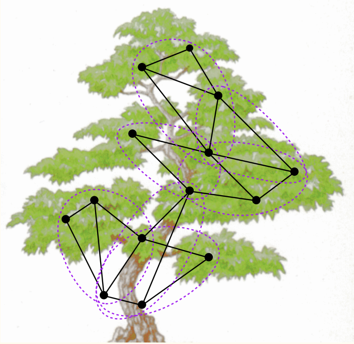
1 Recall: Width Measures

Tree-width $tw(G) \leq k$ if whole G can be covered by bags of size $\leq k+1$, arranged in a "tree-like fashion".



The underlying idea: G is recursively decomposed along small v. separators.

1 Recall: Width Measures



Tree-width $tw(G) \le k$ if whole G can be covered by bags of size $\le k+1$, arranged in a "tree-like fashion".

The underlying idea: G is recursively decomposed along small v. separators.

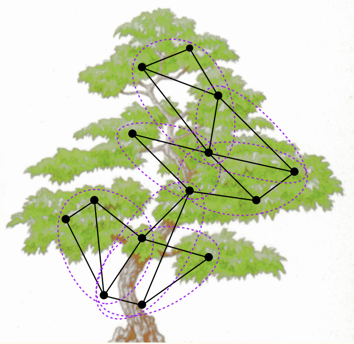
Structural properties

- Monotone under subgraphs and *minors*,

1 Recall: Width Measures

Tree-width $tw(G) \le k$ if whole G can be covered by bags of size $\le k+1$, arranged in a "tree-like fashion".

The underlying idea: G is recursively decomposed along small v. separators.



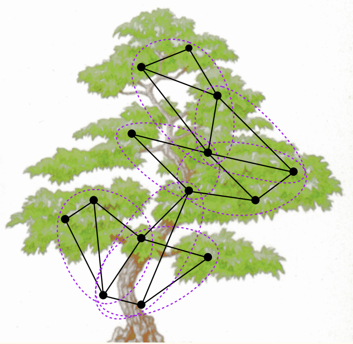
Structural properties

- Monotone under subgraphs and *minors*,
- degenerate and "very" sparse,

1 Recall: Width Measures

Tree-width $tw(G) \leq k$ if whole G can be covered by bags of size $\leq k+1$, arranged in a "tree-like fashion".

The underlying idea: G is recursively decomposed along small v. separators.

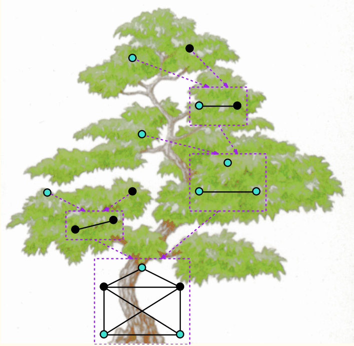


Structural properties

- Monotone under subgraphs and *minors*,
- degenerate and "very" sparse,
- asymptotically equivalent to NO large grid minor.

Clique-width $cwd(G) \leq k$ if G given by a k-*expression* (over k-labelled gr.),

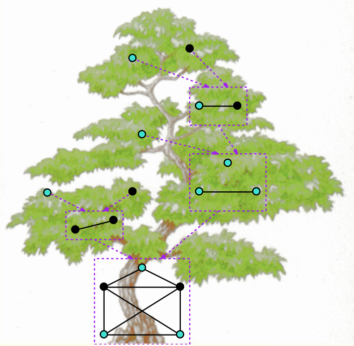
k-expression $\sim$ disjoint unions, relabelling, edge-add. between labels.



Clique-width $\boldsymbol{cwd(G) \leq k}$ if G given by a *k-expression* (over k-labelled gr.),

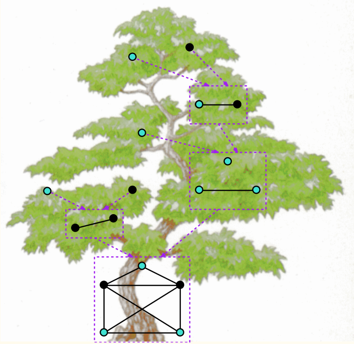
k-expression $\sim$ disjoint unions, relabelling, edge-add. between labels.

The underlying idea: G rec. constructed in a way that only k groups of vertices can be distiguished at any moment.



Clique-width $\boldsymbol{cwd(G) \leq k}$ if G given by a *k-expression* (over k-labelled gr.),

k-expression $\sim$ disjoint unions, relabelling, edge-add. between labels.



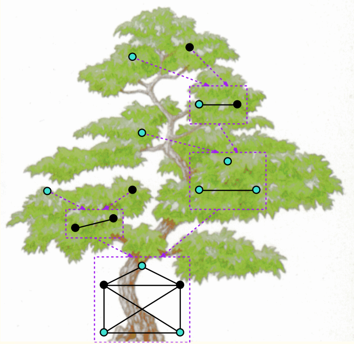
The underlying idea: G rec. constructed in a way that only k groups of vertices can be distiguished at any moment.

Structural properties

- Preserved by ind. subgraphs and "*vertex-minors*" (asympt.),

Clique-width $\boldsymbol{cwd(G) \leq k}$ if G given by a *k-expression* (over k-labelled gr.),

k-expression $\sim$ disjoint unions, relabelling, edge-add. between labels.



The underlying idea: G rec. constructed in a way that only k groups of vertices can be distiguished at any moment.

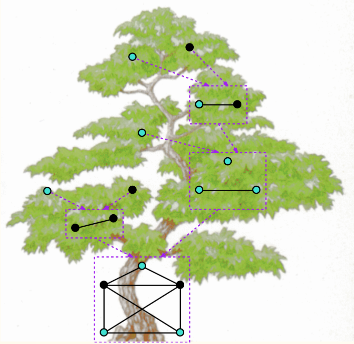
Structural properties

- Preserved by ind. subgraphs and "*vertex-minors*" (asympt.),
- no nice "excluded something" characterization known so far,

Clique-width $\boldsymbol{cwd(G) \leq k}$ if G given by a *k-expression* (over k-labelled gr.),

k-expression $\sim$ disjoint unions, relabelling, edge-add. between labels.

The underlying idea: G rec. constructed in a way that only k groups of vertices can be distiguished at any moment.

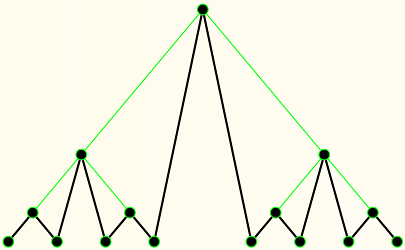


Structural properties

- Preserved by ind. subgraphs and "*vertex-minors*" (asympt.),
- no nice "excluded something" characterization known so far,
- but can be characterized by MSO_1 interpretations into trees.

