Lacon- and Shrub-Decompositions

A New Characterization of First-Order Transductions of Bounded Expansion Classes

---

Jan Dreier
TU Wien, Austria

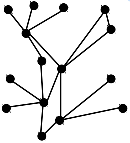
LICS 2021
May 19, 2021

Algorithmic Meta-Theorems

“All problems expressible in a certain logic can be solved efficiently on certain graphs”

"All problems expressible in a certain logic can be solved efficiently on certain graphs"

MSO on treewidth

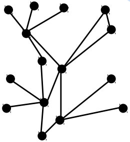


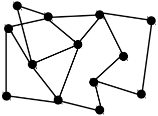
Algorithmic Meta-Theorems

"All problems expressible in a certain logic can be solved efficiently on certain graphs"

MSO on treewidth

FO on sparse graphs



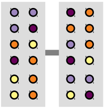


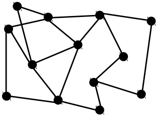
Algorithmic Meta-Theorems

"All problems expressible in a certain logic can be solved efficiently on certain graphs"

MSO on cliquewidth

FO on sparse graphs



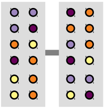


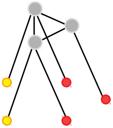
Algorithmic Meta-Theorems

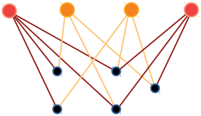
"All problems expressible in a certain logic can be solved efficiently on certain graphs"

MSO on cliquewidth

?







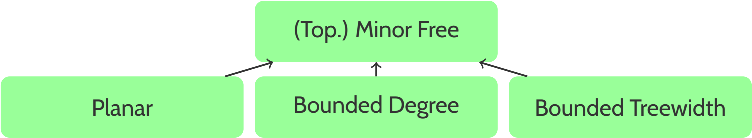
Some Sparse Graph Classes

Planar

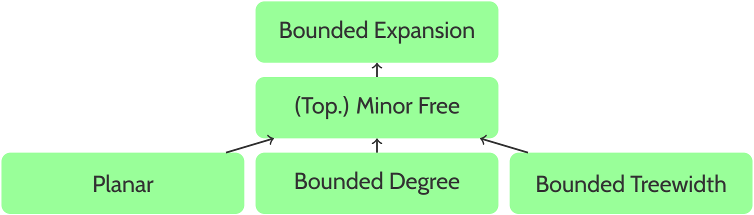
Bounded Degree

Bounded Treewidth

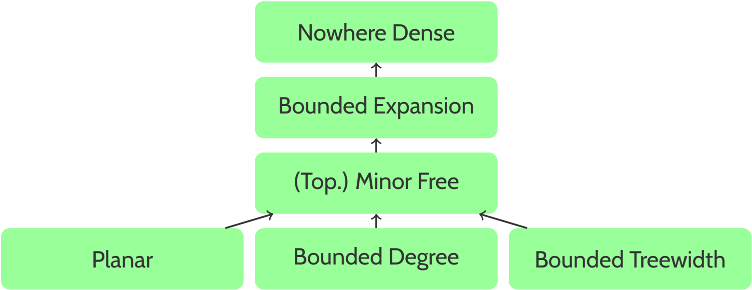
Some Sparse Graph Classes



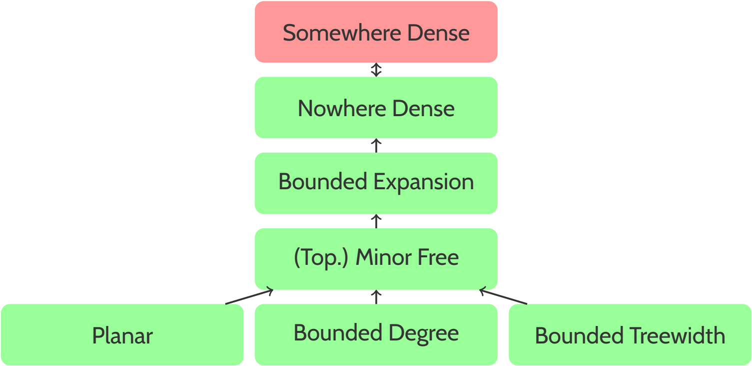
Some Sparse Graph Classes



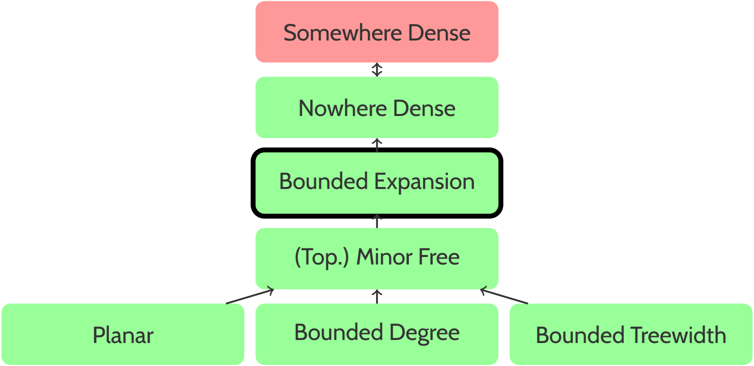
Some Sparse Graph Classes



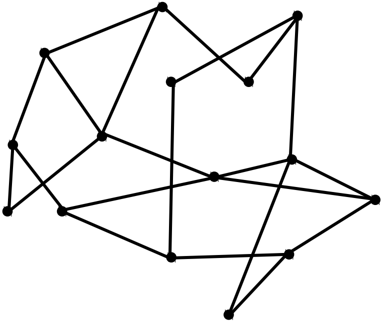
Some Sparse Graph Classes



Some Sparse Graph Classes



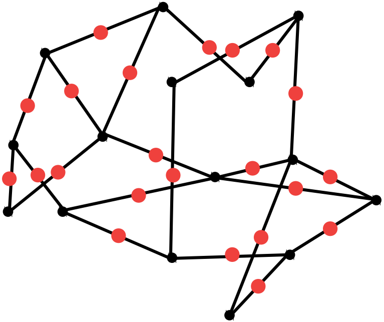
Bounded Expansion — Minors



$$\frac{|E|}{|V|} \leq c$$

for every graph
in the graph class.

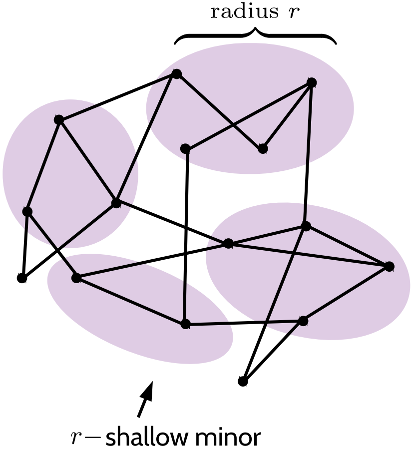
Bounded Expansion — Minors



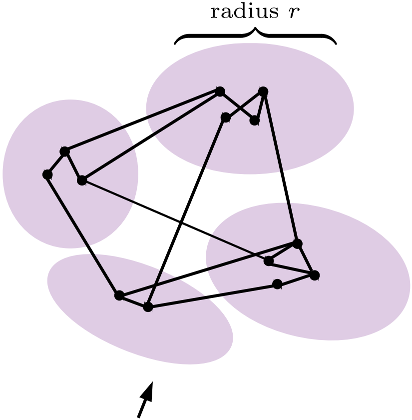
$$\frac{|E|}{|V|} \leq 2$$

for every graph
in the graph class.

Bounded Expansion — Minors

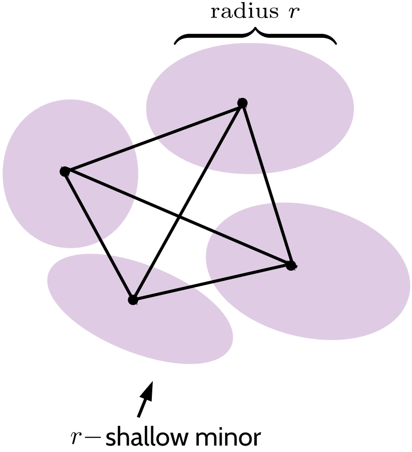


Bounded Expansion — Minors

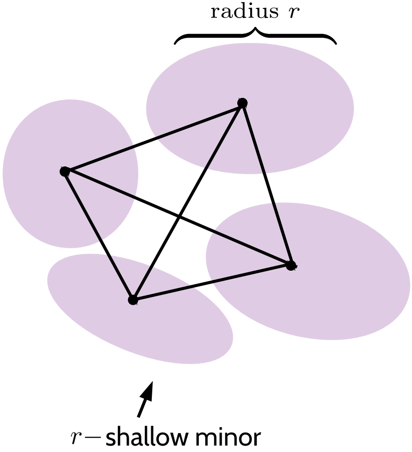


$r-$shallow minor

Bounded Expansion — Minors



Bounded Expansion — Minors



$$\frac{|E|}{|V|} \leq f(r)$$

for every r-shallow minor of every graph in the graph class.

Algorithmic Meta-Theorems

FO Model-Checking [Dvořák, Král, Thomas 2010]

First-order formulas φ can be evaluated on bounded expansion classes in time $f(|\varphi|)n$.

Algorithmic Meta-Theorems

FO Model-Checking [Dvořák, Král, Thomas 2010]

First-order formulas φ can be evaluated on bounded expansion classes in time $f(|\varphi|)n$.

- independent set of size k:

$$\exists x_1 \dots \exists x_k \bigwedge_{i,j} x_i \not\sim x_j \wedge x_i \neq x_j$$

FO Model-Checking [Dvořák, Král, Thomas 2010]

First-order formulas φ can be evaluated on bounded expansion classes in time $f(|\varphi|)n$.

- independent set of size k:

$$\exists x_1 \dots \exists x_k \bigwedge_{i,j} x_i \not\sim x_j \wedge x_i \neq x_j$$

- dominating set of size k:

$$\exists x_1 \dots \exists x_k \forall y \bigvee_{i} y \sim x_i \vee y = x_i$$

FO Model-Checking [Dvořák, Král, Thomas 2010]

First-order formulas φ can be evaluated on bounded expansion classes in time $f(|\varphi|)n$.

- independent set of size k:

$$\exists x_1 \ldots \exists x_k \bigwedge_{i,j} x_i \not\sim x_j \wedge x_i \neq x_j$$

- dominating set of size k:

$$\exists x_1 \ldots \exists x_k \forall y \bigvee_{i} y \sim x_i \vee y = x_i$$

Best algorithms on general graphs: $n^{O(k)}$

Algorithmic Meta-Theorems

FO Model-Checking [Dvořák, Král, Thomas 2010]

First-order formulas φ can be evaluated on bounded expansion classes in time $f(|\varphi|)n$.

- independent set of size k:

$$\exists x_1 \ldots \exists x_k \bigwedge_{i,j} x_i \not\sim x_j \wedge x_i \neq x_j$$

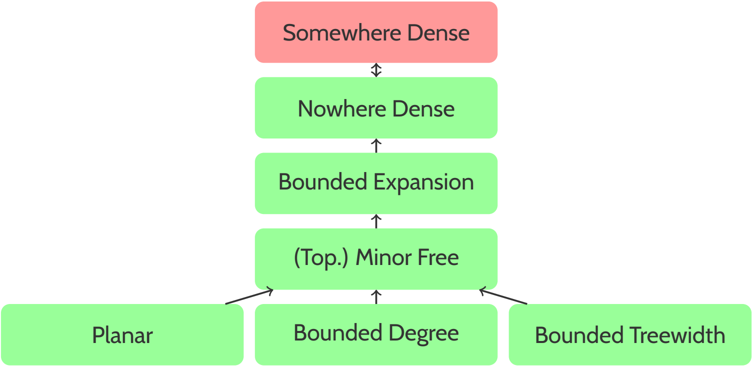
- dominating set of size k:

$$\exists x_1 \ldots \exists x_k \forall y \bigvee_{i} y \sim x_i \vee y = x_i$$

Best algorithms on general graphs: $n^{O(k)}$

On bounded expansion: $f(k)n$

Exact Characterization

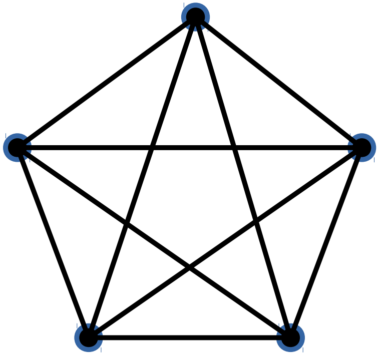


For graph classes $\mathcal{G}$ *closed under subgraphs*,
FO model-checking is tractable iff $\mathcal{G}$ is nowhere dense.

[Grohe, Kreutzer, Sieberz 2011]

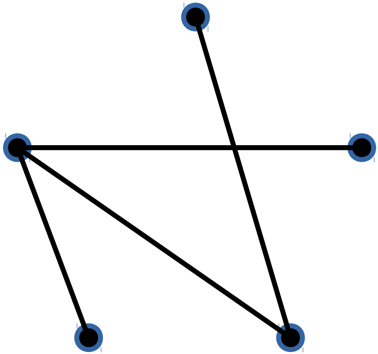
Dense Graphs

- What dense graph classes are tractable?



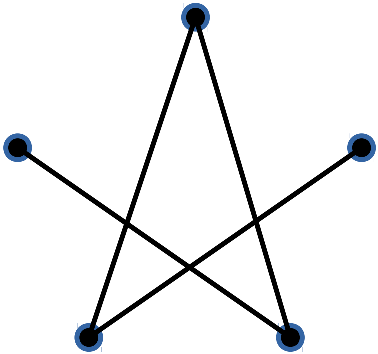
Dense Graphs

- What dense graph classes are tractable?
- Closure under subgraphs is not a good requirement.



