Decompositions and algorithms for interpretations of sparse graphs

Jakub Gajarský
University of Warsaw

The FO model checking problem

Input: finite graph G, sentence φ

Task: determine whether $G \models \varphi$

Example of a FO formula about graphs:

$$\varphi := \exists x_1 \exists x_2 \exists x_3 \forall y \,.\, (y = x_1) \lor (y = x_2) \lor (y = x_3) \lor E(y, x_1) \lor E(y, x_2) \lor E(y, x_3)$$

φ says that graph has dominating set of size at most 3

The FO model checking problem

Input: finite graph G, sentence φ

Task: determine whether $G \models \varphi$

Easy: runtime $n^{|\varphi|}$

Wanted: runtime $f(\varphi) \cdot n^c$ for fixed c (FPT)

The FO model checking problem

Input: finite graph G, sentence φ

Task: determine whether $G \models \varphi$

Easy: runtime $n^{|\varphi|}$

Wanted: runtime $f(\varphi) \cdot n^c$ for fixed c (FPT)

Not possible on the class of all finite graphs, but known to be true on many classes of "nice" graph classes, such as bounded degree graphs, planar graphs, …

The FO model checking problem

Input: finite graph G, sentence φ

Task: determine whether $G \models \varphi$

Easy: runtime $n^{|\varphi|}$

Wanted: runtime $f(\varphi) \cdot n^c$ for fixed c (FPT)

Motivation:

- Fundamental problem on its own
- Existence of a FPT algorithm implies the existence of FPT algorithms for many parameterized problems (independent set, dominating set, …)

The FO model checking problem

Basic results:

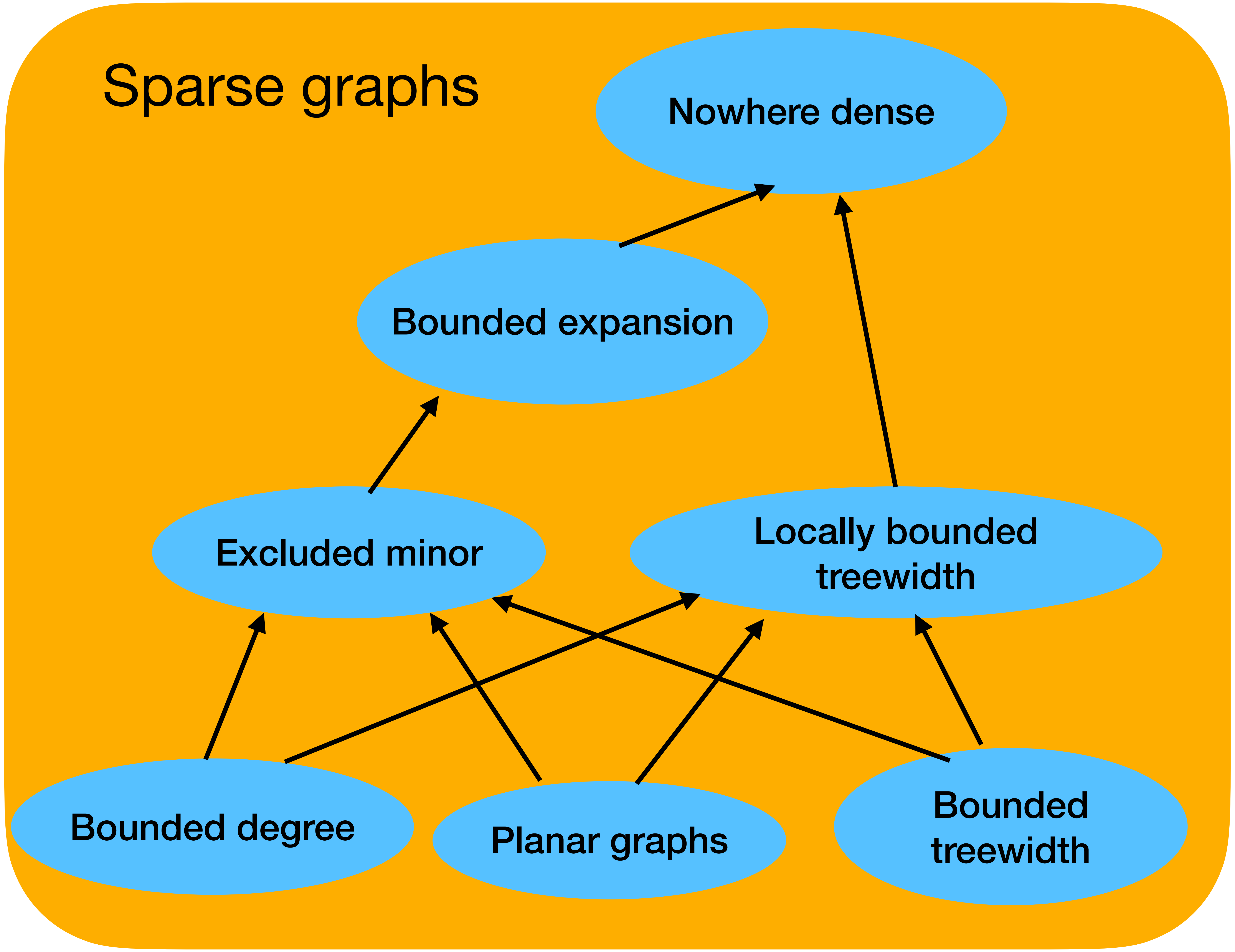
Theorem (Courcelle, 1990)
The FO model checking is solvable in time $f(\varphi) \cdot n^c$ for any class of graphs of bounded treewidth.