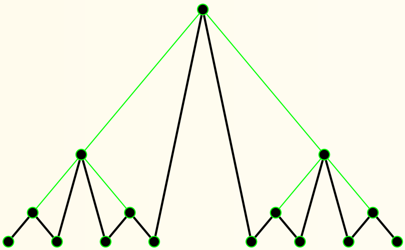
and Depth Measures

Tree-depth $td(G) \le k$ if whole G is contained in the closure of a rooted forest of height $\le k+1$.



and Depth Measures

Tree-depth $td(G) \leq k$ if whole G is contained in the closure of a rooted forest of height $\leq k+1$.

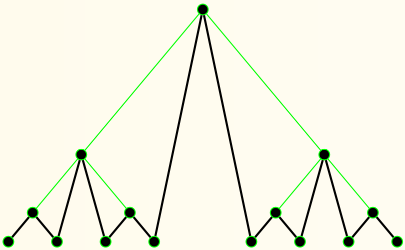


Structural properties

- Monotone under subgraphs and *minors*,

and Depth Measures

Tree-depth $\boldsymbol{td(G) \leq k}$ if whole G is contained in the closure of a rooted forest of height $\leq k+1$.

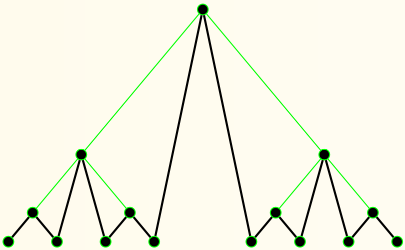


Structural properties

- Monotone under subgraphs and *minors*,
- again degenerate and "very" sparse,

and Depth Measures

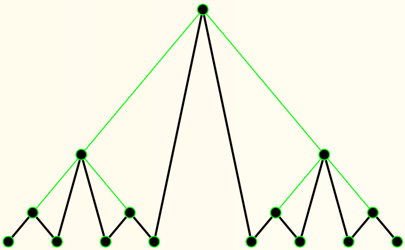
Tree-depth $td(G) \leq k$ if whole G is contained in the closure of a rooted forest of height $\leq k+1$.



Structural properties

- Monotone under subgraphs and *minors*,
- again degenerate and "very" sparse,
- equiv. also to bounding the height of a tree-decomposition,

and Depth Measures



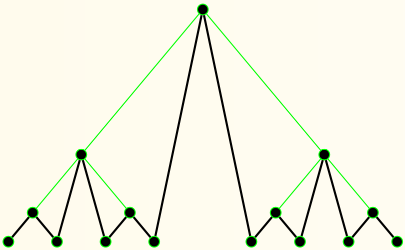
Tree-depth $\boldsymbol{td(G) \leq k}$ if whole G is contained in the closure of a rooted forest of height $\leq k+1$.

Structural properties

- Monotone under subgraphs and *minors*,
- again degenerate and "very" sparse,
- equiv. also to bounding the height of a tree-decomposition,
- asymptotically equivalent to a no long path subgraph,

and Depth Measures

Tree-depth $td(G) \leq k$ if whole G is contained in the closure of a rooted forest of height $\leq k+1$.



Structural properties

- Monotone under subgraphs and *minors*,
- again degenerate and "very" sparse,
- equiv. also to bounding the height of a tree-decomposition,
- asymptotically equivalent to a no long path subgraph,
- and well-behaved wrt. MSO_2 interpretations.

“Clique-depth” ???

“Clique-depth” ???

How to restrict the height of k-expression? Not working...

(though, *NLC-width* provides some hint, and further simplified)

"Clique-depth" ???

How to restrict the height of k-expression? Not working...

(though, *NLC-width* provides some hint, and further simplified)

Desired structural properties

- Ones similar to clique-width, but of small depth?

  (Related to clique-width as tree-depth related to tree-width...)

"Clique-depth" ???

How to restrict the height of k-expression? Not working...

(though, *NLC-width* provides some hint, and further simplified)

Desired structural properties

- Ones similar to clique-width, but of small depth?
  (Related to clique-width as tree-depth related to tree-width...)
- Small also on some dense graphs, stable under complements.
  Recursive construction with limited inform. and of small depth.

"Clique-depth" ???

How to restrict the height of k-expression? Not working...

(though, *NLC-width* provides some hint, and further simplified)

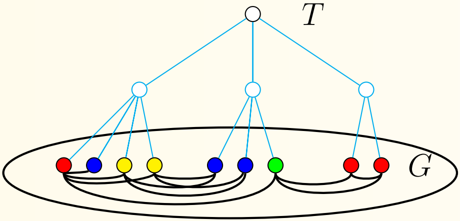
Desired structural properties

- Ones similar to clique-width, but of small depth?
  (Related to clique-width as tree-depth related to tree-width...)
- Small also on some dense graphs, stable under complements.
  Recursive construction with limited inform. and of small depth.
- Stable under FO / MSO_1 interpretations!

2 Tree-models and Shrub-depth

Tree-model of m colours and depth d:

- a rooted tree T of height d,

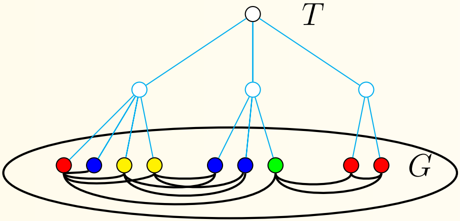


2 Tree-models and Shrub-depth

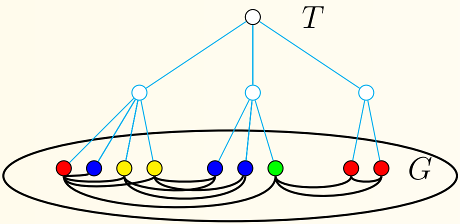
Tree-model of m colours and depth d:

- a rooted tree T of height d,
- leaves are the **vertices of G**,

  each leaf has one of m colours,



2 Tree-models and Shrub-depth



Tree-model of m colours and depth d:

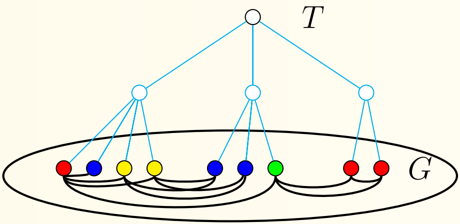
- a rooted tree T of height d,
- leaves are the **vertices of G**,

  each leaf has one of m colours,
- an associated *signature* determining the **edge set of G** as follows:

  for $i = 1, 2, \ldots, d$, let u and v be leaves with the least common ancestor at height i in T,

2 Tree-models and Shrub-depth

Tree-model of m colours and depth d:

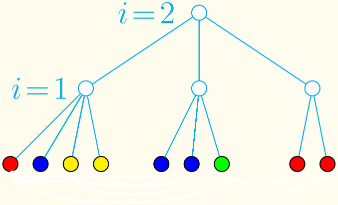


- a rooted tree T of height d,
- leaves are the **vertices of G**,

  each leaf has one of m colours,
- an associated *signature* determining the **edge set of G** as follows:

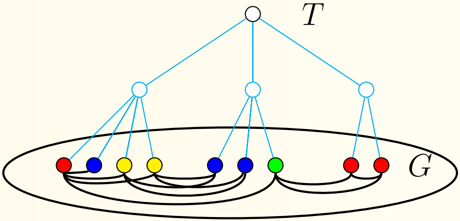
  for $i = 1, 2, \dots, d$, let u and v be leaves with the least common ancestor at height i in T,

  then $uv \in E(G)$ iff the colour pair of u, v is in the signature at height i.



