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”
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”
“All problems expressible in a certain logic can be solved efficiently on certain graphs”
Some Sparse Graph Classes

- Planar
- Bounded Degree
- Bounded Treewidth

(minor free, bounded expansion, somewhere dense, nowhere dense)
Some Sparse Graph Classes

- Planar
- (Top.) Minor Free
- Bounded Degree
- Bounded Treewidth
Some Sparse Graph Classes

- Bounded Expansion
- (Top.) Minor Free
- Planar
- Bounded Degree
- Bounded Treewidth
Some Sparse Graph Classes

- Nowhere Dense
- Bounded Expansion
- (Top.) Minor Free
- Planar
- Bounded Degree
- Bounded Treewidth
Some Sparse Graph Classes

- Somewhere Dense
- Nowhere Dense
- Bounded Expansion
- (Top.) Minor Free
- Planar
- Bounded Degree
- Bounded Treewidth
Some Sparse Graph Classes

- Somewhere Dense
- Nowhere Dense
- Bounded Expansion
- (Top.) Minor Free
  - Planar
  - Bounded Degree
  - Bounded Treewidth
Bounded Expansion — Minors

\[ \frac{|E|}{|V|} \leq C \]

for every graph in the graph class.
Bounded Expansion — Minors

For every graph in the graph class,

\[ \frac{|E|}{|V|} \leq 2 \]

for every graph in the graph class.
Bounded Expansion — Minors

radius $r$

$r$–shallow minor
Bounded Expansion — Minors

radius $r$

$r$—shallow minor
Bounded Expansion — Minors

radius $r$

$r$ — shallow minor
Bounded Expansion — Minors

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

for every r-shallow minor of every graph in the graph class.
### FO Model-Checking

[Dvořák, Král, Thomas 2010]

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

<table>
<thead>
<tr>
<th>Problem</th>
<th>Algorithmic Meta-Theorems</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>FO Model-Checking</strong></td>
</tr>
<tr>
<td></td>
<td>First-order formulas $\varphi$ can be evaluated on bounded expansion classes in time $f(</td>
</tr>
</tbody>
</table>
FO Model-Checking

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

- independent set of size $k$:

$$\exists x_1 \ldots \exists x_k \bigwedge_{i,j} x_i \not\sim x_j \land x_i \neq x_j$$
FO Model-Checking

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

- independent set of size $k$:

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

- dominating set of size $k$:

  $$\exists x_1 \ldots \exists x_k \forall y \bigvee_{i} y \sim x_i \lor y = x_i$$
Algorithmic Meta-Theorems

FO Model-Checking

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

- independent set of size $k$:

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

- dominating set of size $k$:

  $$\exists x_1 \ldots \exists x_k \forall y \bigvee_i y \sim x_i \lor y = x_i$$

Best algorithms on general graphs: $n^{O(k)}$
Algorithmic Meta-Theorems

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

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

- Independent set of size $k$:
  \[ \exists x_1 \ldots \exists x_k \bigwedge_{i,j} x_i \not\sim x_j \land x_i \neq x_j \]

- Dominating set of size $k$:
  \[ \exists x_1 \ldots \exists x_k \forall y \bigvee_{i} y \sim x_i \lor y = x_i \]

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]
What dense graph classes are tractable?
What dense graph classes are tractable?

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

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

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

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

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

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

Closure under subgraphs is not a good requirement.
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

\[ x_1 \not\sim x_2 \]

\[ x_1 \sim x_2 \]
Example: Complements

\[
\exists x_1 \exists x_2 \exists x_3 \\
x_1 \not\sim x_2 \land x_2 \not\sim x_3 \\
\land x_1 \not\sim x_3
\]

\[
\exists x_1 \exists x_2 \exists x_3 \\
x_1 \sim x_2 \land x_2 \sim x_3 \\
\land x_1 \sim x_3
\]
Example: Fully Bipartite
Example: Fully Bipartite
Example: Fully Bipartite