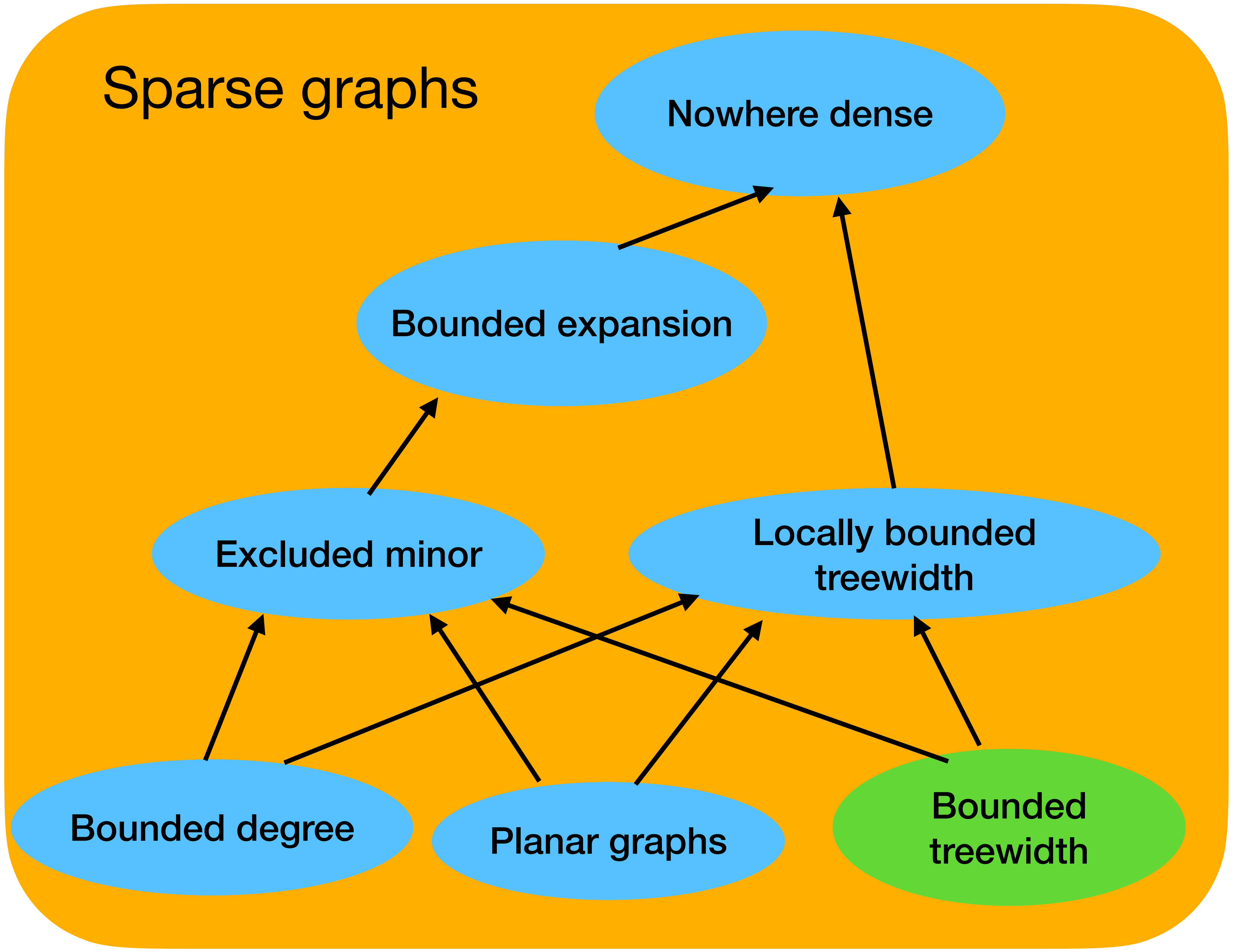
The FO model checking problem

Basic results:

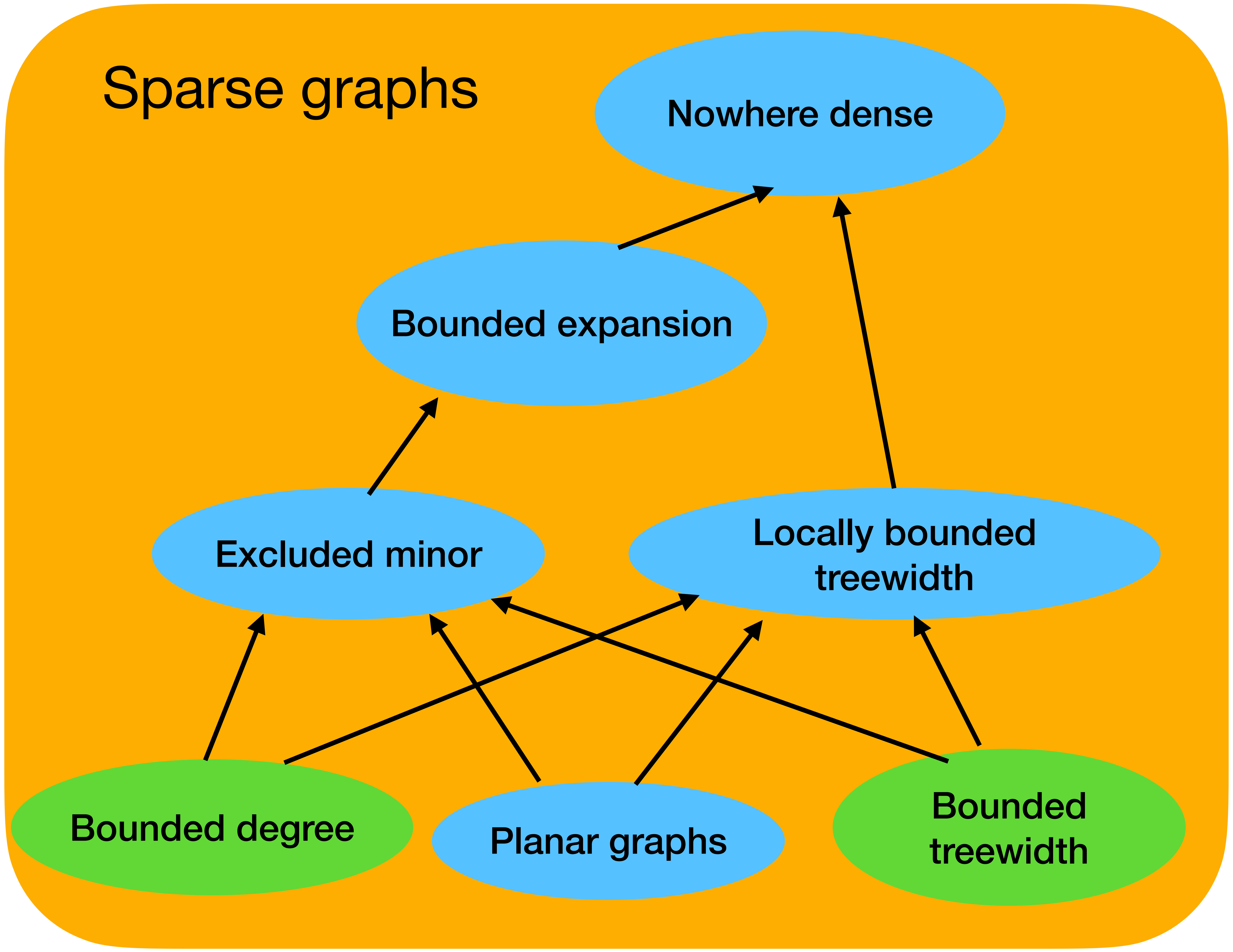
Theorem (Courcelle, 1990)
The FO model checking is solvable in time $f(\varphi) \cdot n^c$ for any class of graphs of bounded treewidth.

Theorem (Courcelle, Makowski, Rotics, 2000)
The FO model checking is solvable in time $f(\varphi) \cdot n^c$ for any class of graphs of bounded clique-width.