2 Tree-models and Shrub-depth

Tree-model of m colours and depth d:



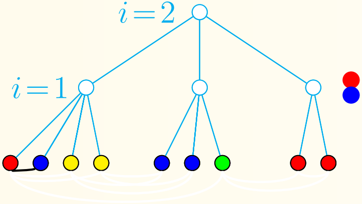
- a rooted tree T of height d,
- leaves are the **vertices of G**,

  each leaf has one of m colours,

- an associated *signature* determining the **edge set of G** as follows:

  for $i = 1, 2, \ldots, d$, let u and v be leaves with the least common ancestor at height i in T,

  then $uv \in E(G)$ iff the colour pair of u, v is in the signature at height i.

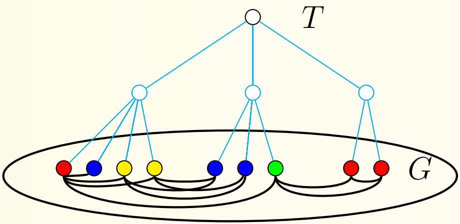


2 Tree-models and Shrub-depth

Tree-model of m colours and depth d:

- a rooted tree T of height d,
- leaves are the **vertices of G**,

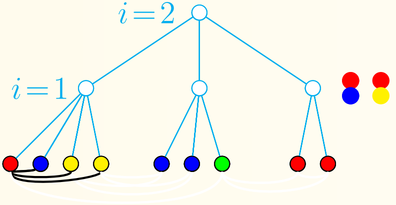
  each leaf has one of m colours,



- an associated *signature* determining the **edge set of G** as follows:

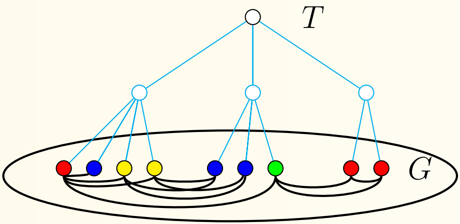
  for $i = 1, 2, \ldots, d$, let u and v be leaves with the least common ancestor at height i in T,

  then $uv \in E(G)$ iff the colour pair of u, v is in the signature at height i.



2 Tree-models and Shrub-depth

Tree-model of m colours and depth d:

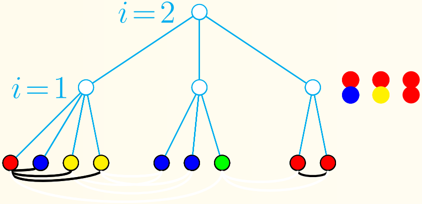


- a rooted tree T of height d,
- leaves are the **vertices of G**,

  each leaf has one of m colours,
- an associated *signature* determining the **edge set of G** as follows:

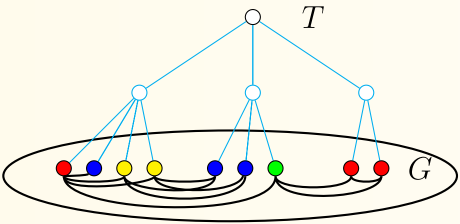
  for $i = 1, 2, \ldots, d$, let u and v be leaves with the least common ancestor at height i in T,

  then $uv \in E(G)$ iff the colour pair of u, v is in the signature at height i.



2 Tree-models and Shrub-depth

Tree-model of m colours and depth d:

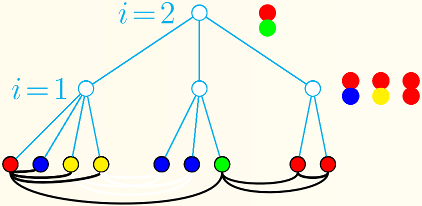


- a rooted tree T of height d,
- leaves are the **vertices of G**,

  each leaf has one of m colours,
- an associated *signature* determining the **edge set of G** as follows:

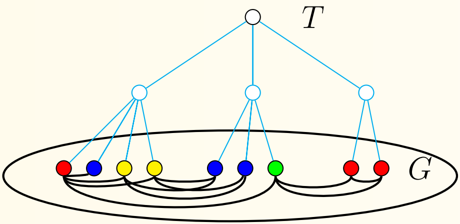
  for $i = 1, 2, \ldots, d$, let u and v be leaves with the least common ancestor at height i in T,

  then $uv \in E(G)$ iff the colour pair of u, v is in the signature at height i.



2 Tree-models and Shrub-depth

Tree-model of m colours and depth d:

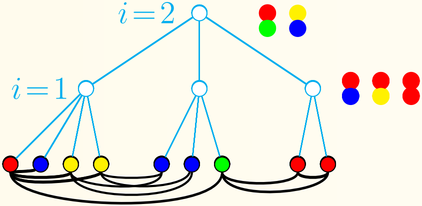


- a rooted tree T of height d,
- leaves are the **vertices of G**,

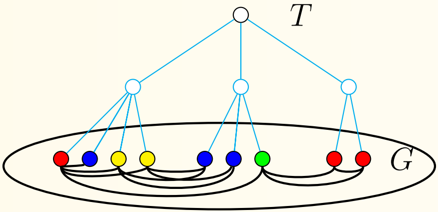
  each leaf has one of m colours,
- an associated *signature* determining the **edge set of G** as follows:

  for $i = 1, 2, \ldots, d$, let u and v be leaves with the least common ancestor at height i in T,

  then $uv \in E(G)$ iff the colour pair of u, v is in the signature at height i.