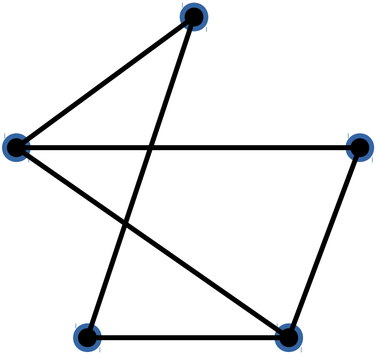
Dense Graphs

- What dense graph classes are tractable?
- Closure under subgraphs is not a good requirement.



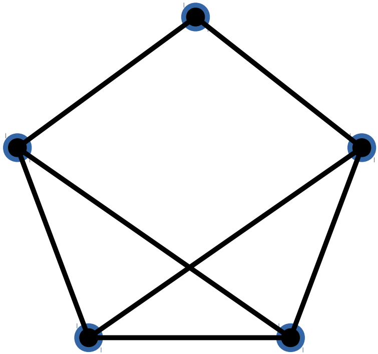
Dense Graphs

- What dense graph classes are tractable?
- Closure under subgraphs is not a good requirement.



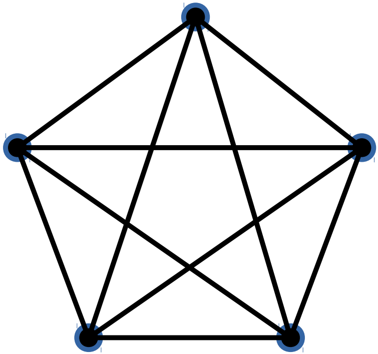
Dense Graphs

- What dense graph classes are tractable?
- Closure under subgraphs is not a good requirement.



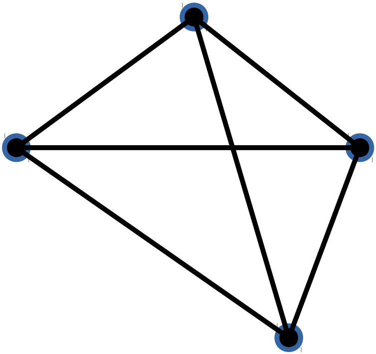
Dense Graphs

- What dense graph classes are tractable?
- Closure under subgraphs is not a good requirement.



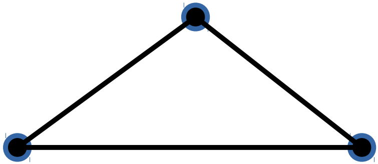
Dense Graphs

- What dense graph classes are tractable?
- Closure under subgraphs is not a good requirement.



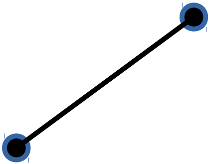
Dense Graphs

- What dense graph classes are tractable?
- Closure under subgraphs is not a good requirement.



Dense Graphs

- What dense graph classes are tractable?
- Closure under subgraphs is not a good requirement.



Dense Graphs

- What dense graph classes are tractable?
- Closure under subgraphs is not a good requirement.

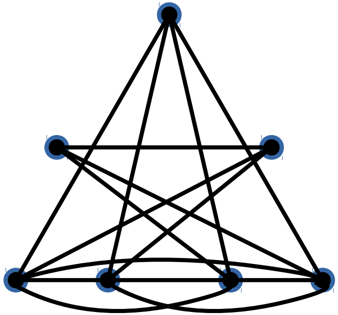
Dense Graphs

- What dense graph classes are tractable?
- Closure under subgraphs is not a good requirement.
- **Goal:**

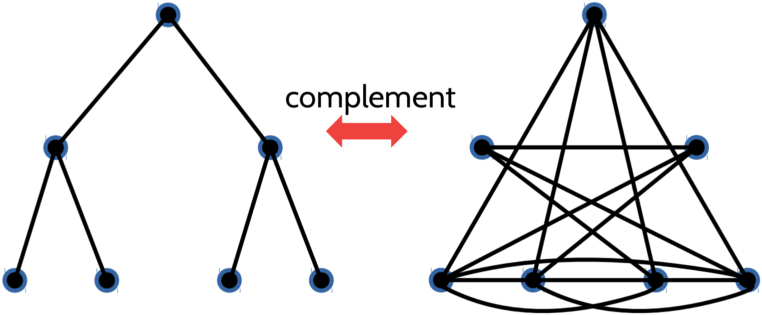
Theorem

For graph classes $\mathcal{G}$ *closed under induced subgraphs*,
FO model-checking is tractable iff [...].

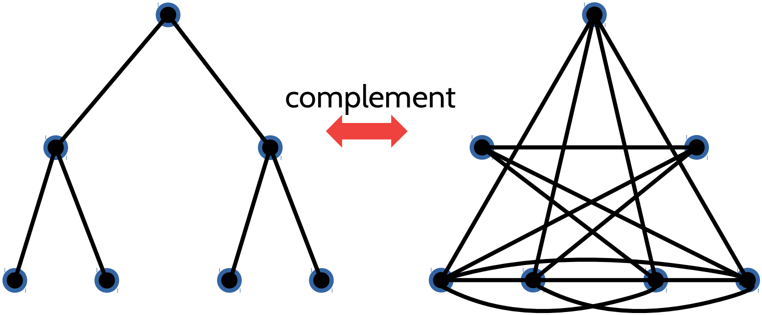
Example: Complements



Example: Complements



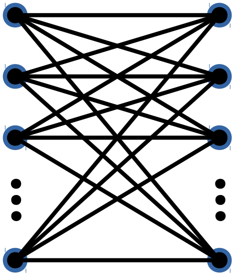
$x_1 \not\sim x_2$ $\qquad$ $x_1 \sim x_2$



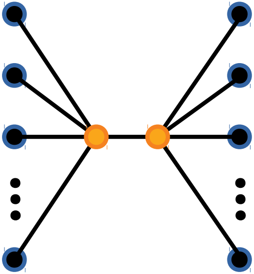
$$\exists x_1 \exists x_2 \exists x_3 \; x_1 \not\sim x_2 \wedge x_2 \not\sim x_3 \wedge x_1 \not\sim x_3$$

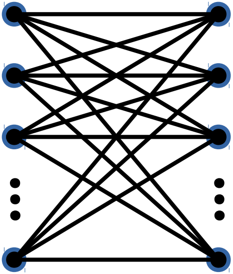
$$\exists x_1 \exists x_2 \exists x_3 \; x_1 \sim x_2 \wedge x_2 \sim x_3 \wedge x_1 \sim x_3$$

Example: Fully Bipartite

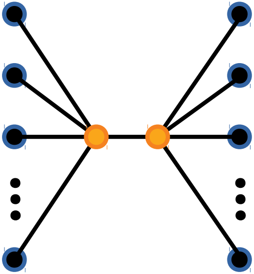


Example: Fully Bipartite

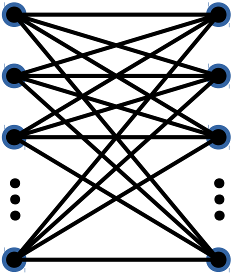




Example: Fully Bipartite

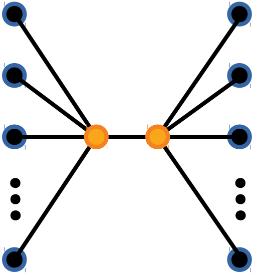


$$\mathrm{dist}(x, y) = 3$$



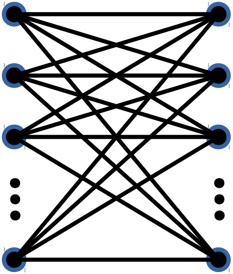
$$x \sim y$$

Example: Fully Bipartite



$$\mathrm{dist}(x, y) = 3$$

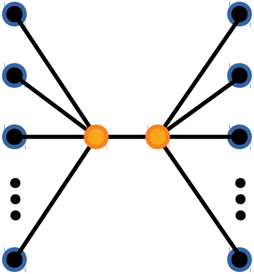
$$\exists x \ \mathrm{blue}(x) \wedge \varphi$$



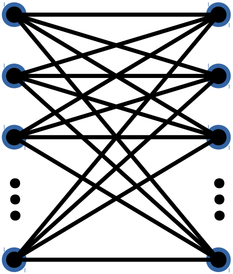
$$x \sim y$$

$$\exists x \ \varphi$$

Example: Fully Bipartite



$$\exists x_1 \exists x_2 \exists x_3$$
$$\mathrm{blue}(x_1) \wedge \mathrm{blue}(x_2) \wedge \mathrm{blue}(x_3)$$
$$\mathrm{dist}(x_1, x_2) = 3 \wedge \mathrm{dist}(x_2, x_3) = 3$$
$$\wedge \mathrm{dist}(x_1, x_3) = 3$$



$$\exists x_1 \exists x_2 \exists x_3$$
$$x_1 \sim x_2 \wedge x_2 \sim x_3$$
$$\wedge x_1 \sim x_3$$

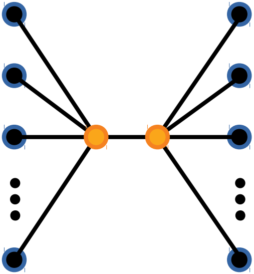
Interpretations

$$I = (\nu(x), \mu(x, y))$$

Interpretations

$$I = (\nu(x), \mu(x,y))$$

G $I(G)$

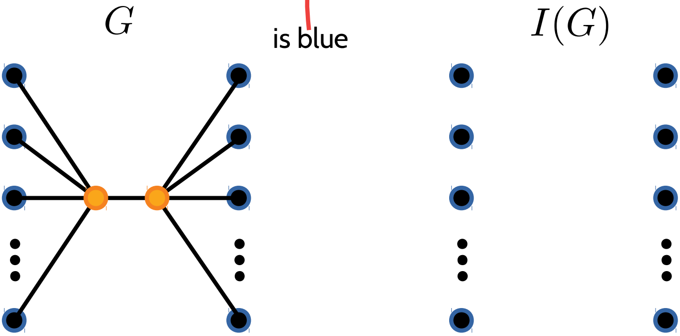


vertices: $\{v \mid G \models \nu(v)\}$

edges: $\{uv \mid G \models \mu(u,v)\}$

$$I = (\nu(x), \mu(x, y))$$

is blue



vertices: $\{v \mid G \models \nu(v)\}$

edges: $\{uv \mid G \models \mu(u, v)\}$

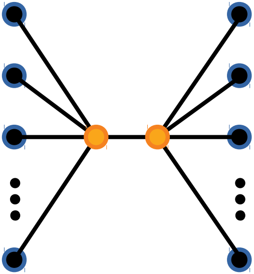
Interpretations

$$I = (\nu(x), \mu(x, y))$$

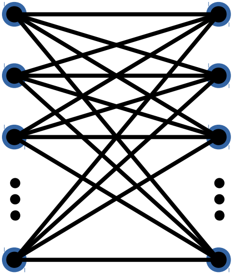
have distance three

is blue

G



$I(G)$



vertices: $\{v \mid G \models \nu(v)\}$

edges: $\{uv \mid G \models \mu(u, v)\}$

Structurally Property X

A graph class $\mathcal{G}$ has *structurally property X* if there exists

Structurally Property X

A graph class $\mathcal{G}$ has *structurally property X* if there exists

- a class $\mathcal{G}'$ with property X,

Structurally Property X

A graph class $\mathcal{G}$ has *structurally property X* if there exists

- a class $\mathcal{G}'$ with property X,
- an interpretation $I = (\nu(x), \mu(x, y))$,

Structurally Property X

A graph class $\mathcal{G}$ has *structurally property X* if there exists

- a class $\mathcal{G}'$ with property X,
- an interpretation $I = (\nu(x), \mu(x, y))$,

such that for every $G \in \mathcal{G}$ there is $G' \in \mathcal{G}'$ with $G = I(G')$.

Structurally Property X

A graph class $\mathcal{G}$ has *structurally property X* if there exists

- a class $\mathcal{G}'$ with property X,
- an interpretation $I = (\nu(x), \mu(x, y))$,

such that for every $G \in \mathcal{G}$ there is $G' \in \mathcal{G}'$ with $G = I(G')$.

The class of all fully bipartite graphs has *structurally treewidth 1*:

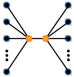
Structurally Property X

A graph class $\mathcal{G}$ has *structurally property X* if there exists

- a class $\mathcal{G}'$ with property X,
- an interpretation $I = (\nu(x), \mu(x, y))$,

such that for every $G \in \mathcal{G}$ there is $G' \in \mathcal{G}'$ with $G = I(G')$.