Sparse graphs
Nowhere dense
Bounded expansion
Excluded minor
Locally bounded treewidth
Bounded degree
Planar graphs
Bounded treewidth

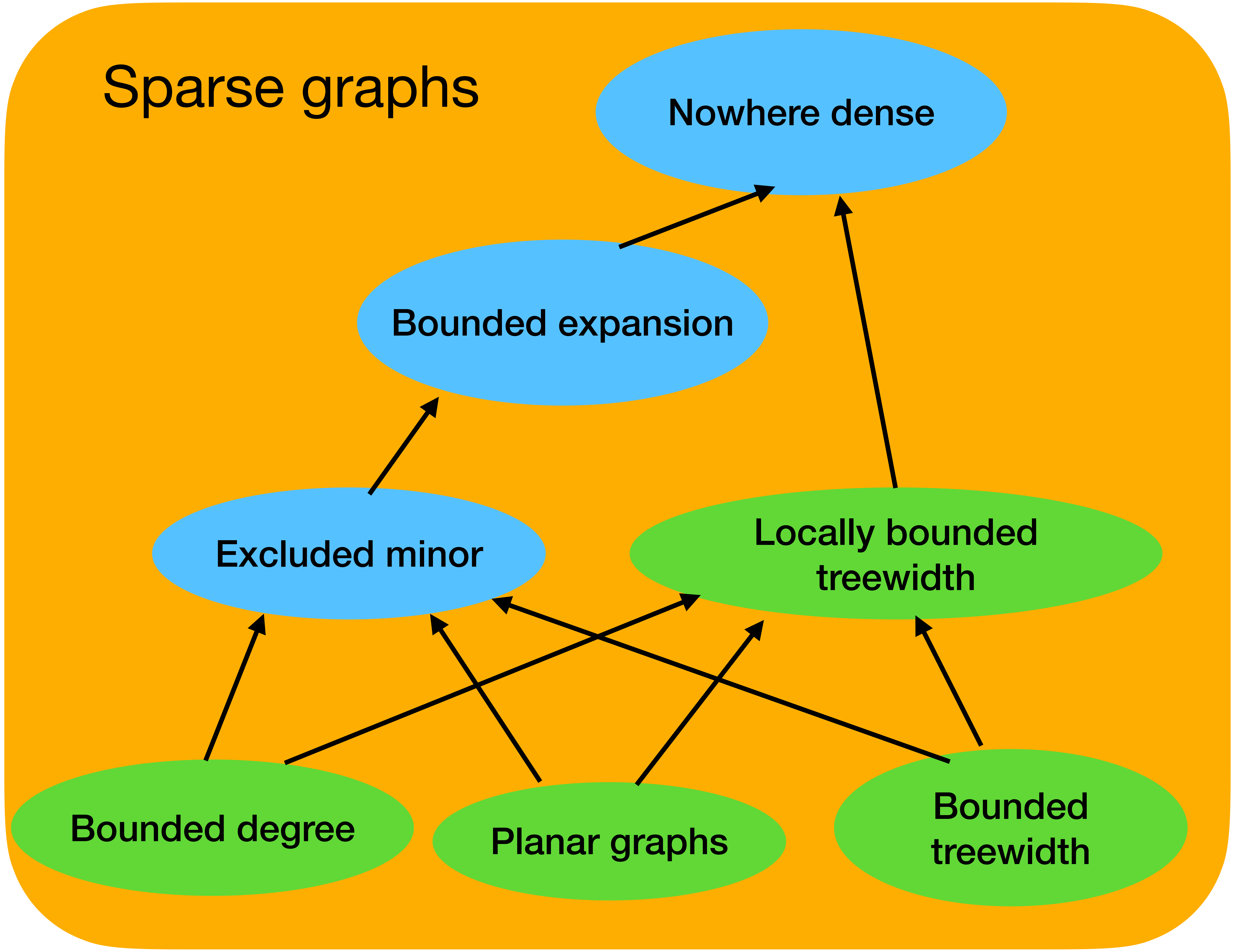


Courcelle, 1990



Seese, 1994

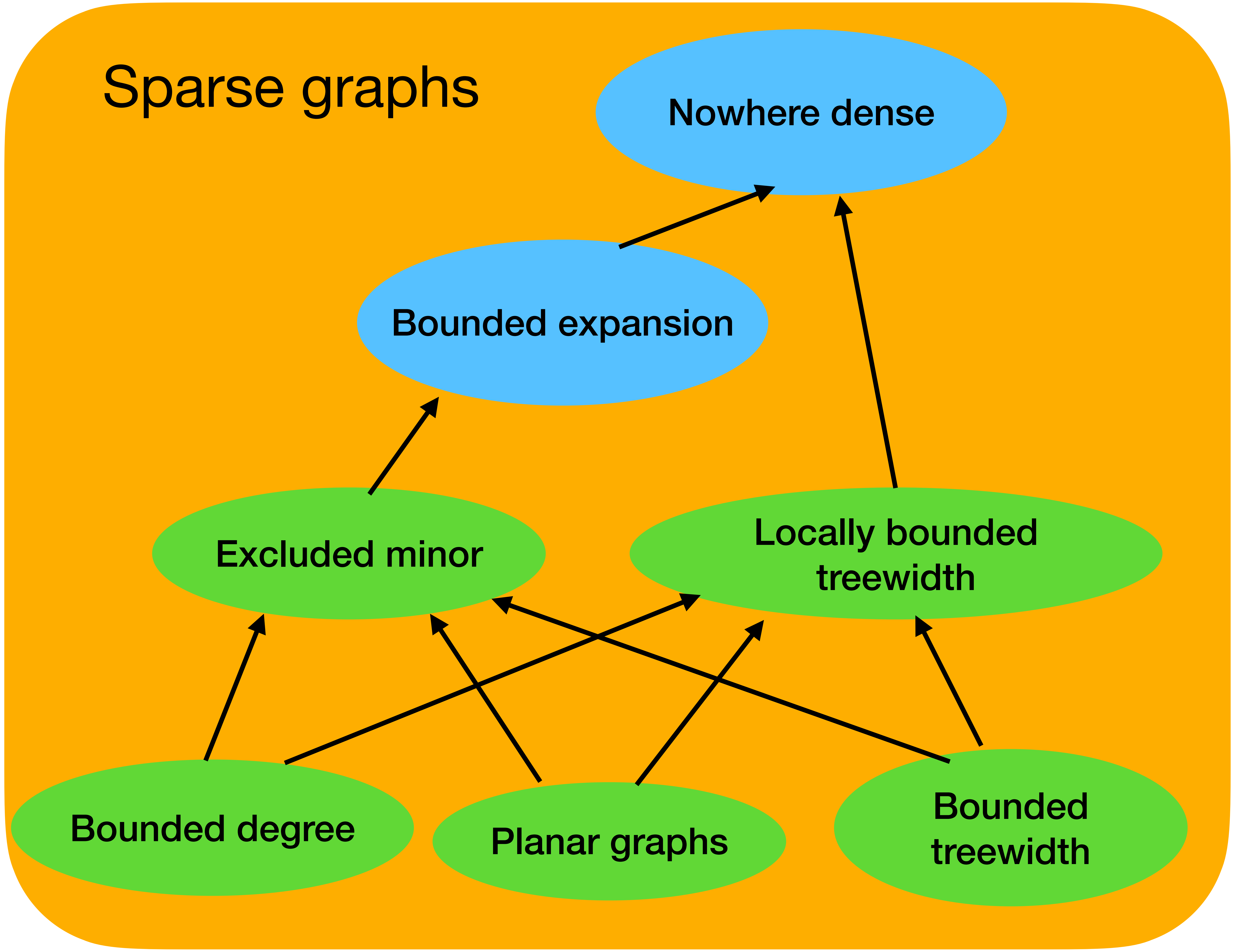
Courcelle, 1990



Frick, Grohe, 1999

Seese, 1994

Courcelle, 1990

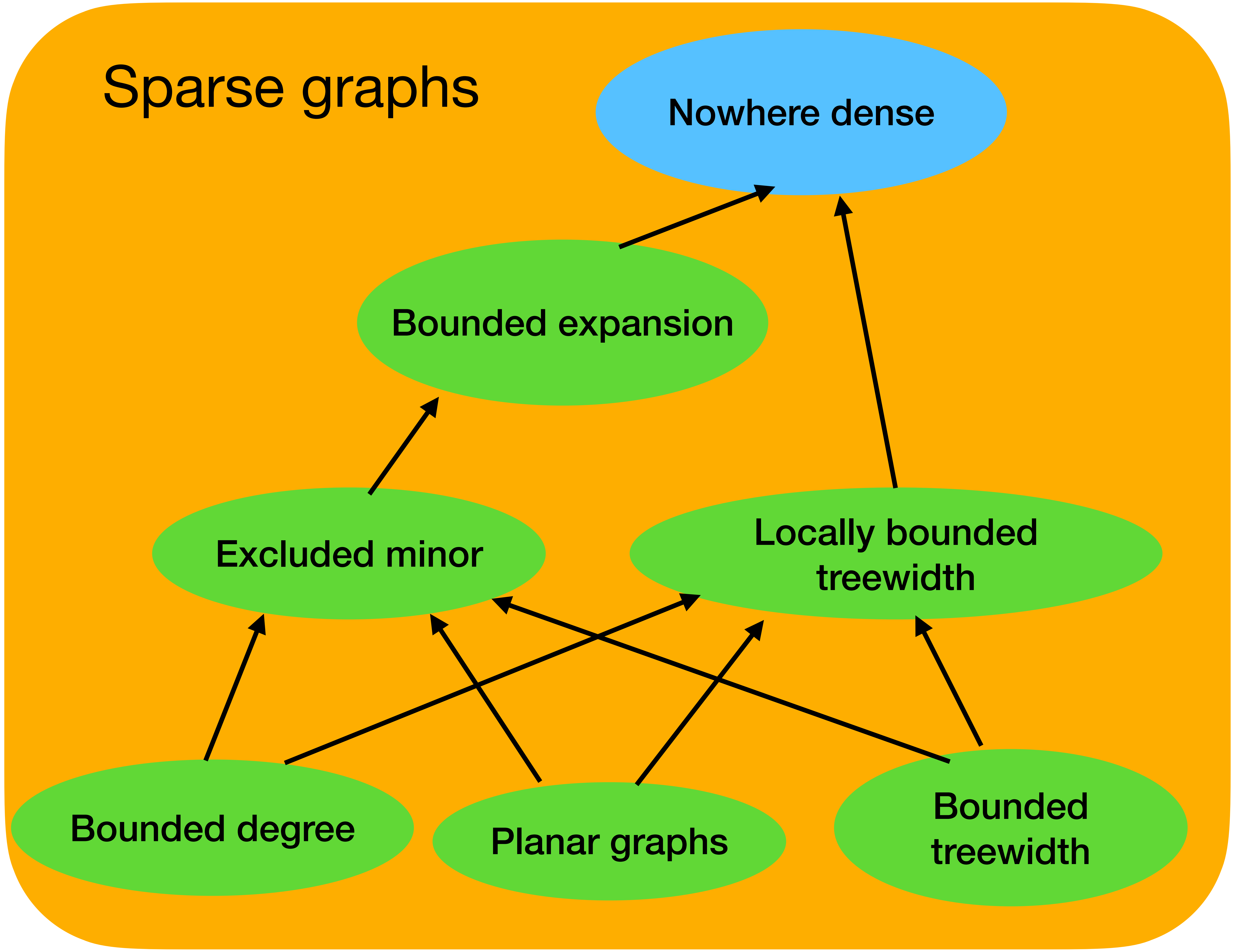


Flum, Grohe, 2001

Frick, Grohe, 1999

Seese, 1994

Courcelle, 1990



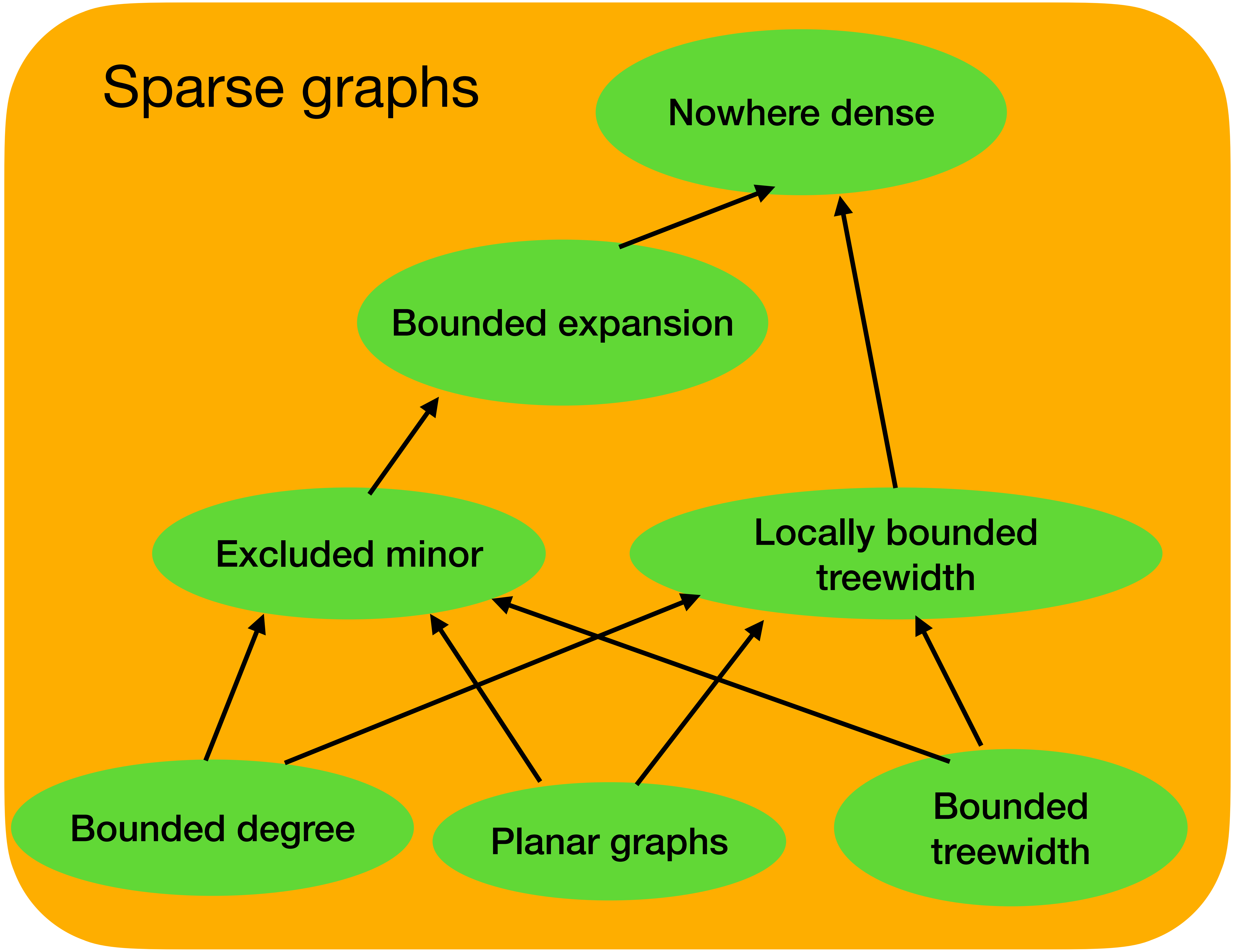
Dvořák, Král, Thomas, 2009

Flum, Grohe, 2001

Frick, Grohe, 1999

Seese, 1994

Courcelle, 1990



Grohe, Kreutzer, Siebertz, 2014

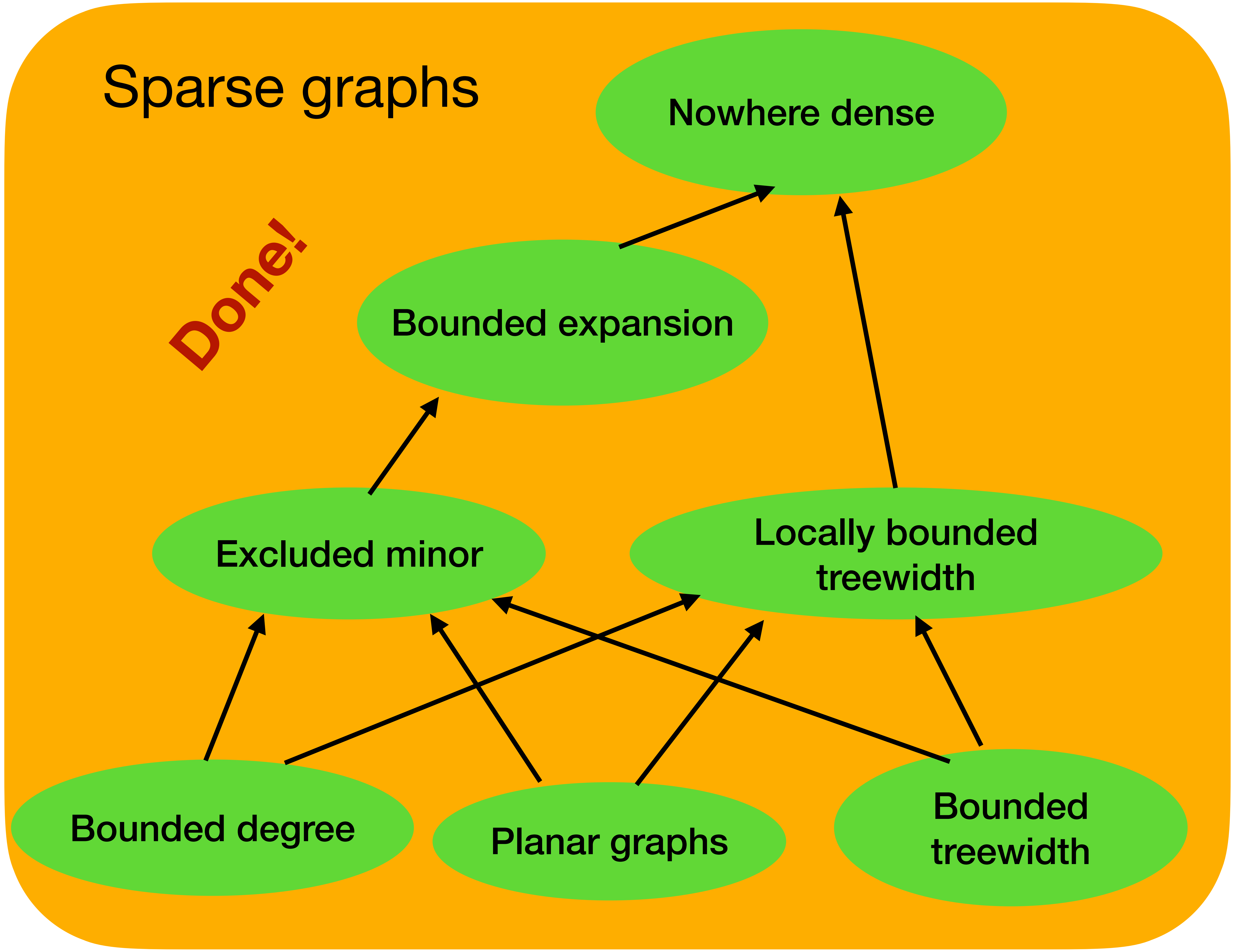
Dvořák, Král, Thomas, 2009

Flum, Grohe, 2001

Frick, Grohe, 1999

Seese, 1994

Courcelle, 1990



Grohe, Kreutzer, Siebertz, 2014