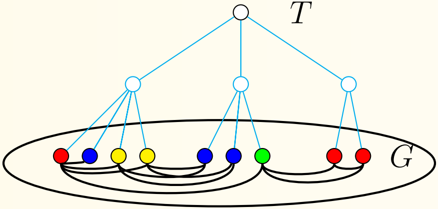
Shrub-depth



Class $\mathcal{TM}_m(d)$ $=$

$\{$ graphs with a tree-model of m *colours and depth* d $\}$

Shrub-depth

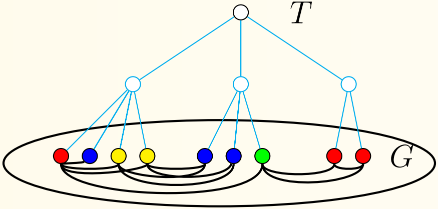


Class $\mathcal{TM}_m(d)$ =

{ graphs with a tree-model of m *colours and depth* d }

– closed under complements and induced subgraphs,

Shrub-depth



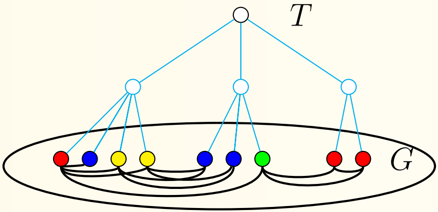
Class $\mathcal{TM}_m(d)$ =

{ graphs with a tree-model of m *colours and depth* d }

– closed under complements and induced subgraphs,

– but neither under disjoint unions nor under subgraphs.

Shrub-depth



Class $\mathcal{TM}_m(d)$ $=$

$\{$ graphs with a tree-model of m *colours and depth* d $\}$

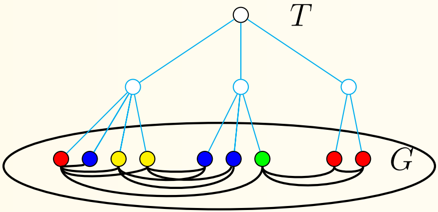
- closed under complements and induced subgraphs,
- but neither under disjoint unions nor under subgraphs.

Definition

A graph class $\mathcal{G}$ is of **shrub-depth d** iff

there exists m such that $\mathcal{G} \subseteq \mathcal{TM}_m(d)$ (same m for all $\mathcal{G}$!),

Shrub-depth



Class $\mathcal{TM}_m(d)$ $=$

{ graphs with a tree-model of m *colours and depth* d }

- closed under complements and induced subgraphs,
- but neither under disjoint unions nor under subgraphs.

Definition

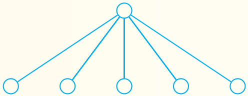
A graph class $\mathcal{G}$ is of **shrub-depth d** iff

there exists m such that $\mathcal{G} \subseteq \mathcal{TM}_m(d)$ (same m for all $\mathcal{G}$!),

while for all m' we have $\mathcal{G} \not\subseteq \mathcal{TM}_{m'}(d-1)$.

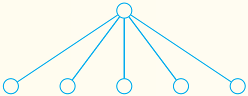
Examples and simple properties

- Shrub-depth 1: e.g., cliques, stars, ...
  gen. BND – bounded neighbourhood diversity.



Examples and simple properties

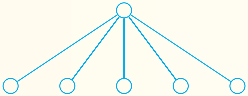
- Shrub-depth $\mathbf{1}$: e.g., cliques, stars, ...
  gen. BND – bounded neighbourhood diversity.



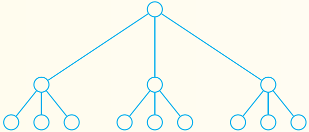
- Shrub-depth $\mathbf{2}$: e.g., matchings, union of cliques,

Examples and simple properties

- Shrub-depth $\mathbf{1}$: e.g., cliques, stars, ...
  gen. BND – bounded neighbourhood diversity.

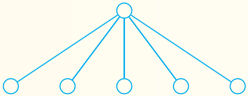


- Shrub-depth $\mathbf{2}$: e.g., matchings, union of cliques,
  generally – take a BND graph, and expand each vertex by any (other) BND graph.

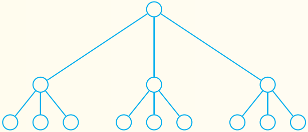


Examples and simple properties

- Shrub-depth $\mathbf{1}$: e.g., cliques, stars, ...
  gen. BND – bounded neighbourhood diversity.



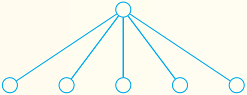
- Shrub-depth $\mathbf{2}$: e.g., matchings, union of cliques,
  generally – take a BND graph, and expand each vertex by any (other) BND graph.



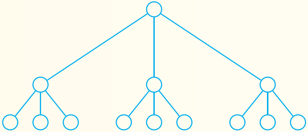
- The previous can be generalized to d levels...

Examples and simple properties

- Shrub-depth $\mathbf{1}$: e.g., cliques, stars, ...
  gen. BND – bounded neighbourhood diversity.



- Shrub-depth $\mathbf{2}$: e.g., matchings, union of cliques,
  generally – take a BND graph, and expand
  each vertex by any (other) BND graph.

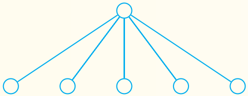


- The previous can be generalized to d levels...

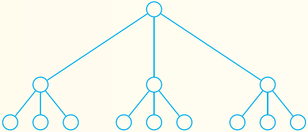
- Bounded tree-depth $\Rightarrow$ bounded shrub-depth.

Examples and simple properties

- Shrub-depth $\mathbf{1}$: e.g., cliques, stars, ...
  gen. BND – bounded neighbourhood diversity.



- Shrub-depth $\mathbf{2}$: e.g., matchings, union of cliques,
  generally – take a BND graph, and expand each vertex by any (other) BND graph.



- The previous can be generalized to d levels...

- Bounded tree-depth $\Rightarrow$ bounded shrub-depth.

- Bounded shrub-depth $\Rightarrow$ bounded linear clique-width.

Alternative: SC-depth

Goal: to get a depth parameter valid for a single graph. . .

Alternative: SC-depth

Goal: to get a depth parameter valid for a single graph...

SC-classes ("subset-complementation"):

- $\boldsymbol{\mathcal{SC}}(\mathbf{0}) = \{K_1\}$,

Alternative: SC-depth

Goal: to get a depth parameter valid for a single graph. . .

SC-classes ("subset-complementation"):

- $\boldsymbol{\mathcal{SC}}(\mathbf{0}) = \{K_1\}$,
- having $G_1, G_2, \ldots, G_p \in \boldsymbol{\mathcal{SC}}(\boldsymbol{k})$,

  take the disjoint union $H := G_1 \cup G_2 \cup \ldots \cup G_p$ and,

Alternative: SC-depth

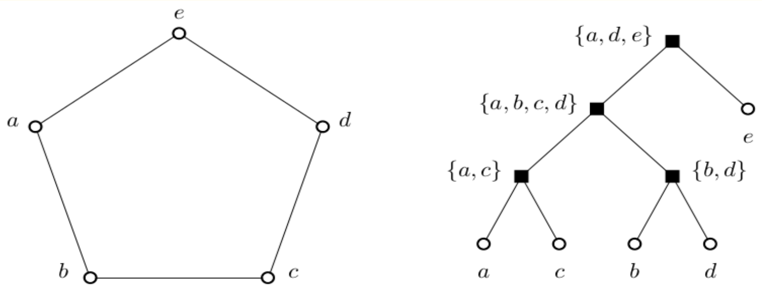
Goal: to get a depth parameter valid for a single graph...