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

$x \sim y$
Example: Fully Bipartite

\[ \text{dist}(x, y) = 3 \]

\[ \exists x \; \text{blue}(x) \land \varphi \]

\[ x \sim y \]

\[ \exists x \; \varphi \]
Example: Fully Bipartite

\[ \exists x_1 \exists x_2 \exists x_3 \]
\[ \text{blue}(x_1) \land \text{blue}(x_2) \land \text{blue}(x_3) \]
\[ \text{dist}(x_1, x_2) = 3 \land \text{dist}(x_2, x_3) = 3 \]
\[ \land \text{dist}(x_1, x_3) = 3 \]

\[ \exists x_1 \exists x_2 \exists x_3 \]
\[ x_1 \sim x_2 \land x_2 \sim x_3 \]
\[ \land x_1 \sim x_3 \]
\[ I = (\nu(x), \mu(x, y)) \]
Interpretations

\[ I = (\nu(x), \mu(x, y)) \]

Diagram:

- **vertices:** \( \{v \mid G \models \nu(v)\} \)
- **edges:** \( \{uv \mid G \models \mu(u, v)\} \)
Interpretations

\[ I = (\nu(x), \mu(x, y)) \]

\[ G \]

is blue

\[ I(G) \]

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

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

\[ I = (\nu(x), \mu(x, y)) \]

\( G \)

\( I(G) \)

is blue

have distance three

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

edges: \( \{uv \mid G \models \mu(u, v)\} \)
A graph class $\mathcal{G}$ has *structurally property* $X$ if there exists
A graph class $\mathcal{G}$ has \textit{structurally property} $X$ if there exists

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

The class of all fully bipartite graphs has \textit{structurally treewidth} $1$: 

For every there is with $\mathcal{G}' = I(\mathcal{G})$. 

10
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

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

For every there is with $\mathcal{G}'$ with $\mathcal{G} = I(\mathcal{G}')$. 

A graph class $G$ has *structurally property* $X$ if there exists

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

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

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

For every there is with $G = I(G')$. 


A graph class $\mathcal{G}$ has \textit{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 \textit{structurally treewidth} 1:
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 \[\begin{array}{c}
\text{(a)}
\end{array}\] has treewidth 1
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 \[\begin{array}{c}
\end{array}\] has treewidth 1
- For every \[\begin{array}{c}
\end{array}\] there is \[\begin{array}{c}
\end{array}\] with \[\begin{array}{c}
\end{array}\] = $I(\begin{array}{c}
\end{array})$. 
Model-Checking in Sparse and Dense Classes

Sparse

Somewhere dense

Nowhere dense

Bounded expansion

(Top.) minor free

Planar Bounded treewidth Bounded degree

Dense

Structurally nowhere dense

Structurally bounded expansion

Structurally bounded degree

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

$I(G)$
Structurally Bounded Expansion

Bounded Expansion $G$

Structurally Bounded Expansion $I(G)$
Structurally Bounded Expansion

Bounded Expansion $G$

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

Structurally Bounded Expansion $I(G')$

$x \sim y \rightarrow \mu(x, y)$

$\exists x \rightarrow \exists x \nu(x) \land$

$\varphi'$

$\varphi$
Structurally Bounded Expansion

\[ G \]

\[ I = (\nu(x), \mu(x, y)) \]

\[ x \sim y \rightarrow \mu(x, y) \]
\[ \exists x \rightarrow \exists x \nu(x) \land \]

MC-algorithm

\[ \varphi' \]

\[ \varphi \]

\[ I(G) \]
Structurally Bounded Expansion

\[ I(G) \]

\[ I = (\nu(x), \mu(x, y)) \]

\[ x \sim y \rightarrow \mu(x, y) \]
\[ \exists x \rightarrow \exists x \nu(x) \wedge \]

\[ \varphi' \]

MC-algorithm

\[ \varphi \]
Structurally Bounded Expansion

Bounded Expansion $G$

hard to find

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

Structurally Bounded Expansion $I(G)$

$x \sim y \rightarrow \mu(x, y)$

$\exists x \rightarrow \exists \nu(x) \land$

MC-algorithm $\varphi'$

$\varphi$
Structurally Bounded Degree

Degree 3

$G$

Structurally
Degree 3

$I(G)$

NP-complete to find preimage
Structurally Bounded Degree

Degree 3

\[ G \]

Degree \( k \)

Structurally

Degree 3

\[ I(G) \]

polynomially computable
Structurally Bounded Degree

Degree $d$

$G$

Degree $f(d)$

Structurally
Degree $d$

$I(G)$

$I$

$I'$
Structurally Bounded Degree

Degree $d$

$G$

Degree $f(d)$

Structurally

Degree $d$

$I(G)$

Computable in FPT time

$I$

$I'$
Model-Checking in Sparse and Dense Classes

Sparse

- Somewhere dense
- Nowhere dense
- Bounded expansion
- (Top.) minor free
  - Planar
  - Bounded treewidth
  - Bounded degree