The class of all fully bipartite graphs has *structurally treewidth 1*:

- The class of all  has treewidth 1

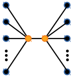
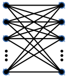
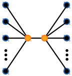
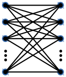
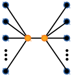
Structurally Property X

A graph class $\mathcal{G}$ has *structurally property X* if there exists

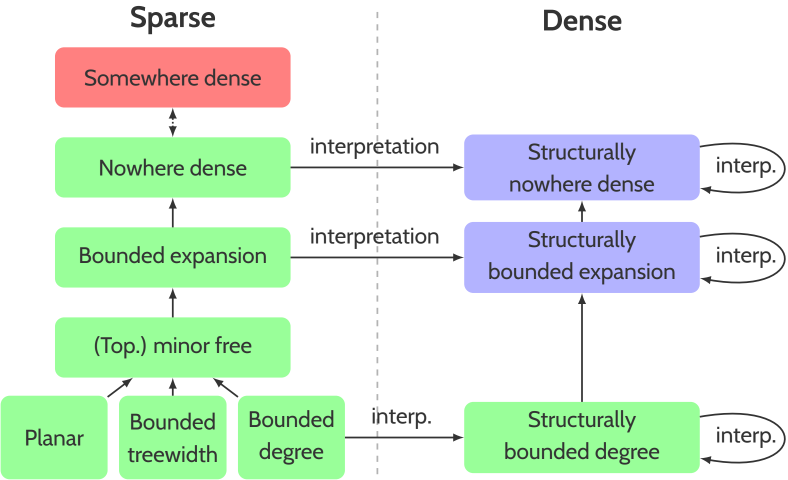
- a class $\mathcal{G}'$ with property X,
- an interpretation $I = (\nu(x), \mu(x, y))$,

such that for every $G \in \mathcal{G}$ there is $G' \in \mathcal{G}'$ with $G = I(G')$.

The class of all fully bipartite graphs has *structurally treewidth 1*:

- The class of all  has treewidth 1
- For every  there is  with  $= I\Big($  $\Big)$.

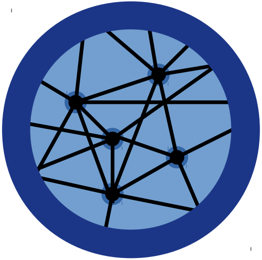
Model-Checking in Sparse and Dense Classes



Nowhere Dense: Grohe, Kreutzer, Sieberz 2011
Structurally Bounded Degree: Gajarský, Hlinenỳ, Obdržálek, Lokshtanov, Ramanujan 2016