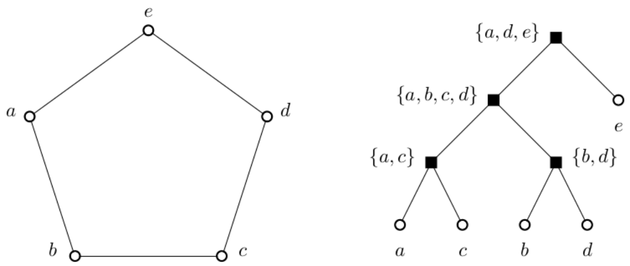
SC-classes ("subset-complementation"):

- $\mathcal{SC}(0) = \{K_1\}$,
- having $G_1, G_2, \ldots, G_p \in \mathcal{SC}(k)$,

  take the disjoint union $H := G_1 \cup G_2 \cup \ldots \cup G_p$ and,
  for every $X \subseteq V(H)$,

  H with complemented edges on $X \hookrightarrow \mathcal{SC}(k+1)$.

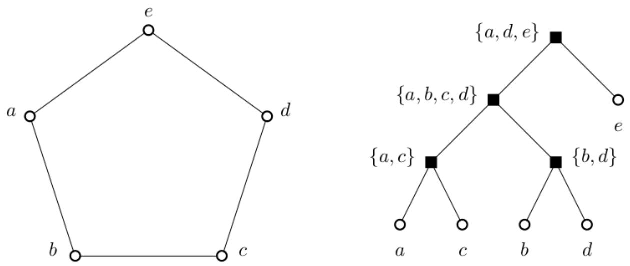




Definition

A graph G is of **SC-depth k** iff

$$G \in \mathcal{SC}(k) \text{ but } G \notin \mathcal{SC}(k-1).$$



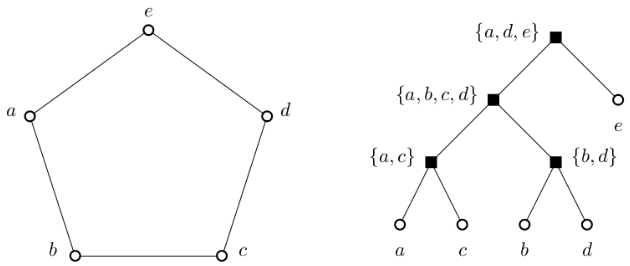
Definition

A graph G is of **SC-depth k** iff

$G \in \mathcal{SC}(k)$ but $G \notin \mathcal{SC}(k-1)$.

Theorem For a graph class $\mathcal{G}$;

bounded shrub-depth $\iff$ bounded SC-depth.



Definition

A graph G is of **SC-depth** k iff

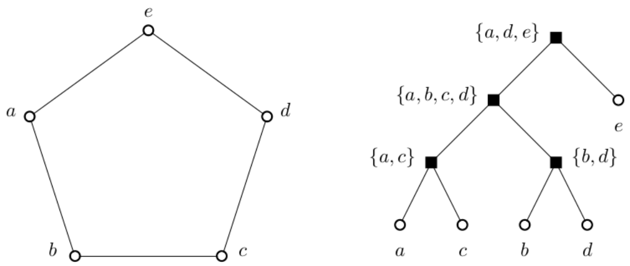
$G \in \mathcal{SC}(k)$ but $G \notin \mathcal{SC}(k-1)$.

Theorem For a graph class $\mathcal{G}$;

bounded shrub-depth $\iff$ bounded SC-depth.

A proof sketch:

– $\mathcal{SC}(k) \subseteq \mathcal{TM}_{2^k}(k)$, and



Definition

A graph G is of **SC-depth k** iff

$G \in \mathcal{SC}(k)$ but $G \notin \mathcal{SC}(k-1)$.

Theorem For a graph class $\mathcal{G}$;

bounded shrub-depth $\iff$ bounded SC-depth.

A proof sketch:

– $\mathcal{SC}(k) \subseteq \mathcal{TM}_{2^k}(k)$, and
– $\mathcal{TM}_m(d) \subseteq \mathcal{SC}(dm(m+1))$.

Alternative II: Rank-depth

Warning: just a very brief sketch (assuming you know branch-width).

Alternative II: Rank-depth

Warning: just a very brief sketch (assuming you know branch-width).

- Branch-width of a submodular function

  $\leadsto$ rank-width of a graph:

  consider the *cut-rank* function on the vertex set of a graph.

Alternative II: Rank-depth

Warning: just a very brief sketch (assuming you know branch-width).

- Branch-width of a submodular function

  $\rightsquigarrow$ rank-width of a graph:

  consider the *cut-rank* function on the vertex set of a graph.

- [Geelen] Branch-width $\rightsquigarrow$ branch-depth:

Alternative II: Rank-depth

Warning: just a very brief sketch (assuming you know branch-width).

- Branch-width of a submodular function

  $\rightsquigarrow$ rank-width of a graph:

  consider the *cut-rank* function on the vertex set of a graph.

- [Geelen] Branch-width $\rightsquigarrow$ branch-depth:

  allowing high-degree nodes, and

  node width measured over all bipartitions of incident subtrees.

Alternative II: Rank-depth

Warning: just a very brief sketch (assuming you know branch-width).

- Branch-width of a submodular function

  $\rightsquigarrow$ rank-width of a graph:

  consider the *cut-rank* function on the vertex set of a graph.

- [Geelen] Branch-width $\rightsquigarrow$ branch-depth:

  allowing high-degree nodes, and

  node width measured over all bipartitions of incident subtrees.

- Branch-depth of a submodular function

  $\rightsquigarrow$ rank-depth of a graph.

Alternative II: Rank-depth

Warning: just a very brief sketch (assuming you know branch-width).

- Branch-width of a submodular function

  $\rightsquigarrow$ rank-width of a graph:

  consider the *cut-rank* function on the vertex set of a graph.

- [Geelen] Branch-width $\rightsquigarrow$ branch-depth:

  allowing high-degree nodes, and

  node width measured over all bipartitions of incident subtrees.

- Branch-depth of a submodular function

  $\rightsquigarrow$ rank-depth of a graph.