Dvořák, Král, Thomas, 2009

Flum, Grohe, 2001

Frick, Grohe, 1999

Seese, 1994

Courcelle, 1990

What to do next?

Nowhere denseness captures the notion of the graph class being ‘simple’ for sparse graphs.

What to do next?

Nowhere denseness captures the notion of the graph class being ‘simple’ for sparse graphs.

Next step: Generalize. Create a notion of simplicity for graphs which are not sparse. In particular, what about classes of dense graphs?

What to do next?

Nowhere denseness captures the notion of the graph class being ‘simple’ for sparse graphs.

Next step: Generalize. Create a notion of simplicity for graphs which are not sparse. In particular, what about classes of dense graphs?

Various classes of dense graphs have been studied in the literature, mainly based on intersection of geometric objects (interval graphs, string graphs, circle graphs, permutation graphs, …)

What to do next?

Nowhere denseness captures the notion of the graph class being ‘simple’ for sparse graphs.

Next step: Generalize. Create a notion of simplicity for graphs which are not sparse. In particular, what about classes of dense graphs?

Various classes of dense graphs have been studied in the literature, mainly based on intersection of geometric objects (interval graphs, string graphs, circle graphs, permutation graphs, …)

Problem: No apparent unifying structure, does not go well with logic.

What to do next?

Creating a notion of ‘simple’ graphs which includes dense graphs.

There are two basic options:

1. Create a theory ad hoc
2. Base a new theory on sparse graphs

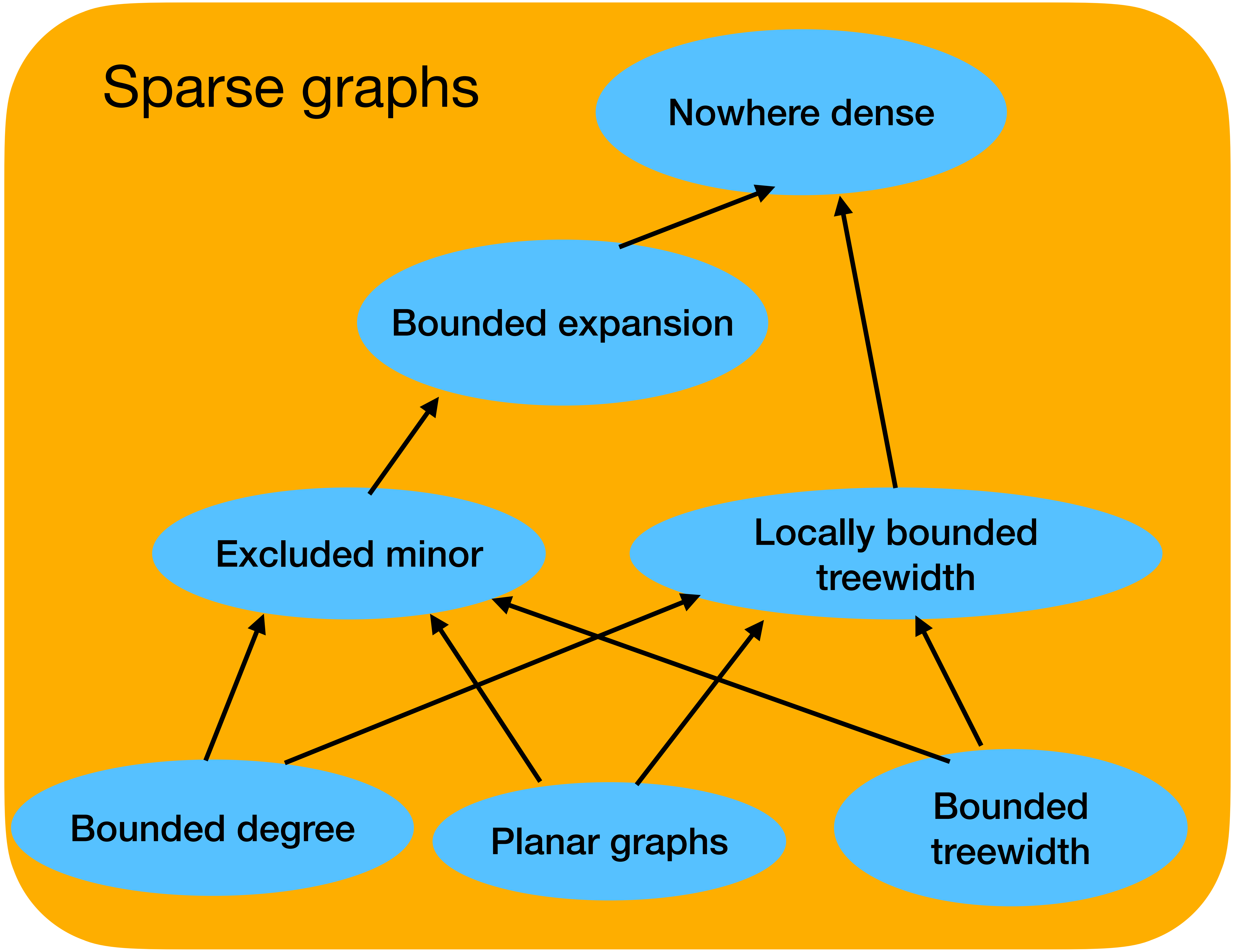
What to do next?

Creating a notion of ‘simple’ graphs which includes dense graphs.

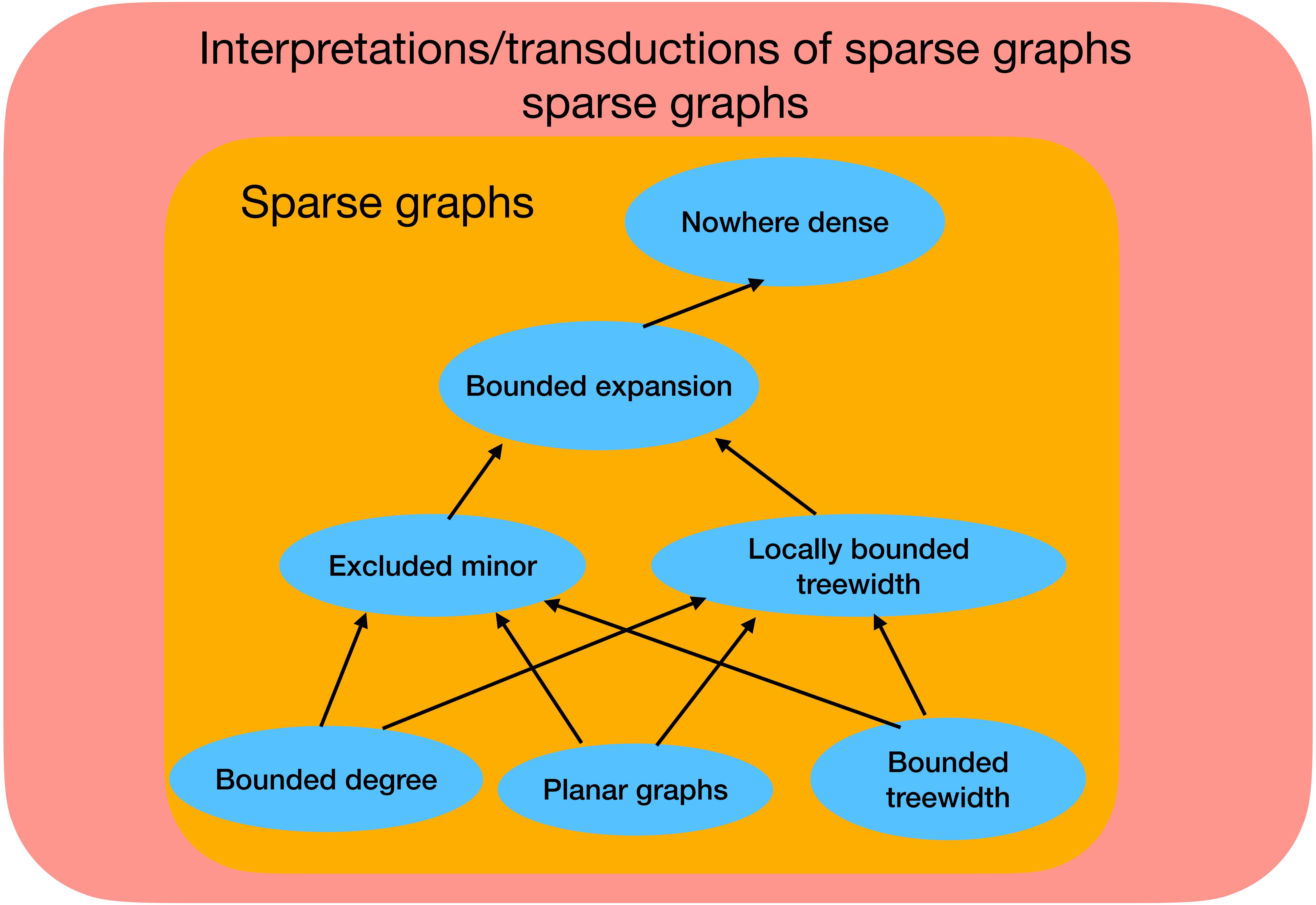
There are two basic options:

1. Create a theory ad hoc
2. Base a new theory on sparse graphs

Simple example - complements of planar graphs.



Sparse graphs
Nowhere dense
Bounded expansion
Excluded minor
Locally bounded treewidth
Bounded degree
Planar graphs
Bounded treewidth


Interpretations/transductions of sparse graphs
sparse graphs
Sparse graphs
Nowhere dense
Bounded expansion
Excluded minor
Locally bounded treewidth
Bounded degree
Planar graphs
Bounded treewidth

Interpretations (graph transformations)

- Start with a graph H
- Put an edge between vertices u,v if dist(u,v) > 5 and at least one of u,v has degree at least 4
- Call the resulting graph G

Interpretations (graph transformations)

- Start with a graph H
- Put an edge between vertices u,v if dist(u,v) > 5 and at least one of u,v has degree at least 4
- Call the resulting graph G

One can write the condition “dist(u,v) > 5 and at least one of u,v has degree at least 4” in FO logic:

$$\psi(x, y) := dist(u, v) < 5 \wedge (deg(u) \geq 4 \vee deg(v) \leq 4)$$

Interpretations (graph transformations)

- Start with a graph H
- Put an edge between vertices u,v if dist(u,v) > 5 and at least one of u,v has degree at least 4
- Call the resulting graph G

One can write the condition “dist(u,v) > 5 and at least one of u,v has degree at least 4” in FO logic:

$$\psi(x, y) := dist(u, v) < 5 \wedge (deg(u) \geq 4 \vee deg(v) \leq 4)$$

In this case the transformation given by $\psi(x, y)$ is called an interpretation

Interpretations (graph transformations)

Examples — put an edge between u and v if:

- There is no edge between u and v (complementation).
- They are at distance at most 2 (squaring).
- There are exactly two paths of length 17 between u and v and on one of them there is a vertex which is in a triangle.