Structurally Bounded Expansion

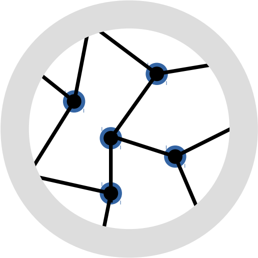
Structurally
Bounded Expansion
$I(G)$



Structurally Bounded Expansion

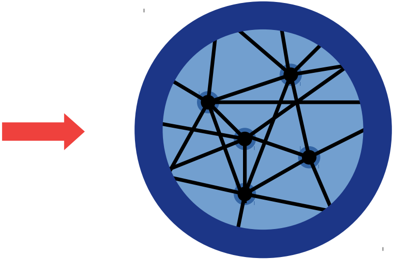
Bounded Expansion

G

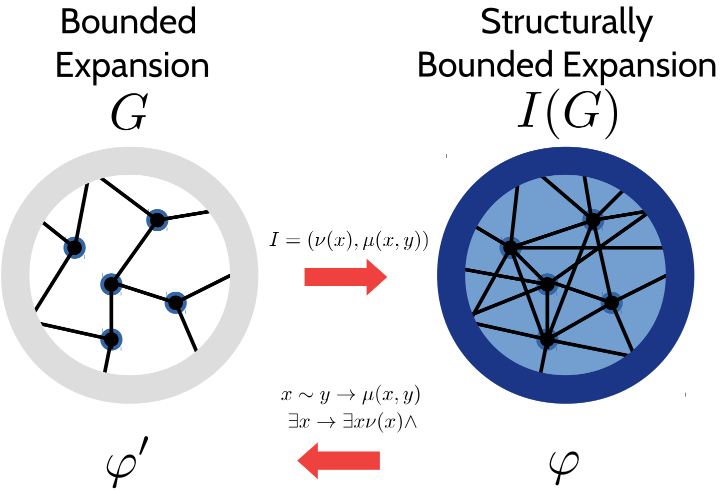


Structurally Bounded Expansion

$I(G)$



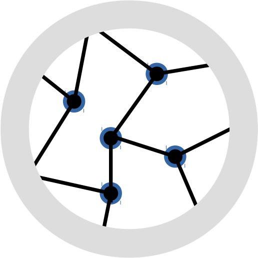
Structurally Bounded Expansion



Structurally Bounded Expansion

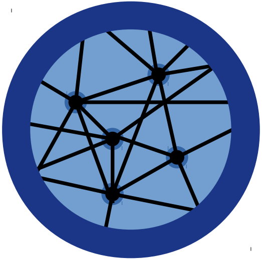
Bounded Expansion

G



Structurally Bounded Expansion

$I(G)$



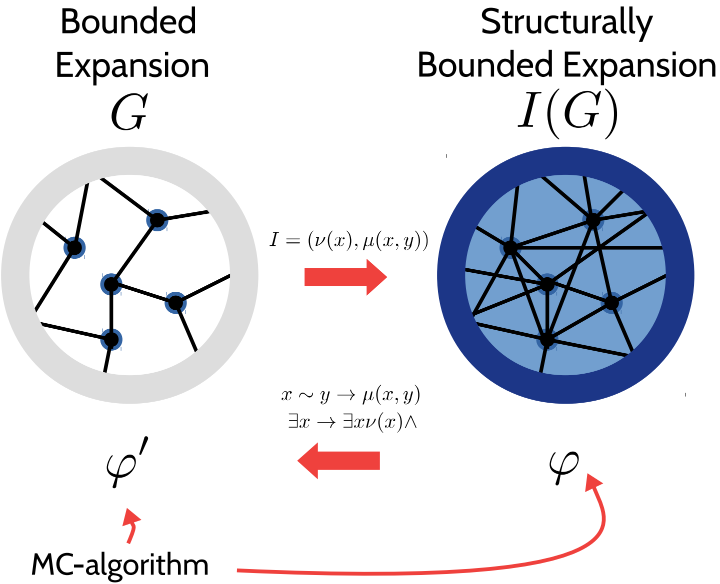
$$I = (\nu(x), \mu(x,y))$$

$$x \sim y \to \mu(x,y)$$
$$\exists x \to \exists x \nu(x) \wedge$$

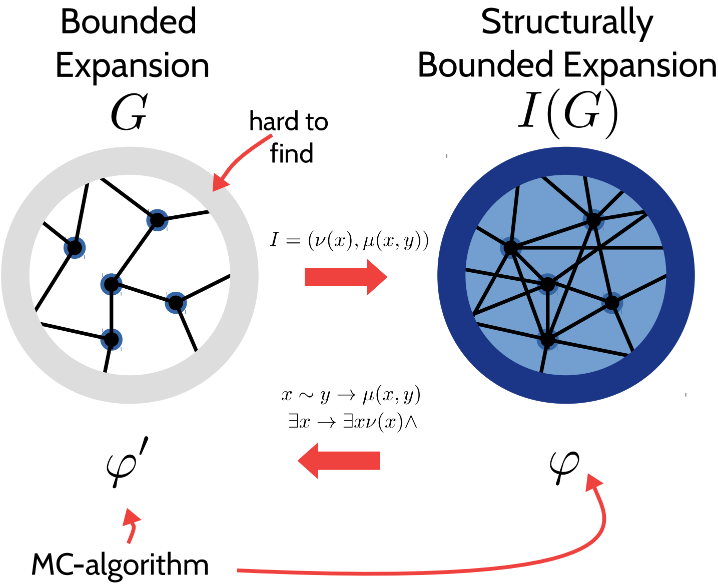
φ' $\qquad$ φ

MC-algorithm

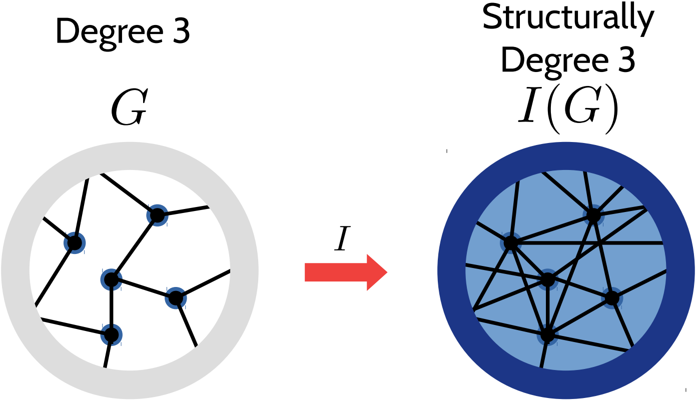
Structurally Bounded Expansion



Structurally Bounded Expansion

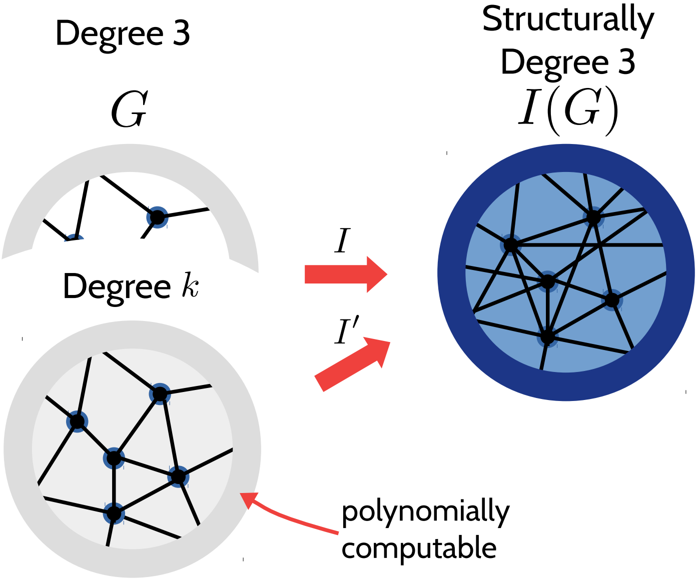


Structurally Bounded Degree

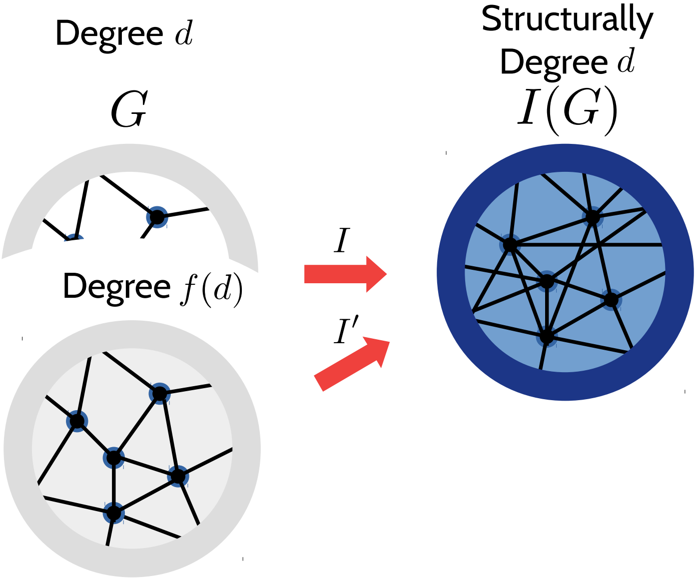


NP-complete to find preimage

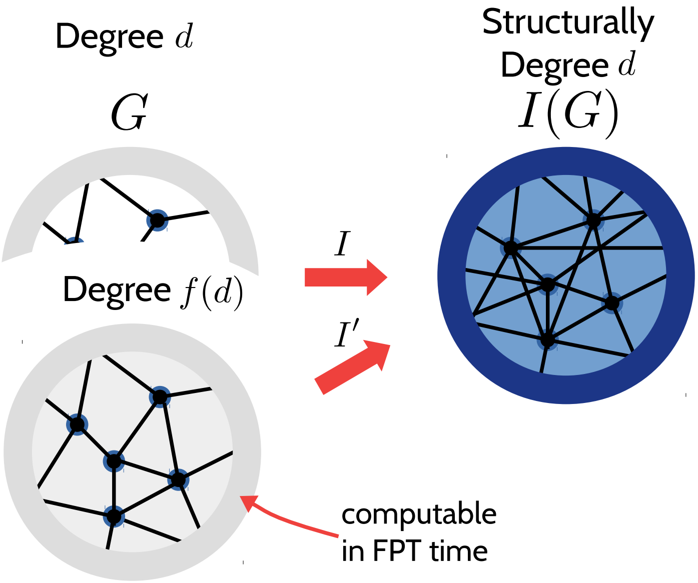
Structurally Bounded Degree



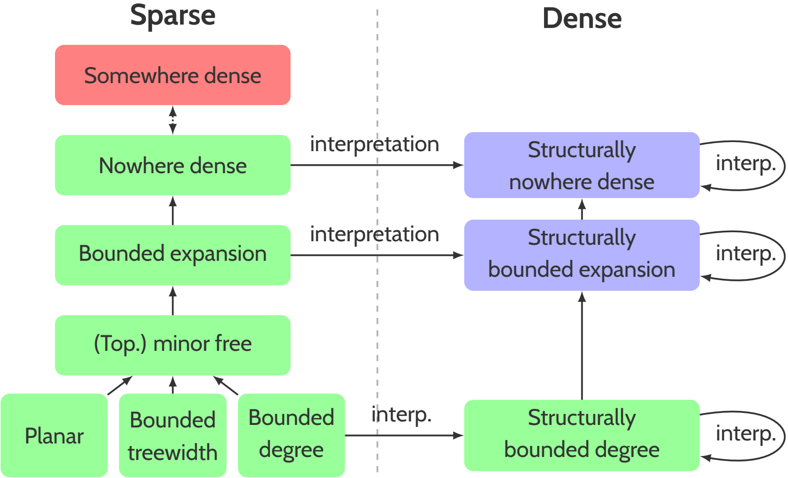
Structurally Bounded Degree



Structurally Bounded Degree

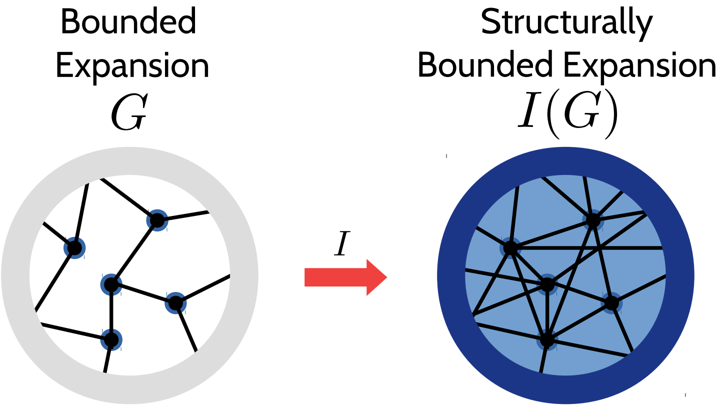


Model-Checking in Sparse and Dense Classes

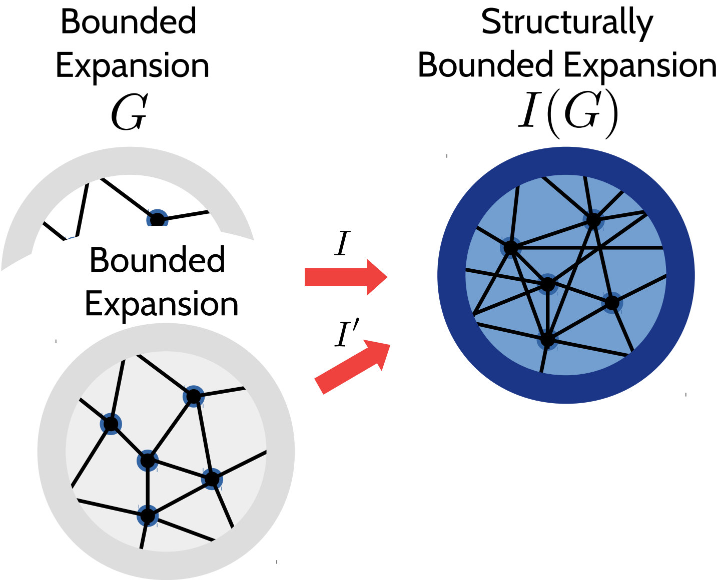


Nowhere Dense: Grohe, Kreutzer, Sieberz 2011
Structurally Bounded Degree: Gajarský, Hlinenỳ, Obdržálek, Lokshtanov, Ramanujan 2016

Big Question

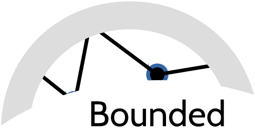


Big Question



Big Question

Bounded
Expansion
G

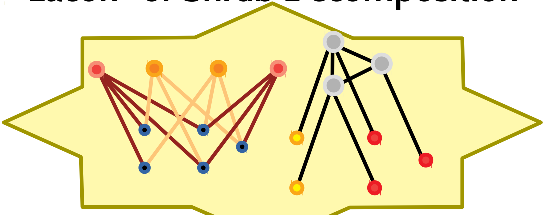


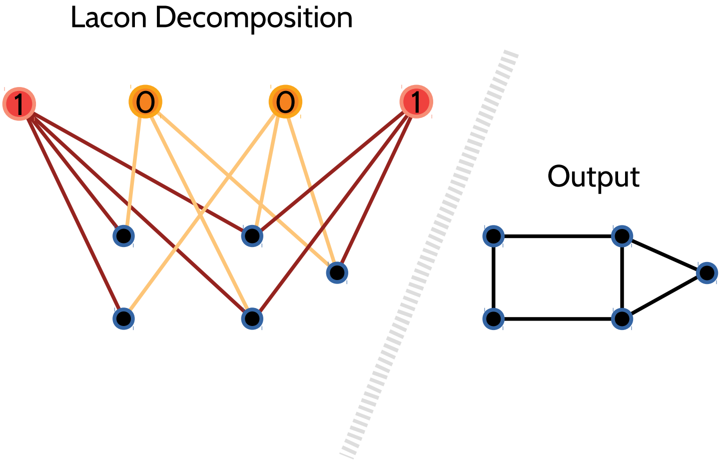
Bounded
Expansion

I

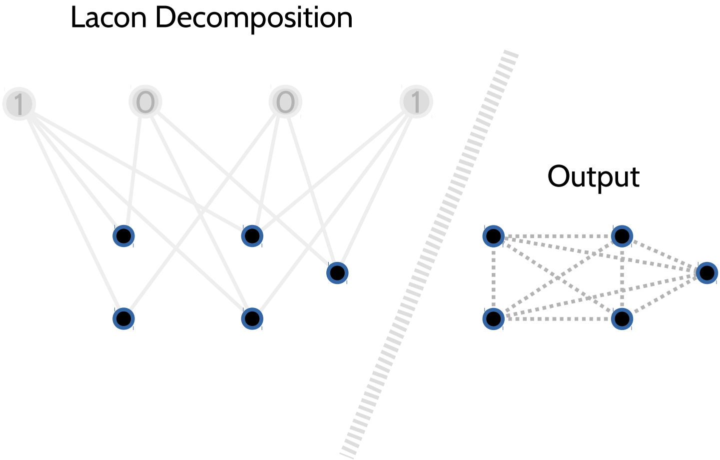
Structurally
Bounded Expansion
$I(G)$

Lacon- or Shrub Decomposition

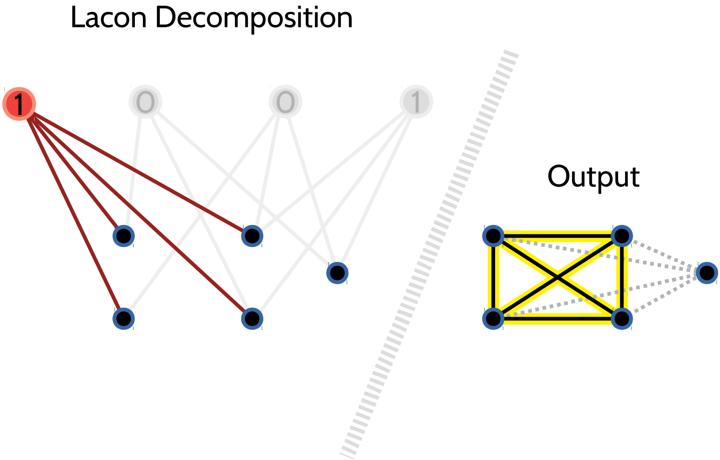



Lacon Decomposition
1
0
0
1
Output

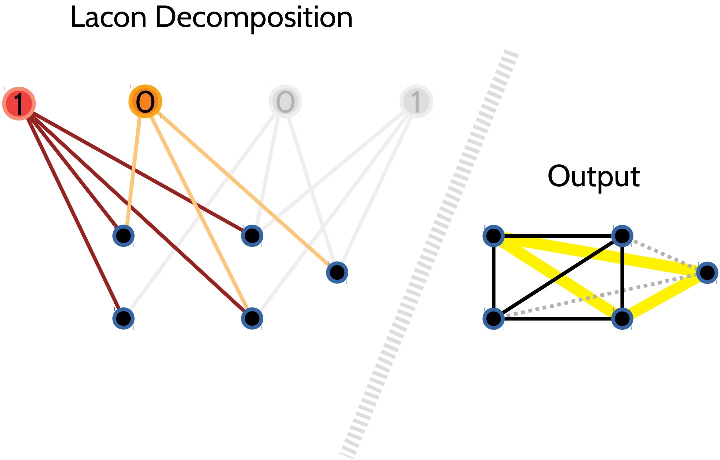
Lacon Decompositions

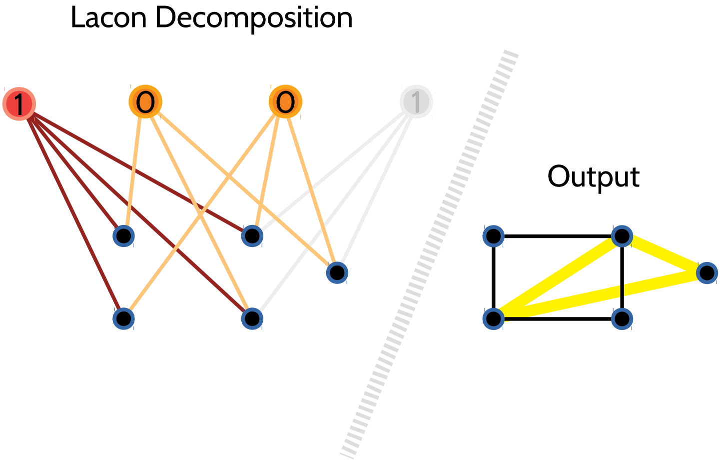


Lacon Decompositions



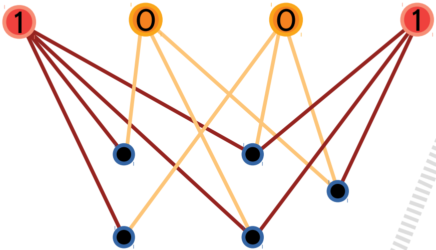
Lacon Decompositions



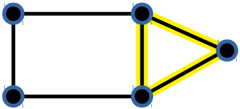

Lacon Decomposition
1
0
0
1
Output

Lacon Decompositions

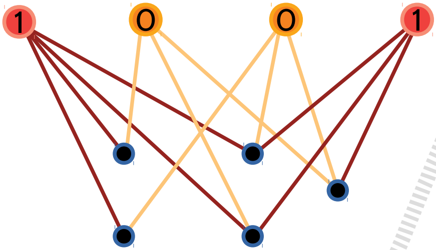
Lacon Decomposition



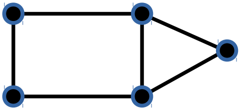
Output

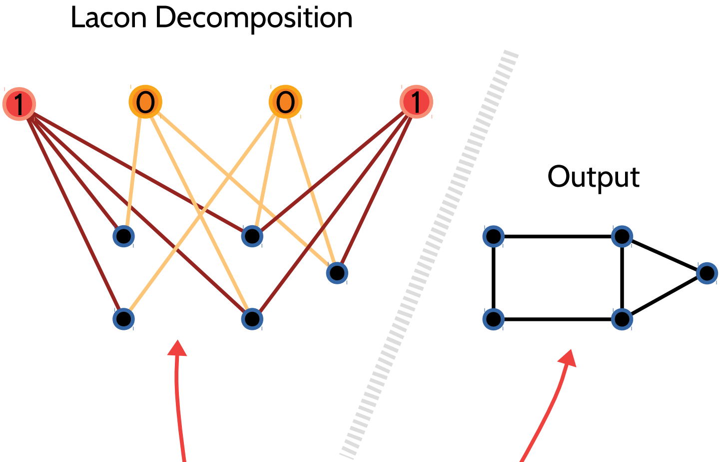


Lacon Decomposition



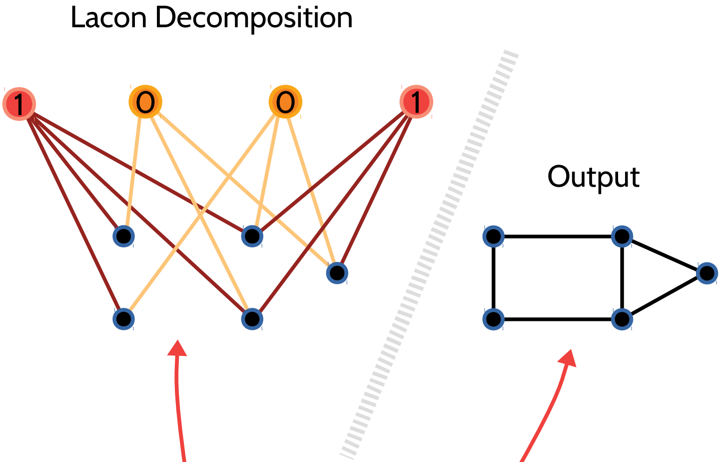
Output





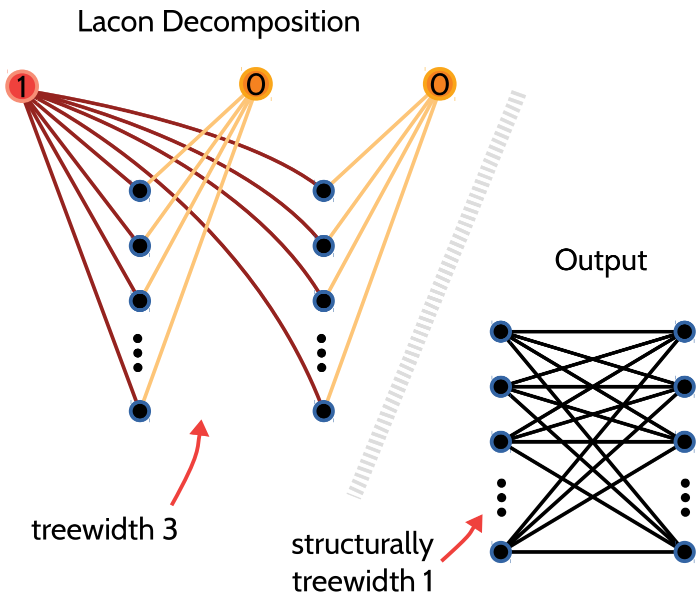
has lacon decomposition with property X $\implies$ structurally property X