Theorem [DeVos, Kwon, Oum] For a graph class $\mathcal{G}$;

bounded shrub-depth $\iff$ bounded rank-depth.

3 Further Combinatorial Properties

Theorem Each of $\mathcal{TM}_m(d)$ and $\mathcal{SC}(k)$ have

finite sets of forbidden induced subgraphs (obstructions).

3 Further Combinatorial Properties

Theorem Each of $\mathcal{TM}_m(d)$ and $\mathcal{SC}(k)$ have

finite sets of forbidden induced subgraphs (obstructions).

Proof sketch:

– e.g., min. obstructions for $\mathcal{TM}_m(d)$ contained in $\mathcal{TM}_{2m+1}(d)$;

3 Further Combinatorial Properties

Theorem Each of $\mathcal{TM}_m(d)$ and $\mathcal{SC}(k)$ have

finite sets of forbidden induced subgraphs (obstructions).

Proof sketch:

– e.g., min. obstructions for $\mathcal{TM}_m(d)$ contained in $\mathcal{TM}_{2m+1}(d)$;

– plus [Ding] WQO of coloured rooted trees of bounded height (simply iterate classical Higman's lemma...).

3 Further Combinatorial Properties

Theorem Each of $\mathcal{TM}_m(d)$ and $\mathcal{SC}(k)$ have

finite sets of forbidden induced subgraphs (obstructions).

Proof sketch:

- e.g., min. obstructions for $\mathcal{TM}_m(d)$ contained in $\mathcal{TM}_{2m+1}(d)$;
- plus [Ding] WQO of coloured rooted trees of bounded height (simply iterate classical Higman's lemma...).

Theorem (shrub-depth "from" tree-depth)

A class $\mathcal{G}$ of bounded shrub-depth $\Rightarrow$ exists d such that

each graph of $\mathcal{G}$ is a vertex-minor of a graph of tree-depth d.

3 Further Combinatorial Properties

Theorem Each of $\mathcal{TM}_m(d)$ and $\mathcal{SC}(k)$ have

finite sets of forbidden induced subgraphs (obstructions).

Proof sketch:

– e.g., min. obstructions for $\mathcal{TM}_m(d)$ contained in $\mathcal{TM}_{2m+1}(d)$;

– plus [Ding] WQO of coloured rooted trees of bounded height (simply iterate classical Higman's lemma. . .).

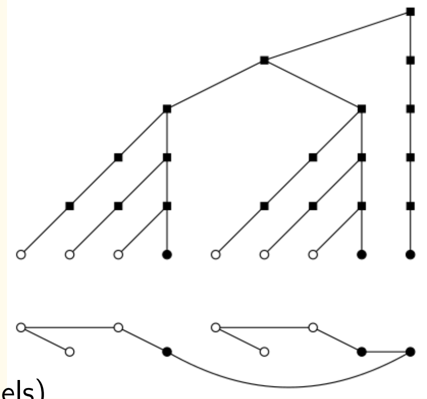
Theorem (shrub-depth "from" tree-depth)

A class $\mathcal{G}$ of bounded shrub-depth $\Rightarrow$ exists d such that

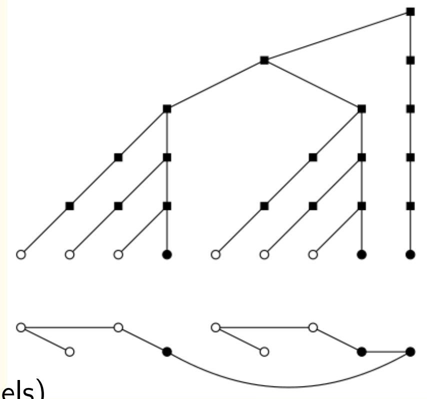
each graph of $\mathcal{G}$ is a vertex-minor of a graph of tree-depth d.

Proof sketch:

– start from an SC-depth tree, and "simulate" subset complem. via extra vert. with local complem.



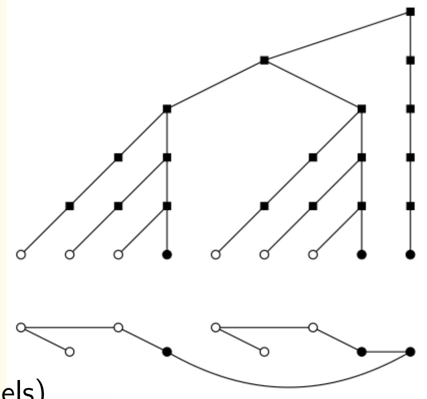
Theorem ("no long paths" in tree-models)



Theorem ("no long paths" in tree-models)

Let $P_\ell :=$ path of length ℓ, where $\ell = 3 \cdot 2^m - 4$. Then

$$P_\ell \in \mathcal{TM}_m(2m+1) \text{ but } \forall d : P_{\ell+1} \notin \mathcal{TM}_m(d).$$



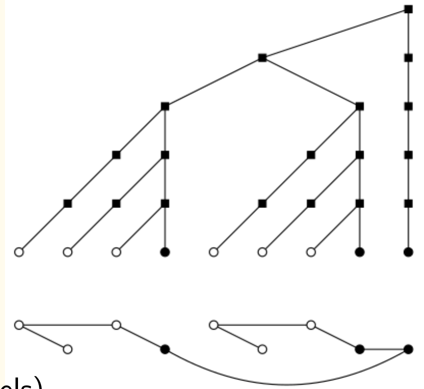
Theorem ("no long paths" in tree-models)

Let $P_\ell :=$ path of length ℓ, where $\ell = 3 \cdot 2^m - 4$. Then

$$P_\ell \in \mathcal{TM}_m(2m+1) \text{ but } \forall d : P_{\ell+1} \notin \mathcal{TM}_m(d).$$

Proof sketch:

– the tight bound comes from a delicate induction (skipped),



Theorem ("no long paths" in tree-models)

Let $P_\ell :=$ path of length ℓ, where $\ell = 3 \cdot 2^m - 4$. Then

$$P_\ell \in \mathcal{TM}_m(2m+1) \text{ but } \forall d : P_{\ell+1} \notin \mathcal{TM}_m(d).$$

Proof sketch:

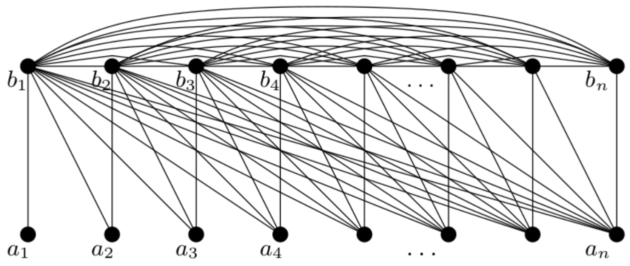
- the tight bound comes from a delicate induction (skipped),
- easy weaker argument: every large tree-model of bounded d, m has triplicate subtrees $\rightarrow$ cannot represent a path.