Dense

- Structurally nowhere dense
- Structurally bounded expansion
- Structurally bounded degree

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

Bounded Expansion

\[ G \]

Structured Bounded Expansion

\[ I(G) \]
Big Question

Bounded Expansion $G$

Structurally Bounded Expansion $I(G)$
Lacon Decompositions

Lacon Decomposition

Output

1 0 0 1
Lacon Decompositions

Lacon Decomposition

Output
Lacon Decompositions

Lacon Decomposition

Output
Lacon Decompositions

Lacon Decomposition

Output
Lacon Decompositions

Lacon Decomposition

Output

16
Lacon Decompositions

Lacon Decomposition

Output

1 0 0 1
Lacon Decompositions

Lacon Decomposition

\[
\begin{bmatrix}
1 & 0 & 0 & 1 \\
\end{bmatrix}
\]

Output

structurally

property X

has lacon decomposition

with property X

has lacon decomposition

with property X

\[\implies\]

structurally

property X

\[\implies\]

property X
Lacon Decompositions

Lacon Decomposition

Output

has lacon decomposition with property X

structurally property X
Lacon Decompositions

Lacon Decomposition

Output

structurally

treewidth 1

treewidth 3

...structurally treewidth 1...
Lacon Decomposition

Output

1 0 0
Lacon Decompositions

Lacon Decomposition

Output
Lacon Decompositions

Lacon Decomposition

Output

1
0
0

Output
Theorem

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

connect vertices with...
- distance 2
- distance 3 and same color
Shrub Decompositions

Shrub Decomposition

connect vertices with...
-
-

Output
Shrub Decompositions

Shrub Decomposition

connect vertices with...
- distance 2
-
Shrub Decompositions

Output

connect vertices with...
- distance 2
-
Shrub Decomposition

- distance 2
- distance 3 and same color

Output

connect vertices with...
Shrub Decompositions

Shrub Decomposition

- distance 2
- distance 3 and same color

connect vertices with...

Output
Shrub Decompositions

Output

connect vertices with...
- distance 2
- distance 3 and same color
Shrub Decompositions

Shrub Decomposition

- distance 2
- distance 3 and same color

connect vertices with...

Output
Shrub Decompositions

- Distance 2
- Distance 3 and same color

connect vertices with...

Shrub Decomposition

Output
Shrub Decompositions

Shrub Decomposition

connect vertices with...
- distance 2
- distance 3 and same color

Output

has shrub decomposition with property X

structurally property X
Theorem

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

1. \( \mathcal{G} \) has structurally bounded expansion.
2. \( \mathcal{G} \) has lacon decompositions with bounded expansion, bounded target vertex degree.
3. \( \mathcal{G} \) has shrub decompositions with bounded expansion, bounded number of colors, bounded diameter.
4. \( \mathcal{G} \) has low shrubdepth covers.

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

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

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

1. $\mathcal{G}$ has structurally bounded expansion.

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

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

<table>
<thead>
<tr>
<th>Lacon- and Shrub Decompositions</th>
<th>Low Shrubdepth Covers</th>
</tr>
</thead>
</table>
Comparison to Low Shrubdepth Covers

Lacon- and Shrub Decompositions

- global

Low Shrubdepth Covers

- local
Comparison to Low Shrubdepth Covers

<table>
<thead>
<tr>
<th>Lacon- and Shrub Decompositions</th>
<th>Low Shrubdepth Covers</th>
</tr>
</thead>
<tbody>
<tr>
<td>global</td>
<td>local</td>
</tr>
<tr>
<td>first-order types</td>
<td>quantifier alternation</td>
</tr>
</tbody>
</table>
Proof Ideas
A Cycle of Implications

- Structurally bounded expansion
- Bounded expansion
- Shrub decomposition
- Bounded expansion
- Lacon decomposition