Lacon Decompositions

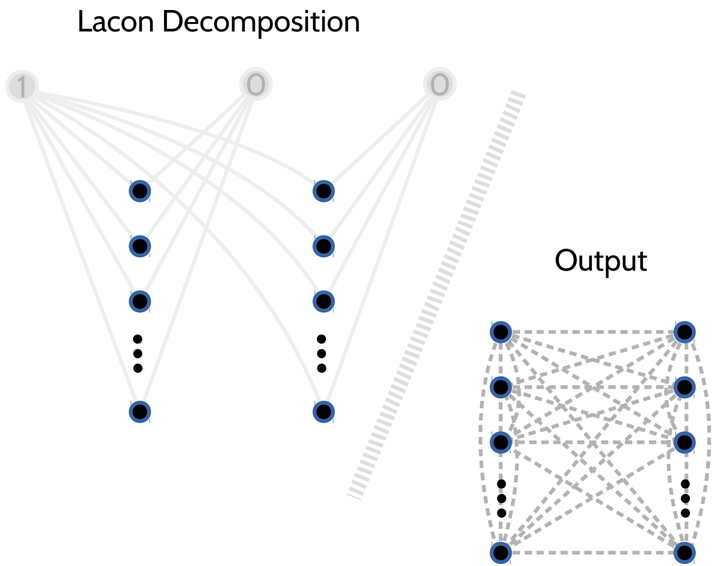


has lacon decomposition with property X $\iff$ structurally property X

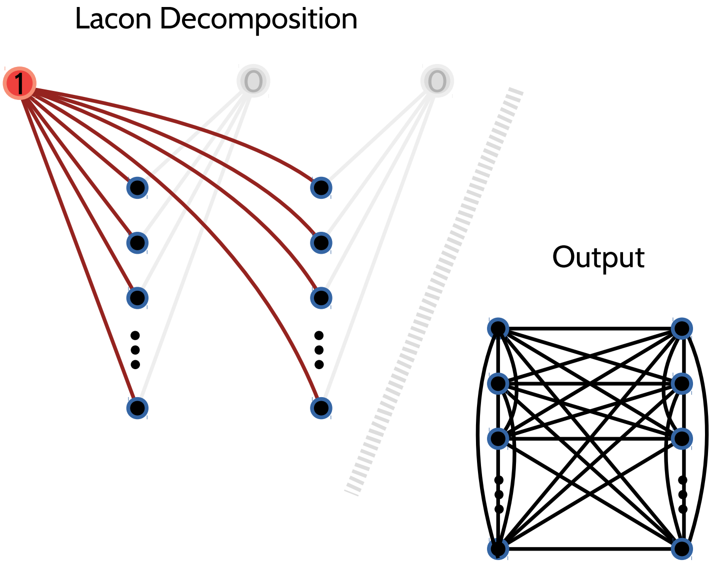
Lacon Decompositions



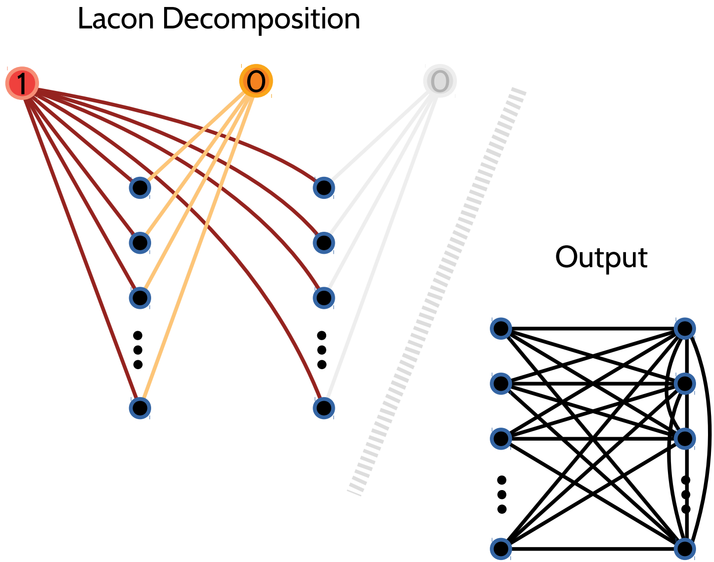
Lacon Decompositions



Lacon Decompositions

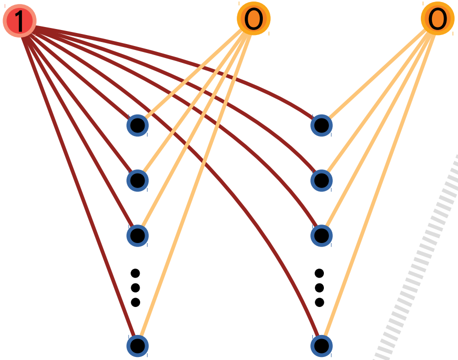


Lacon Decompositions

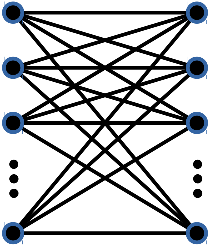


Lacon Decompositions

Lacon Decomposition



Output



Result

Theorem

Let $\mathcal{G}$ be a graph class. The following statements are equivalent.

Result

Theorem

Let $\mathcal{G}$ be a graph class. The following statements are equivalent.

- $\mathcal{G}$ has **structurally bounded expansion.**

Result

Theorem

Let $\mathcal{G}$ be a graph class. The following statements are equivalent.

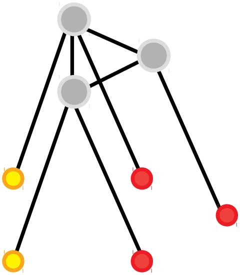
- $\mathcal{G}$ has **structurally bounded expansion.**
- $\mathcal{G}$ has **lacon decompositions** with
  - **bounded expansion,**
  - **bounded target vertex degree.**

Shrub Decompositions

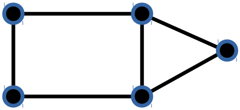
connect vertices with...

- distance 2
- distance 3 and same color

Shrub Decomposition



Output

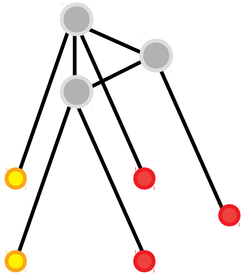


Shrub Decompositions

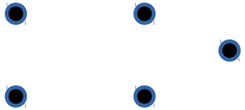
connect vertices with...

-
-

Shrub Decomposition



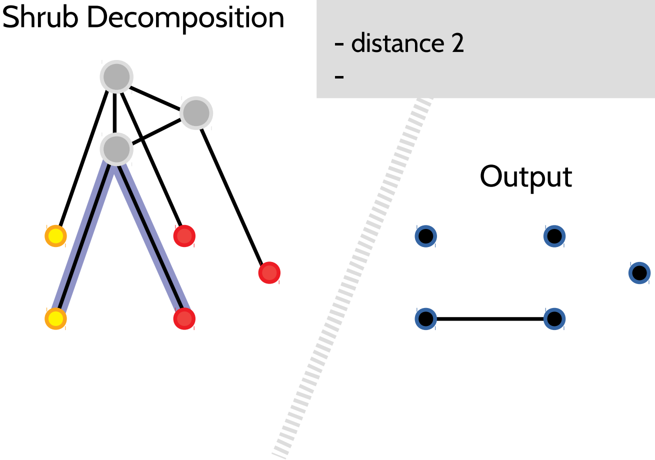
Output



Shrub Decompositions

connect vertices with...

- distance 2
-

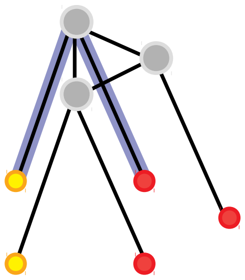


Shrub Decompositions

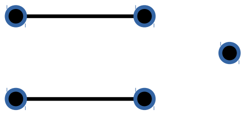
connect vertices with...

- distance 2
-

Shrub Decomposition



Output

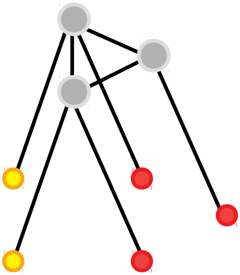


Shrub Decompositions

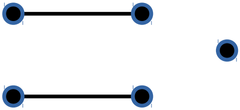
connect vertices with…

- distance 2
- distance 3 and same color

Shrub Decomposition



Output

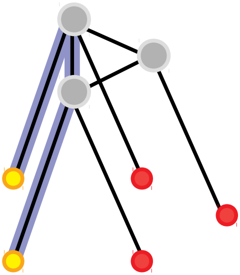


Shrub Decompositions

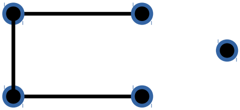
connect vertices with...

- distance 2
- distance 3 and same color

Shrub Decomposition



Output

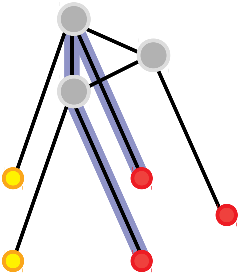


Shrub Decompositions

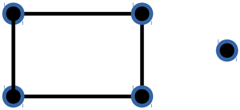
connect vertices with...

- distance 2
- distance 3 and same color

Shrub Decomposition



Output

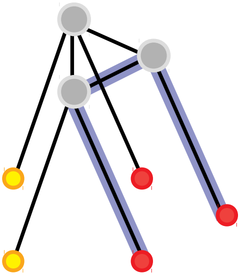


Shrub Decompositions

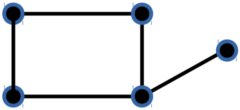
connect vertices with…

- distance 2
- distance 3 and same color

Shrub Decomposition



Output

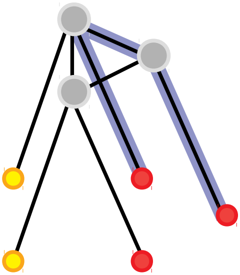


Shrub Decompositions

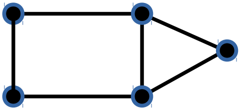
connect vertices with...

- distance 2
- distance 3 and same color

Shrub Decomposition



Output

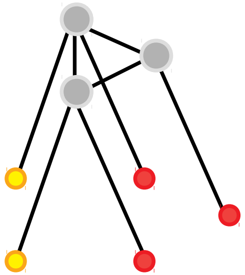


Shrub Decompositions

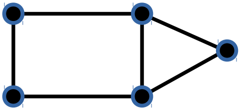
connect vertices with...

- distance 2
- distance 3 and same color

Shrub Decomposition



Output



has shrub decomposition with property X $\Longleftrightarrow$ structurally property X

Result

Theorem

Let $\mathcal{G}$ be a graph class. The following statements are equivalent.

Result

Theorem

Let $\mathcal{G}$ be a graph class. The following statements are equivalent.

- $\mathcal{G}$ has **structurally bounded expansion.**

Result

Theorem

Let $\mathcal{G}$ be a graph class. The following statements are equivalent.

- $\mathcal{G}$ has **structurally bounded expansion.**
- $\mathcal{G}$ has **lacon decompositions** with
  - **bounded expansion,**
  - **bounded target vertex degree.**

Result

Theorem

Let $\mathcal{G}$ be a graph class. The following statements are equivalent.

- $\mathcal{G}$ has **structurally bounded expansion.**
- $\mathcal{G}$ has **lacon decompositions** with
  - **bounded expansion,**
  - **bounded target vertex degree.**
- $\mathcal{G}$ has **shrub decompositions** with
  - **bounded expansion,**
  - **bounded number of colors,**
  - **bounded diameter.**

Result

Theorem

Let $\mathcal{G}$ be a graph class. The following statements are equivalent.

- $\mathcal{G}$ has **structurally bounded expansion.**
- $\mathcal{G}$ has **lacon decompositions** with
  - **bounded expansion,**
  - **bounded target vertex degree.**
- $\mathcal{G}$ has **shrub decompositions** with
  - **bounded expansion,**
  - **bounded number of colors,**
  - **bounded diameter.**
- $\mathcal{G}$ has **low shrubdepth covers** [1].