Observation Forbidding induced paths is not enough!

(to bound shrub-depth/SC-depth)

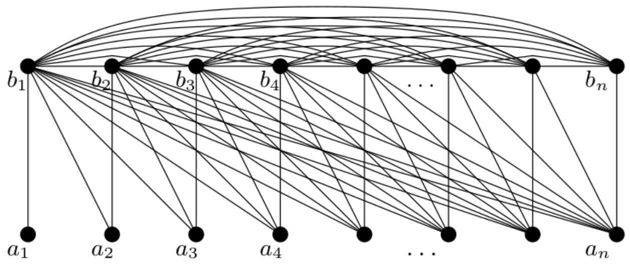
Observation Forbidding induced paths is not enough!

(to bound shrub-depth/SC-depth)



Observation Forbidding induced paths is not enough!

(to bound shrub-depth/SC-depth)

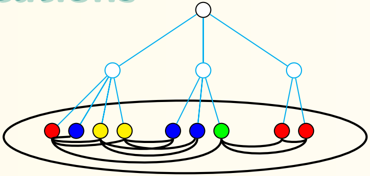


Conjecture A class $\mathcal{G}$ is of bounded shrub-depth

$$\iff$$

there exists t such that no graph of $\mathcal{G}$ contains P_t as a vertex minor.

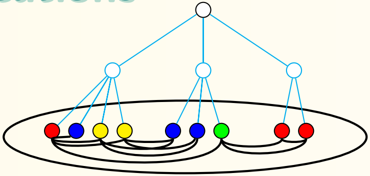
4 Shrub-depth and Interpretations



Observation (tree-model $\rightarrow$ interpretation)

Graphs have FO interpretations in their tree-models from $\mathcal{TM}_m(d)$.

4 Shrub-depth and Interpretations



Observation (tree-model $\rightarrow$ interpretation)

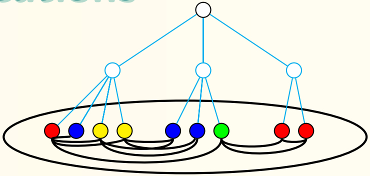
Graphs have FO interpretations in their tree-models from $\mathcal{TM}_m(d)$.

Theorem (interpretation $\rightarrow$ tree-model)

A graph class $\mathcal{G}$ has a simple CMSO_1 interpretation
in a class $\mathcal{T}_d$ of coloured rooted trees of height $\leq d$

$\Rightarrow$ $\mathcal{G}$ is of shrub-depth $\leq d$.

4 Shrub-depth and Interpretations



Observation (tree-model $\rightarrow$ interpretation)

Graphs have FO interpretations in their tree-models from $\mathcal{TM}_m(d)$.

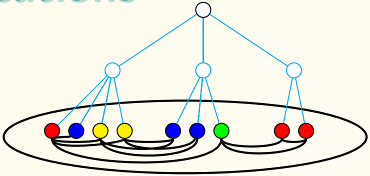
Theorem (interpretation $\rightarrow$ tree-model)

A graph class $\mathcal{G}$ has a simple CMSO$_1$ interpretation
in a class $\mathcal{T}_d$ of coloured rooted trees of height $\leq d$
$\Rightarrow$ $\mathcal{G}$ is of shrub-depth $\leq d$.

Some consequences

- Shrub-depth preserved under vertex-minors (exactly);
  using [Courcelle–Oum].

4 Shrub-depth and Interpretations



Observation (tree-model $\to$ interpretation)

Graphs have FO interpretations in their tree-models from $\mathcal{TM}_m(d)$.

Theorem (interpretation $\to$ tree-model)

A graph class $\mathcal{G}$ has a simple CMSO_1 interpretation
in a class $\mathcal{T}_d$ of coloured rooted trees of height $\leq d$

$\Rightarrow$ $\mathcal{G}$ is of shrub-depth $\leq d$.

Some consequences

- Shrub-depth preserved under vertex-minors (exactly);
  using [Courcelle–Oum].
- The finite levels of the MSO_1 transduction hierarchy are given (almost) by shrub-depth $1, 2, 3 \ldots$; cf. [Blumensath–Courcelle].

Proof sketch

- CMSO_1 interpretation $I(\mathcal{T}_d) = \mathcal{G}$, given by $I = (\alpha, \beta)$:
  - vertex set $V(G) := \{u : T \models \alpha(u)\}$ – can be ignored,

Proof sketch

- CMSO_1 interpretation $I(\mathcal{T}_d) = \mathcal{G}$, given by $I = (\alpha, \beta)$:
  - vertex set $V(G) := \{u : T \models \alpha(u)\}$ – can be ignored,
  - edge set $E(G) := \{uv : T \models \beta(u, v)\}$ – crucial.

Proof sketch

- CMSO$_1$ interpretation $I(\mathcal{T}_d) = \mathcal{G}$, given by $I = (\alpha, \beta)$:
  - vertex set $V(G) := \{u : T \models \alpha(u)\}$ – can be ignored,
  - edge set $E(G) := \{uv : T \models \beta(u, v)\}$ – crucial.
- Avoid free vars.: $\beta' \equiv \exists x, y[L(x) \wedge L(y) \wedge \beta(x, y)]$, and give the new label L to u, v, as needed,

Proof sketch

- CMSO$_1$ interpretation $I(\mathcal{T}_d) = \mathcal{G}$, given by $I = (\alpha, \beta)$:
  - vertex set $V(G) := \{u : T \models \alpha(u)\}$ – can be ignored,
  - edge set $E(G) := \{uv : T \models \beta(u, v)\}$ – crucial.
- Avoid free vars.: $\beta' \equiv \exists x, y[L(x) \wedge L(y) \wedge \beta(x, y)]$,
  and give the new label L to u, v, as needed,

$$T \models \beta(u, v) \iff T[L(u), L(v)] \models \beta'.$$

Proof sketch

- CMSO_1 interpretation $I(\mathcal{T}_d) = \mathcal{G}$, given by $I = (\alpha, \beta)$:
  - vertex set $V(G) := \{u : T \models \alpha(u)\}$ – can be ignored,
  - edge set $E(G) := \{uv : T \models \beta(u,v)\}$ – crucial.
- Avoid free vars.: $\beta' \equiv \exists x, y[L(x) \wedge L(y) \wedge \beta(x,y)]$,

  and give the new label L to u, v, as needed,

$$T \models \beta(u,v) \iff T[L(u), L(v)] \models \beta'.$$

- Now, ... * miracle happens *..., reducing every

$$T[L(u), L(v)] \rightsquigarrow \text{ bounded } T_0[L(u), L(v)] \text{ such that}$$