Interpretations (graph transformations)

Examples — put an edge between u and v if:

- There is no edge between u and v (complementation).
- They are at distance at most 2 (squaring).
- There are exactly two paths of length 17 between u and v and on one of them there is a vertex which is in a triangle.

Notation: $G = \psi(H)$

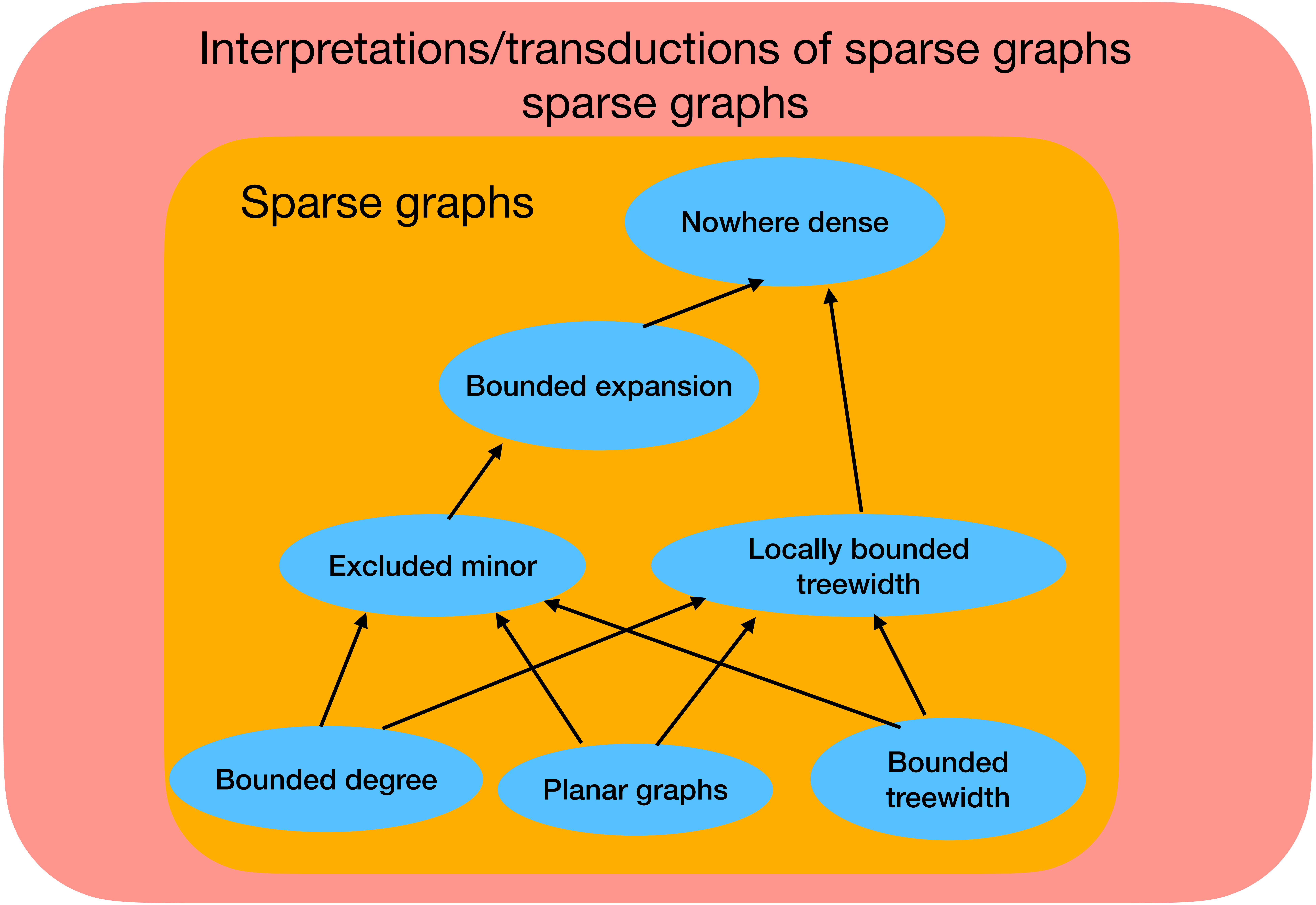
Extends to graph classes: $\mathscr{D} = \psi(\mathscr{C}) = \{\psi(H) \mid H \in \mathscr{C}\}$
(component-wise)

Research program

Let $\mathscr{D}$ be graph class interpretable in a sparse graph class $\mathscr{C}$.

Question: Is FO model checking in FPT on $\mathscr{D}$?

Conjecture: The answer is always YES


Interpretations/transductions of sparse graphs
sparse graphs
Sparse graphs
Nowhere dense
Bounded expansion
Excluded minor
Locally bounded treewidth
Bounded degree
Planar graphs
Bounded treewidth

Research program

Let $\mathscr{D}$ be graph class interpretable in a sparse graph class $\mathscr{C}$.

Question: Is FO model checking in FPT on $\mathscr{D}$?

Conjecture: The answer is always YES

Known:

- Interpretations of graphs of bounded treewidth
- Interpretations of graphs of bounded degree (LICS 2016)
- Interpretations of planar graphs, and more generally graphs of locally bounded treewidth and even more generally graph classes of locally bounded clique-width (LICS 2022)

Research program

Let $\mathscr{D}$ be graph class interpretable in a sparse graph class $\mathscr{C}$.

Question: Is FO model checking in FPT on $\mathscr{D}$?