[1] Gajarský, Kreuzer, Nešetřil, Ossona de Mendez, Siebertz, Toruńczyk 2018

Comparison to Low Shrubdepth Covers

Lacon- and Shrub Decompositions

Low Shrubdepth Covers

Comparison to Low Shrubdepth Covers

Lacon- and Shrub Decompositions

- global

Low Shrubdepth Covers

- local

Comparison to Low Shrubdepth Covers

Lacon- and Shrub Decompositions

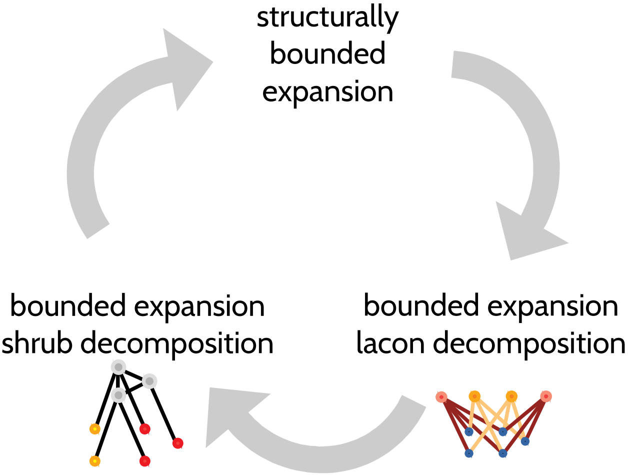
- global
- first-order types

Low Shrubdepth Covers

- local
- quantifier alternation

Proof Ideas

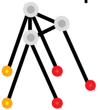
A Cycle of Implications



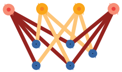
A Cycle of Implications

structurally bounded expansion

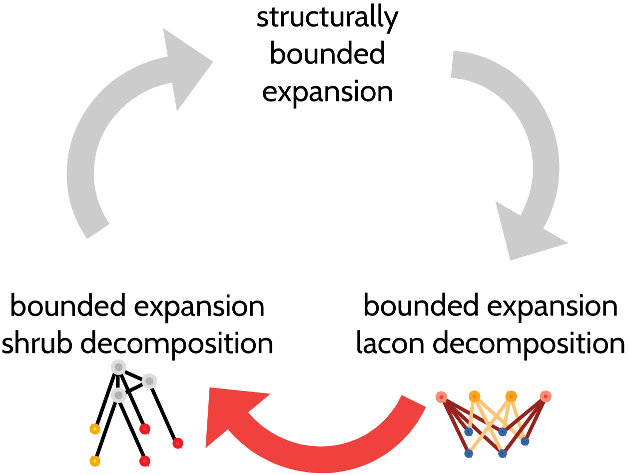
bounded expansion
shrub decomposition



bounded expansion
lacon decomposition



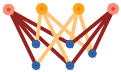
A Cycle of Implications



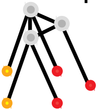
A Cycle of Implications

structurally
bounded
expansion

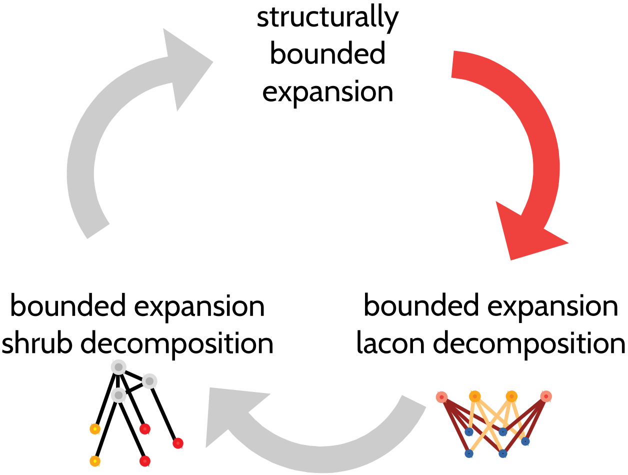
bounded expansion
lacon decomposition



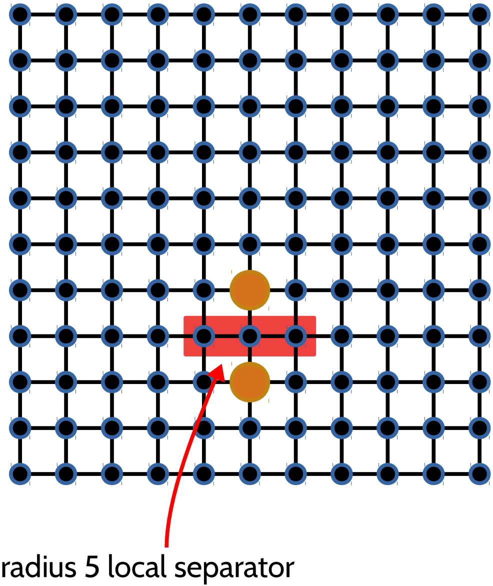
bounded expansion
shrub decomposition



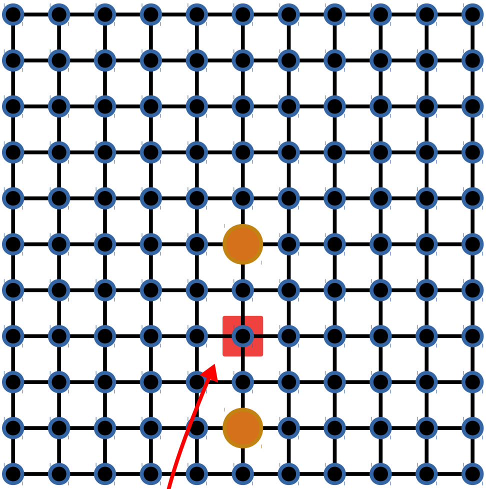
Step 1


structurally
bounded
expansion
bounded expansion
lacon decomposition
bounded expansion
shrub decomposition

Local Separators

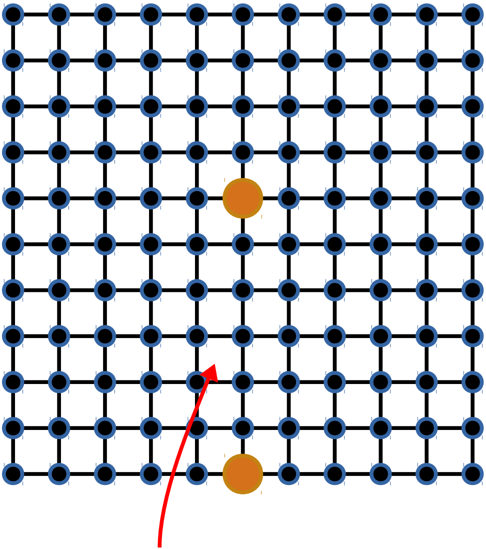


Local Separators

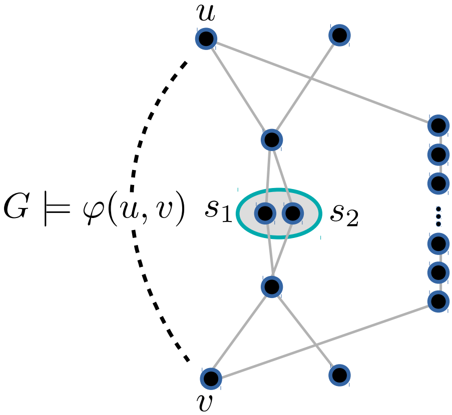


radius 5 local separator

Local Separators



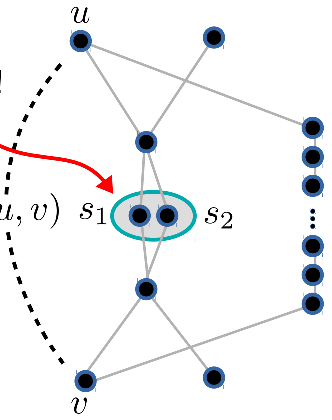
radius 5 local separator


u
$G \models \varphi(u, v)$
s_1
s_2
v

Local Feferman–Vaught

$(f(|\varphi|) = q)$-local separator!

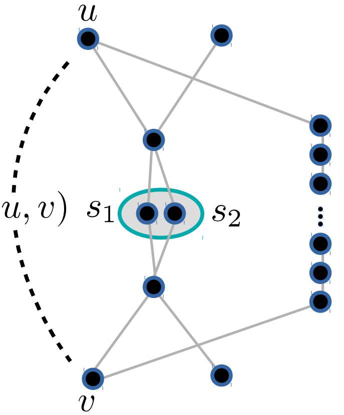
$G \models \varphi(u, v)$



Local Feferman–Vaught

$\mathrm{type}_q(u, s_1, s_2)$

$G \models \varphi(u, v)$



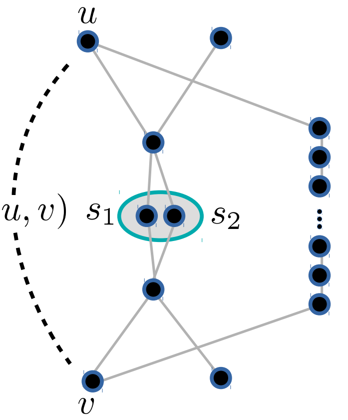
$\mathrm{type}_q(v, s_1, s_2)$

$\mathrm{type}_q(v, s_1, s_2) =$ all q-formulas $\psi(x, y, z)$
with $G \models \psi(v, s_1, s_2)$

Local Feferman–Vaught

$\mathrm{type}_q(u, s_1, s_2)$

$G \models \varphi(u, v)$



$\mathrm{type}_q(v, s_1, s_2)$

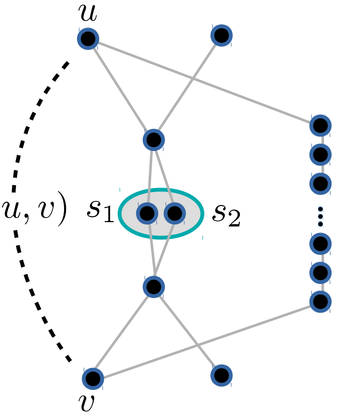
$\mathrm{type}_q(v, s_1, s_2)$ = all q-formulas $\psi(x, y, z)$ with $G \models \psi(v, s_1, s_2)$

FV: $\varphi(u, v)$ is determined by $\mathrm{type}_q(u, s_1, s_2)$ and $\mathrm{type}_q(v, s_1, s_2)$

Local Feferman–Vaught

$\mathrm{type}_q(u, s_1, s_2) = \tau_j$

$G \models \varphi(u, v)$



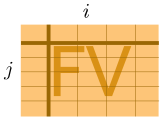
$\mathrm{type}_q(v, s_1, s_2) = \tau_i$

$\mathrm{type}_q(v, s_1, s_2) =$ all q-formulas $\psi(x, y, z)$ with $G \models \psi(v, s_1, s_2)$

FV: $\varphi(u, v)$ is determined by $\mathrm{type}_q(u, s_1, s_2)$ and $\mathrm{type}_q(v, s_1, s_2)$

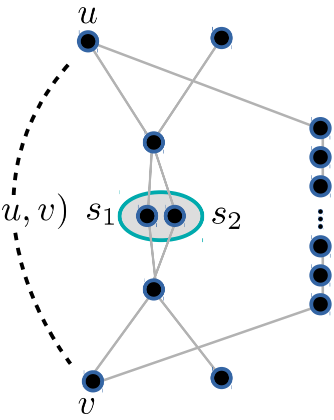
Local Feferman–Vaught

$\mathrm{type}_q(u, s_1, s_2) = \tau_j$



$G \models \varphi(u, v)$

$\mathrm{type}_q(v, s_1, s_2) = \tau_i$

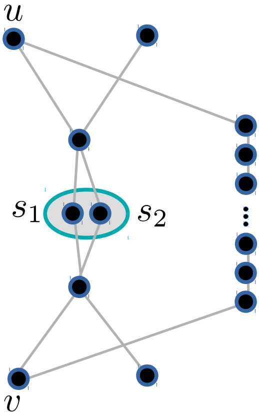


$\mathrm{type}_q(v, s_1, s_2)$ = all q-formulas $\psi(x, y, z)$ with $G \models \psi(v, s_1, s_2)$

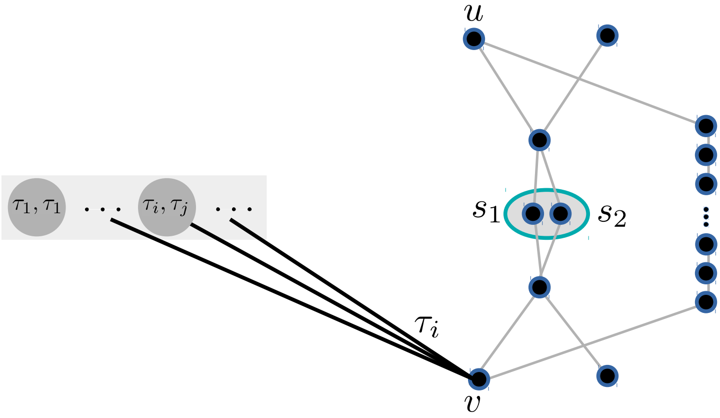
FV: $\varphi(u, v)$ is determined by $\mathrm{type}_q(u, s_1, s_2)$ and $\mathrm{type}_q(v, s_1, s_2)$

Constructing Lacon Decompositions

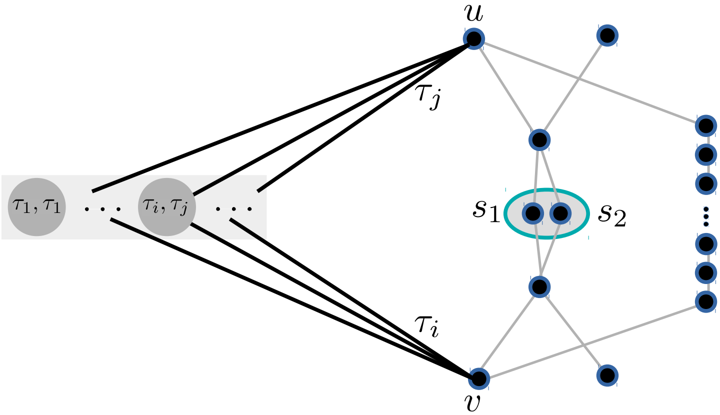
hidden vertices for
local separator s_1, s_2