Proof sketch

- CMSO$_1$ interpretation $I(\mathcal{T}_d) = \mathcal{G}$, given by $I = (\alpha, \beta)$:
  - vertex set $V(G) := \{u : T \models \alpha(u)\}$ – can be ignored,
  - edge set $E(G) := \{uv : T \models \beta(u, v)\}$ – crucial.
- Avoid free vars.: $\beta' \equiv \exists x, y[L(x) \wedge L(y) \wedge \beta(x, y)]$,

  and give the new label L to u, v, as needed,

$$T \models \beta(u, v) \iff T[L(u), L(v)] \models \beta'.$$

- Now, . . . * miracle happens *. . . , reducing every

  $T[L(u), L(v)] \rightsquigarrow$ bounded $T_0[L(u), L(v)]$ such that

$$T[L(u), L(v)] \models \beta' \iff T_0[L(u), L(v)] \models \beta'.$$

Proof sketch

- CMSO_1 interpretation $I(\mathcal{T}_d) = \mathcal{G}$, given by $I = (\alpha, \beta)$:
  - vertex set $V(G) := \{u : T \models \alpha(u)\}$ – can be ignored,
  - edge set $E(G) := \{uv : T \models \beta(u,v)\}$ – crucial.
- Avoid free vars.: $\beta' \equiv \exists x, y[L(x) \wedge L(y) \wedge \beta(x,y)]$,

  and give the new label L to u, v, as needed,

$$T \models \beta(u,v) \iff T[L(u), L(v)] \models \beta'.$$

- Now, . . . * miracle happens *. . . , reducing every

$$T[L(u), L(v)] \rightsquigarrow \text{ bounded } T_0[L(u), L(v)] \text{ such that}$$

$$T[L(u), L(v)] \models \beta' \iff T_0[L(u), L(v)] \models \beta'.$$

  Moreover, doing this carefully, there is such universal $T \rightsquigarrow T_0$ to which $L(u'), L(v')$ can be added afterwards!

- CMSO$_1$ interpretation $I(\mathcal{T}_d) = \mathcal{G}$, given by $I = (\alpha, \beta)$:
  - vertex set $V(G) := \{u : T \models \alpha(u)\}$ – can be ignored,
  - edge set $E(G) := \{uv : T \models \beta(u, v)\}$ – crucial.

. . . .

- CMSO$_1$ interpretation $I(\mathcal{T}_d) = \mathcal{G}$, given by $I = (\alpha, \beta)$:
  - vertex set $V(G) := \{u : T \models \alpha(u)\}$ – can be ignored,
  - edge set $E(G) := \{uv : T \models \beta(u, v)\}$ – crucial.

. . . .

- Recall univ. $T \rightsquigarrow T_0$, and choose $L(u), L(v)$ to "query" $uv \in^? E(G)$
  - this T_0 is bounded $\rightarrow$ part of the signature, but

- CMSO$_1$ interpretation $I(\mathcal{T}_d) = \mathcal{G}$, given by $I = (\alpha, \beta)$:
  - vertex set $V(G) := \{u : T \models \alpha(u)\}$ – can be ignored,
  - edge set $E(G) := \{uv : T \models \beta(u,v)\}$ – crucial.

. . . .

- Recall univ. $T \rightsquigarrow T_0$, and choose $L(u), L(v)$ to "query" $uv \in^? E(G)$
  - this T_0 is bounded $\rightarrow$ part of the signature, but
  - how to get appropriate $T_0[L(u'), L(v')]$ then?

- CMSO$_1$ interpretation $I(\mathcal{T}_d) = \mathcal{G}$, given by $I = (\alpha, \beta)$:
  - vertex set $V(G) := \{u : T \models \alpha(u)\}$ – can be ignored,
  - edge set $E(G) := \{uv : T \models \beta(u, v)\}$ – crucial.

. . . .

- Recall univ. $T \rightsquigarrow T_0$, and choose $L(u), L(v)$ to "query" $uv \in^? E(G)$
  - this T_0 is bounded $\rightarrow$ part of the signature, but
  - how to get appropriate $T_0[L(u'), L(v')]$ then?
- The final step:

  each $u \in V(T) \;\mapsto\;$ orbit $O(u')$ of $Aut(T_0)$,

- CMSO$_1$ interpretation $I(\mathcal{T}_d) = \mathcal{G}$, given by $I = (\alpha, \beta)$:
  - vertex set $V(G) := \{u : T \models \alpha(u)\}$ – can be ignored,
  - edge set $E(G) := \{uv : T \models \beta(u, v)\}$ – crucial.

. . . .

- Recall univ. $T \rightsquigarrow T_0$, and choose $L(u), L(v)$ to "query" $uv \in^? E(G)$
  - this T_0 is bounded $\rightarrow$ part of the signature, but
  - how to get appropriate $T_0[L(u'), L(v')]$ then?
- The final step:

$$\text{each } u \in V(T) \;\mapsto\; \text{orbit } O(u') \text{ of } Aut(T_0),$$

and we can assign $O(u')$ as the colour of u in our tree-model. □

Concluding Remarks

- Asymptotic shrub-depth *or* concrete SC-depth (rank-depth)?

Concluding Remarks

- Asymptotic shrub-depth *or* concrete SC-depth (rank-depth)?

  – actually, choose any one you want (always good to have options).

Concluding Remarks

- Asymptotic shrub-depth *or* concrete SC-depth (rank-depth)?

  – actually, choose any one you want (always good to have options).

- Generalize shrub-depth to relational structures? Straightforward…

Concluding Remarks

- Asymptotic shrub-depth *or* concrete SC-depth (rank-depth)?

  – actually, choose any one you want (always good to have options).

- Generalize shrub-depth to relational structures? Straightforward...

- Characterise (combinatorially) classes of unbounded shrub-depth?

  (Recall long paths as vertex minors...)

Concluding Remarks

- Asymptotic shrub-depth *or* concrete SC-depth (rank-depth)?

  – actually, choose any one you want (always good to have options).

- Generalize shrub-depth to relational structures? Straightforward...

- Characterise (combinatorially) classes of unbounded shrub-depth?

  (Recall long paths as vertex minors...)

- Get a better grip on shrub-depth (similarly to tree-depth)?

Concluding Remarks

- Asymptotic shrub-depth *or* concrete SC-depth (rank-depth)?

  – actually, choose any one you want (always good to have options).

- Generalize shrub-depth to relational structures? Straightforward...

- Characterise (combinatorially) classes of unbounded shrub-depth?

  (Recall long paths as vertex minors...)

- Get a better grip on shrub-depth (similarly to tree-depth)?

Thank you for your attention.