Diagram showing the cycle of implications.
A Cycle of Implications

structurally bounded expansion

bounded expansion shrub decomposition

bounded expansion lacon decomposition
A Cycle of Implications

- Structurally bounded expansion
- Shrubs decomposition
- Bounded expansion
- Lacon decomposition
A Cycle of Implications

structurally bounded expansion

bounded expansion shrub decomposition

bounded expansion lacon decomposition
Step 1

structurally bounded expansion

bounded expansion shrub decomposition

bounded expansion lacon decomposition
Local Separators

radius 5 local separator
Local Separators

radius 5 local separator
Local Separators

radius 5 local separator
Local Feferman–Vaught

\[ G \models \varphi(u, v) \]

\[ s_1, s_2 \]
Local Feferman–Vaught

\[(f(\|\varphi\|) = q)\text{-local separator!}\]

\[G \models \varphi(u, v)\]

\[s_1 \quad s_2\]
Local Feferman–Vaught

\[ \text{type}_q(u, s_1, s_2) \]

\[ G \models \varphi(u, v) \]

\[ \text{type}_q(v, s_1, s_2) \]

\[
\text{type}_q(v, s_1, s_2) = \text{all } q\text{-formulas } \psi(x, y, z) \text{ with } G \models \psi(v, s_1, s_2)
\]
Local Feferman–Vaught

\[ \text{type}_q(u, s_1, s_2) \]

\[ G \models \varphi(u, v) \]

\[ \text{type}_q(v, s_1, s_2) \]

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

FV: \( \varphi(u, v) \) is determined by \( \text{type}_q(u, s_1, s_2) \) and \( \text{type}_q(v, s_1, s_2) \)
Local Feferman–Vaught

\[
type_q(u, s_1, s_2) = \tau_j
\]

\[
G \models \varphi(u, v) 
\]

\[
type_q(v, s_1, s_2) = \tau_i
\]

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

**FV:** \( \varphi(u, v) \) is determined by 
\( \text{type}_q(u, s_1, s_2) \) and \( \text{type}_q(v, s_1, s_2) \)
Local Feferman–Vaught

\[ \text{type}_q(u, s_1, s_2) = \tau_j \]

\[ \text{type}_q(v, s_1, s_2) = \tau_i \]

\[ G \models \varphi(u, v) \]

\[ G \models \psi(v, s_1, s_2) \]

**FV:** \( \varphi(u, v) \) is determined by \\
\( \text{type}_q(u, s_1, s_2) \) and \( \text{type}_q(v, s_1, s_2) \)
hidden vertices for local separator $s_1, s_2$
Constructing Lacon Decompositions
Constructing Lacon Decompositions
Constructing Lacon Decompositions
Constructing Lacon Decompositions
Constructing Lacon Decompositions
Constructing Lacon Decompositions
Constructing Lacon Decompositions

\[ G \models \varphi(u, v) \]
Constructing Lacon Decompositions

$G \models \varphi(u, v)$
Constructing Lacon Decompositions

\[ \tau_1, \tau_2 \quad \ldots \quad \tau_i, \tau_j \quad \ldots \]

\[ S_1 \quad S_2 \]

\[ v \quad w \]
Constructing Lacon Decompositions

no local separator!

$\tau_1, \tau_1 \ldots \tau_i, \tau_j \ldots$

$S_1 \quad S_2$

$v \quad w$
Constructing Lacon Decompositions

hidden vertices for local separator $s_1, s_2$

hidden vertices for local separator $s_3$
Constructing Lacon Decompositions

hidden vertices for local separator $s_1, s_2$

block of more general separator comes first
Constructing Lacon Decompositions

empty local separator
Constructing Lacon Decompositions
Constructing Lacon Decompositions

block for empty separator
Constructing Lacon Decompositions
Constructing Lacon Decompositions

block for empty separator
Constructing Lacon Decompositions

block for empty separator
Constructing Lacon Decompositions

block for empty separator

more general separator ≤ less general separator
Constructing Lacon Decompositions
Step 2

structurally bounded expansion

bounded expansion shrub decomposition

bounded expansion lacon 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

structured bounded expansion
bounded expansion
lacon decomposition
bounded expansion
shrub decomposition
Step 3

connect vertices with...
- distance 2
- distance 3 and same color
Step 3

connect vertices with...
  - distance 2
  - distance 3 and same color

\[ \varphi(x, y) := \]
\[ \text{dist}(x, y) = 2 \lor \]
\[ \text{dist}(x, y) = 3 \land c(x) = c(y) \]
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?
Big Question

Structurally
Nowhere Dense
Big Question

Has Nowhere Dense Decomposition?

Structurally Nowhere Dense
Thanks!