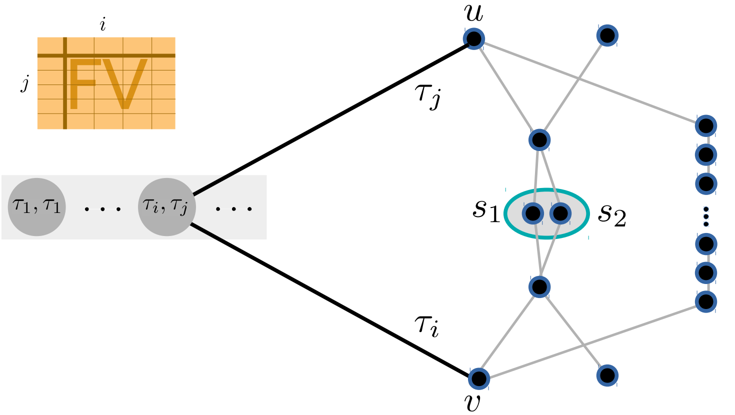
Constructing Lacon Decompositions

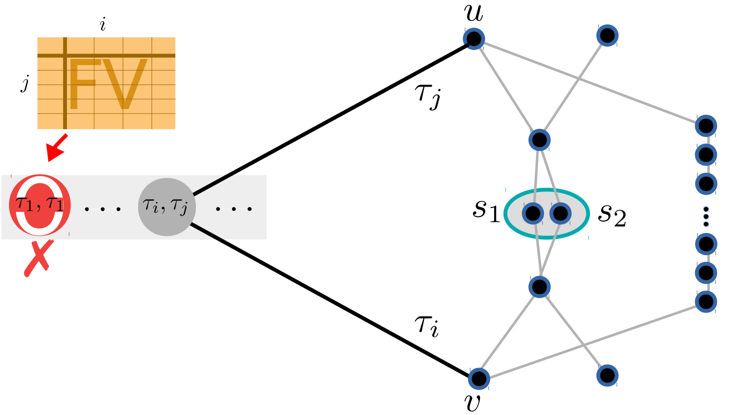


Constructing Lacon Decompositions



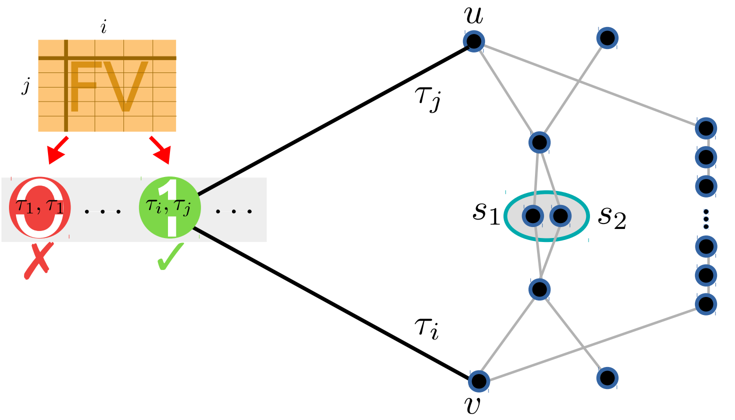
Constructing Lacon Decompositions



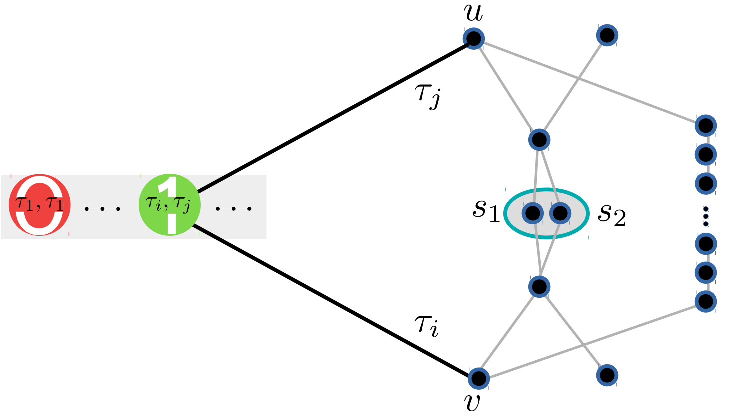


i
j
FV
τ_1, τ_1
τ_i, τ_j
u
τ_j
s_1
s_2
τ_i
v

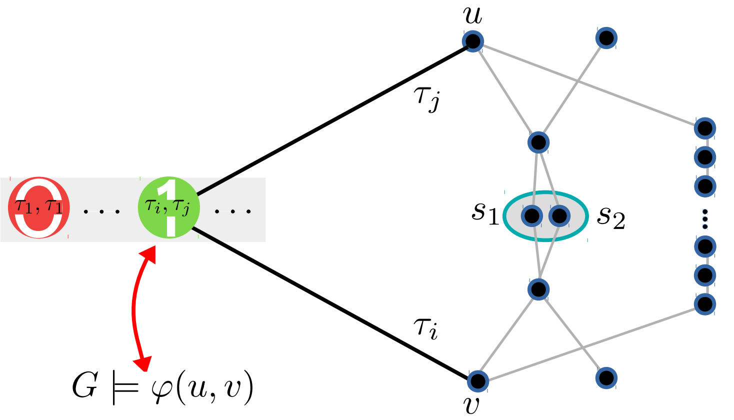
Constructing Lacon Decompositions



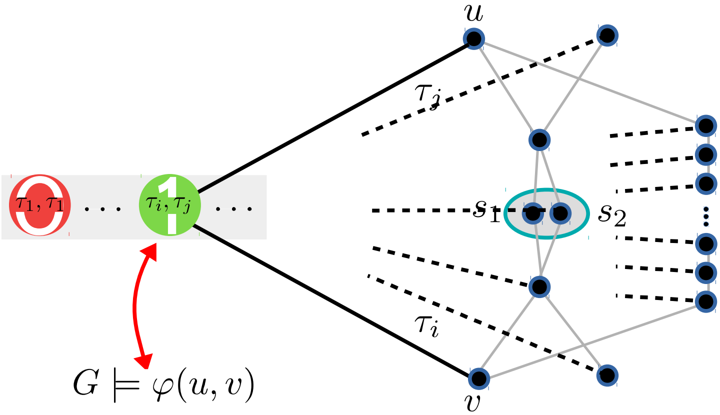
Constructing Lacon Decompositions



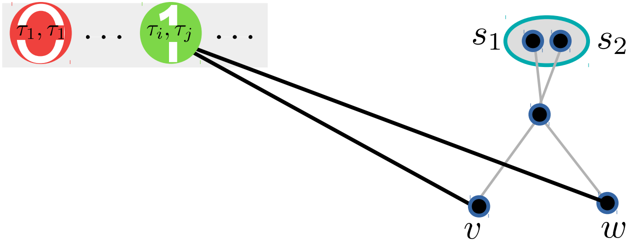
Constructing Lacon Decompositions



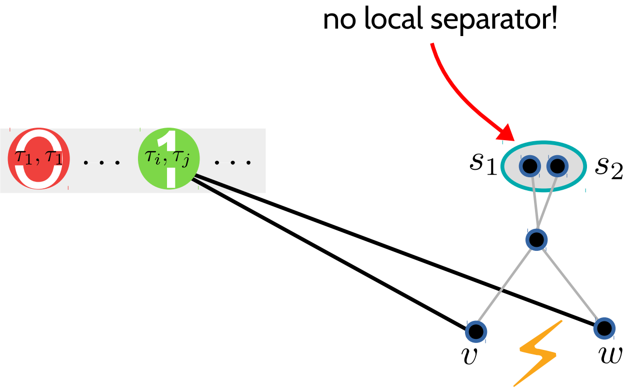
Constructing Lacon Decompositions



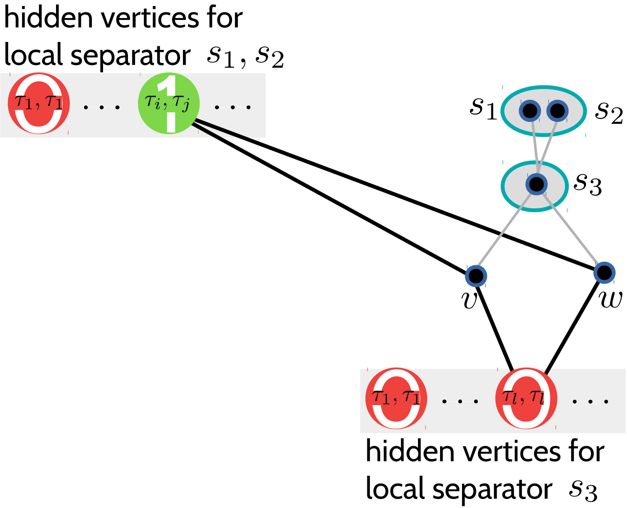
Constructing Lacon Decompositions



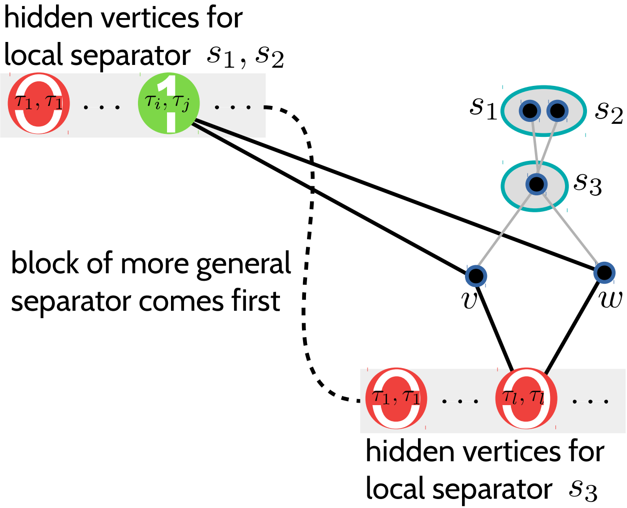
Constructing Lacon Decompositions



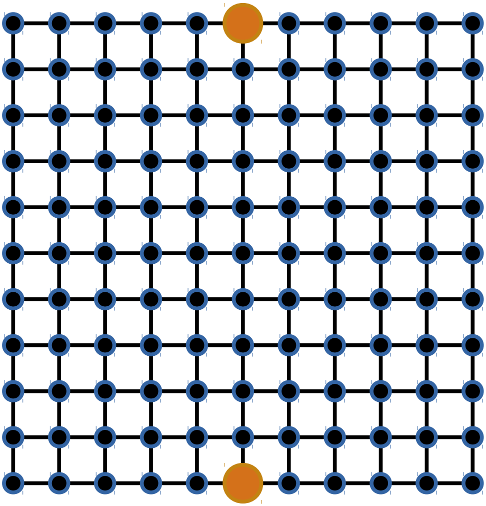
Constructing Lacon Decompositions



Constructing Lacon Decompositions

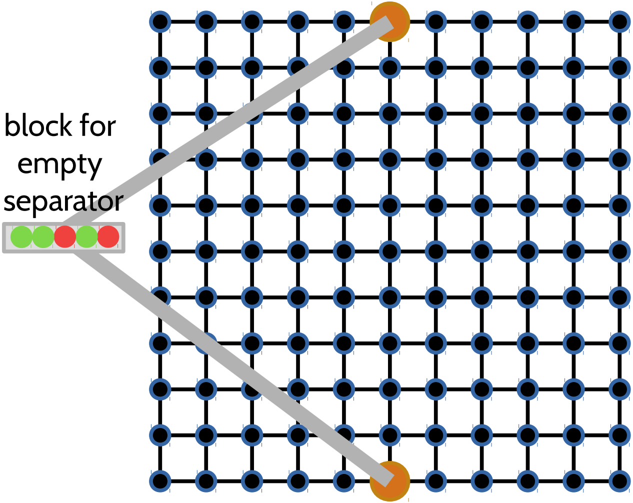


Constructing Lacon Decompositions

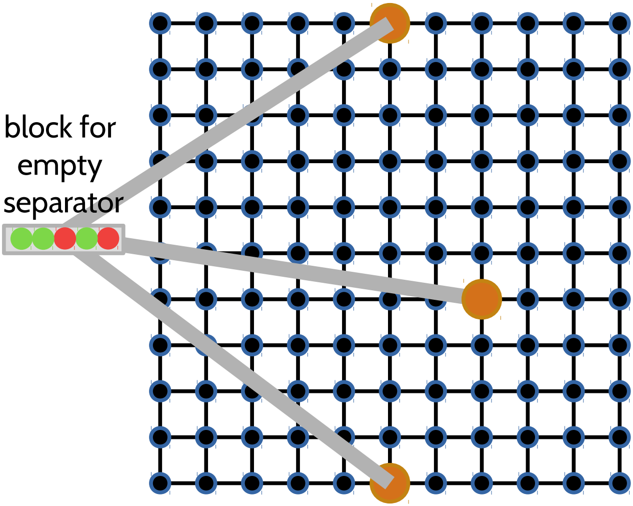


empty local separator

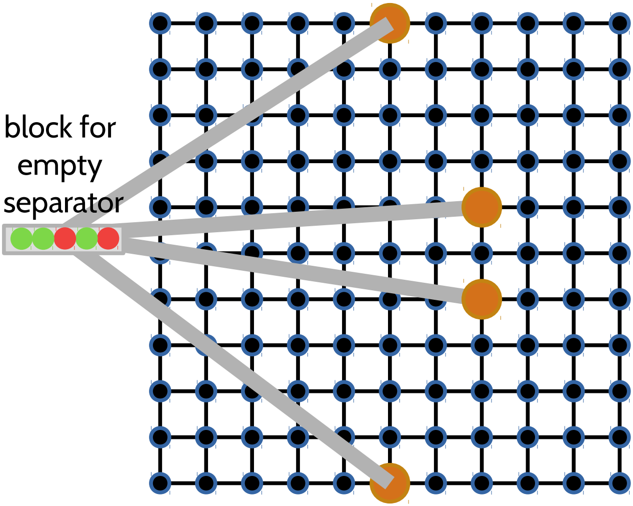
Constructing Lacon Decompositions



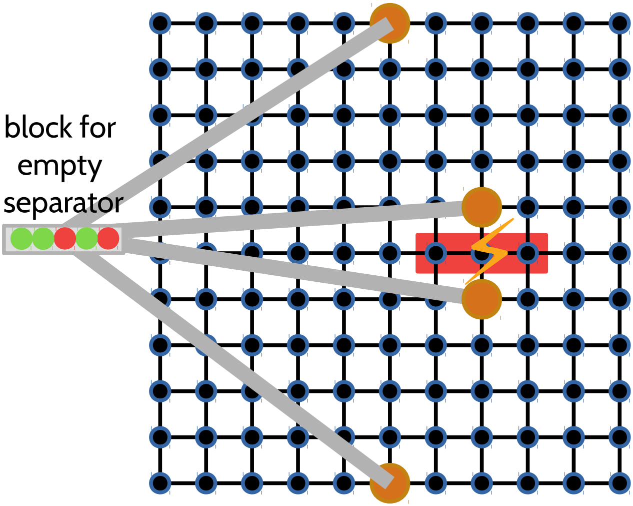
Constructing Lacon Decompositions



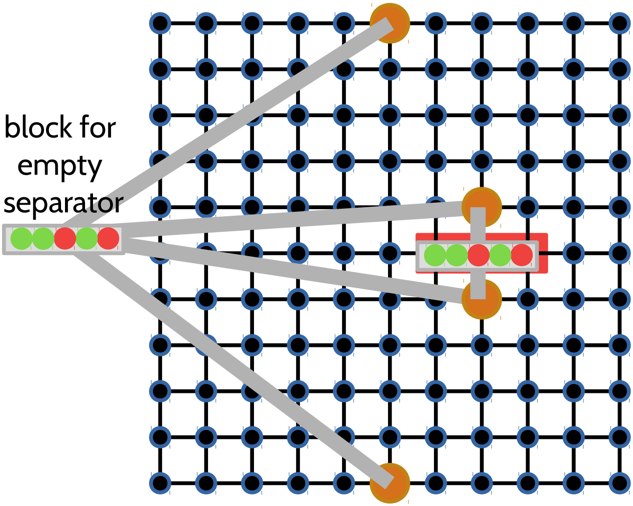
Constructing Lacon Decompositions



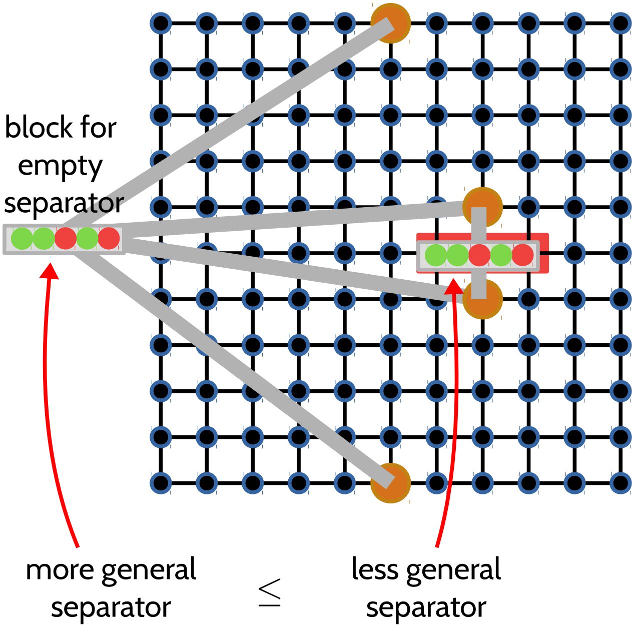
Constructing Lacon Decompositions



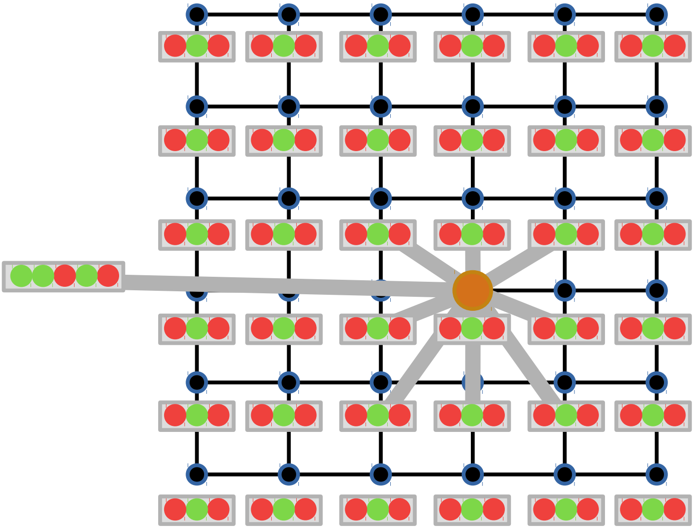
Constructing Lacon Decompositions



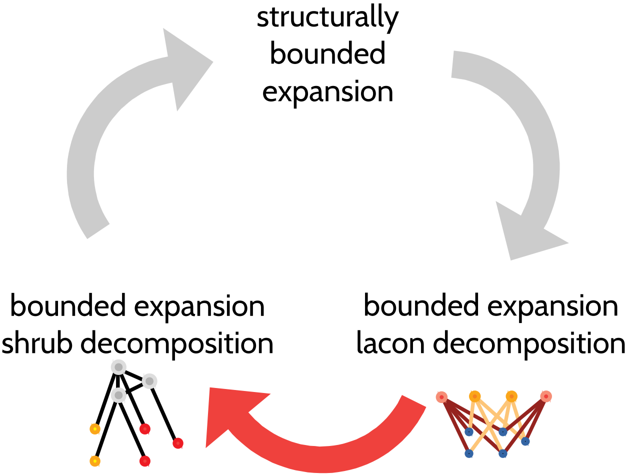
Constructing Lacon Decompositions



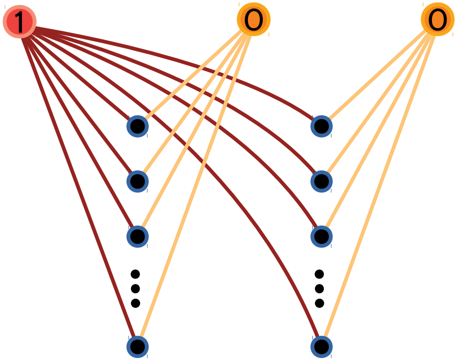
Constructing Lacon Decompositions



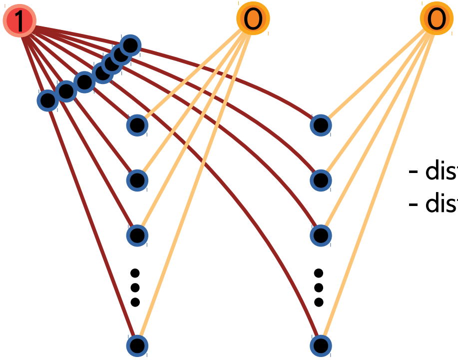
Step 2


structurally
bounded
expansion
bounded expansion
lacon decomposition
bounded expansion
shrub decomposition

Step 2

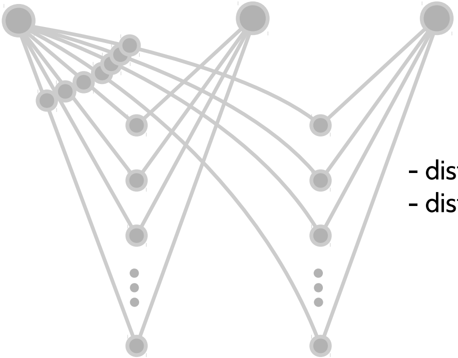


Step 2



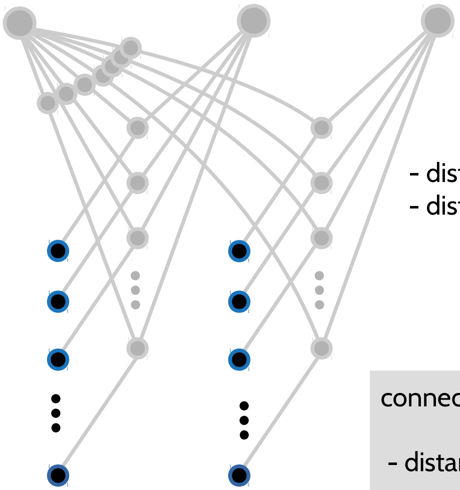
- distance 2: zero vertex
- distance 4: one vertex

Step 2



- distance 2: zero vertex
- distance 4: one vertex

Step 2

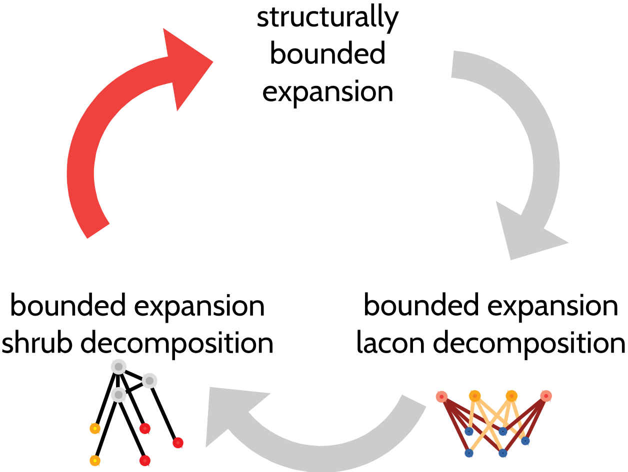


- distance 2: zero vertex
- distance 4: one vertex

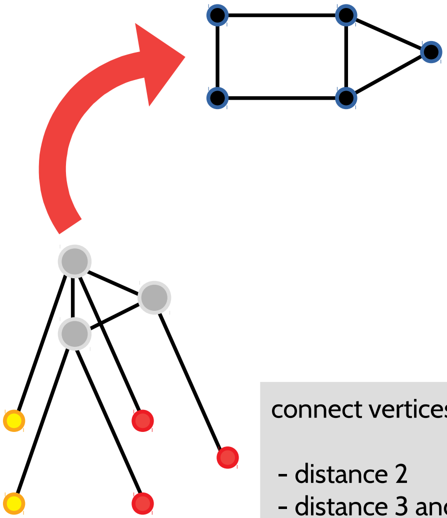
connect vertices with...

- distance 6

Step 3


structurally
bounded
expansion
bounded expansion
shrub decomposition
bounded expansion
lacon decomposition

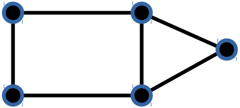
Step 3

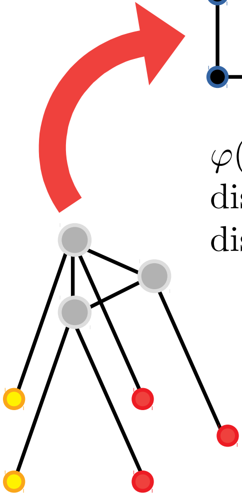


connect vertices with...

- distance 2
- distance 3 and same color

Step 3





$$\varphi(x, y) :=$$
$$\mathrm{dist}(x, y) = 2 \vee$$
$$\mathrm{dist}(x, y) = 3 \wedge c(x) = c(y)$$

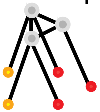
connect vertices with…

- distance 2
- distance 3 and same color

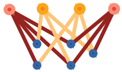
A Cycle of Implications

structurally
bounded
expansion

bounded expansion
shrub decomposition

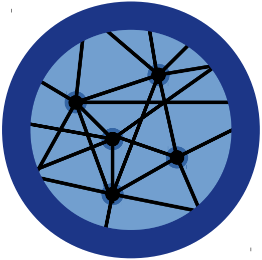


bounded expansion
lacon decomposition



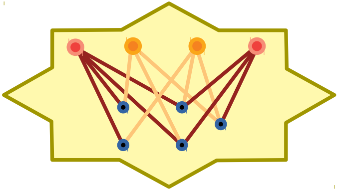
Big Question

Structurally
Bounded Expansion

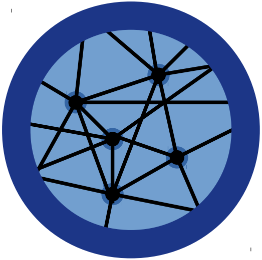


Big Question

has Decomposition
with Bounded Expansion

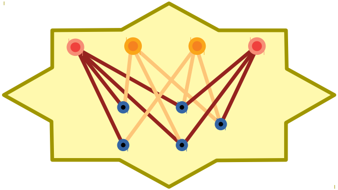


Structurally
Bounded Expansion



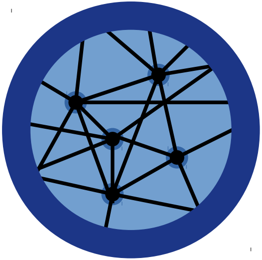
Big Question

has Decomposition
with Bounded Expansion

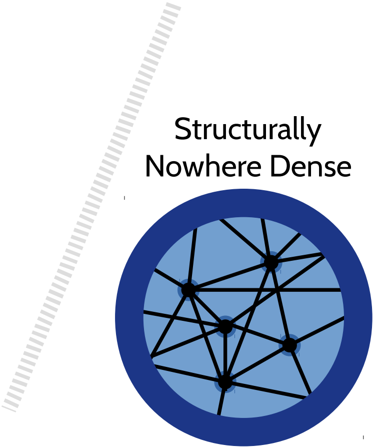


Can we compute it?

Structurally
Bounded Expansion

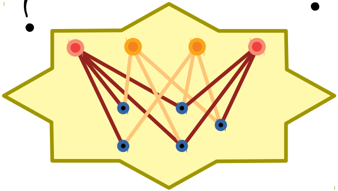


Big Question

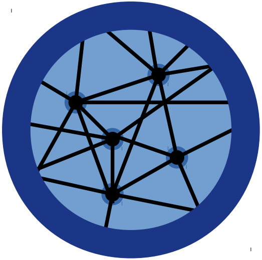


Big Question

has Nowhere Dense Decomposition?



Structurally Nowhere Dense



End

Thanks!