Conjecture: The answer is always YES

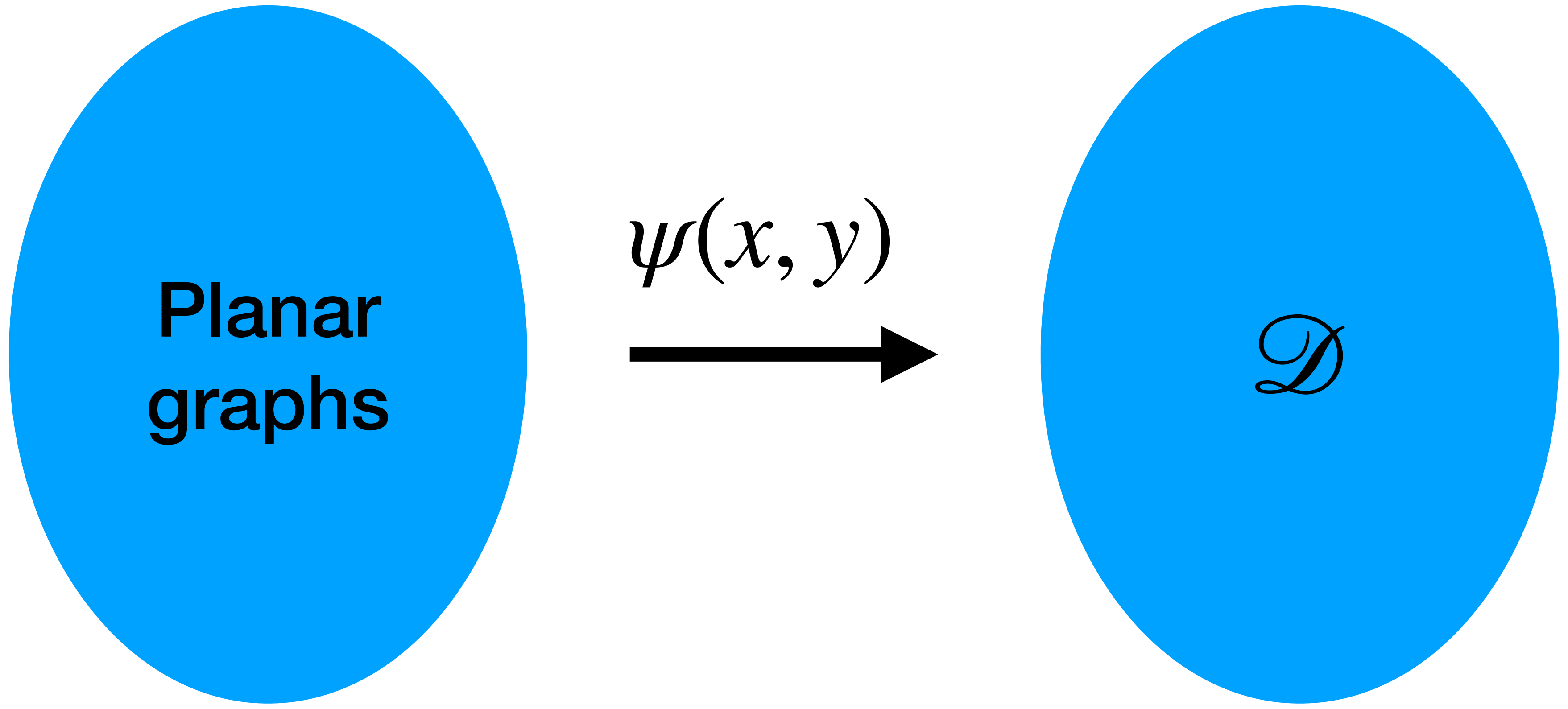
Why should/could something like this hold?

Why could the conjecture hold?

Setting: Take any sparse graph class $\mathscr{C}$ and any interpretation formula $\psi(x, y)$. Consider $\mathscr{D} = \psi(\mathscr{C})$.

Why could the conjecture hold?

Setting: Take any sparse graph class $\mathscr{C}$ and any interpretation formula $\psi(x, y)$. Consider $\mathscr{D} = \psi(\mathscr{C})$.

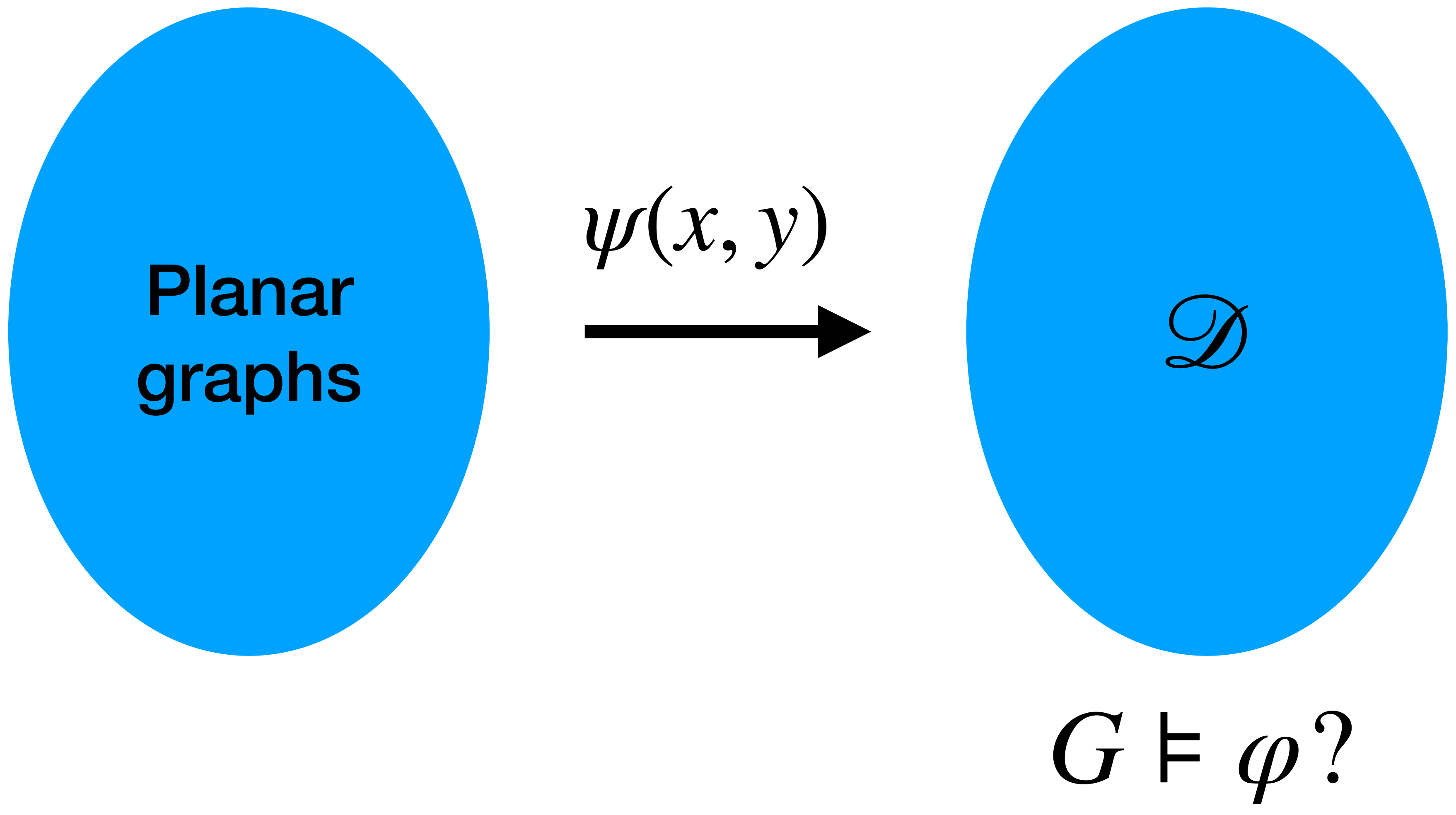


For concreteness:
$\mathscr{C}$ = class of all planar graphs
$\psi(x, y)$ = x and y are at distance at most 4
$\mathscr{D} = \psi(\mathscr{C})$

Why could the conjecture hold?

Setting: Take any sparse graph class $\mathscr{C}$ and any interpretation formula $\psi(x, y)$. Consider $\mathscr{D} = \psi(\mathscr{C})$.

For concreteness:
$\mathscr{C}$ = class of all planar graphs
$\psi(x, y)$ = x and y are at distance at most 4
$\mathscr{D} = \psi(\mathscr{C})$

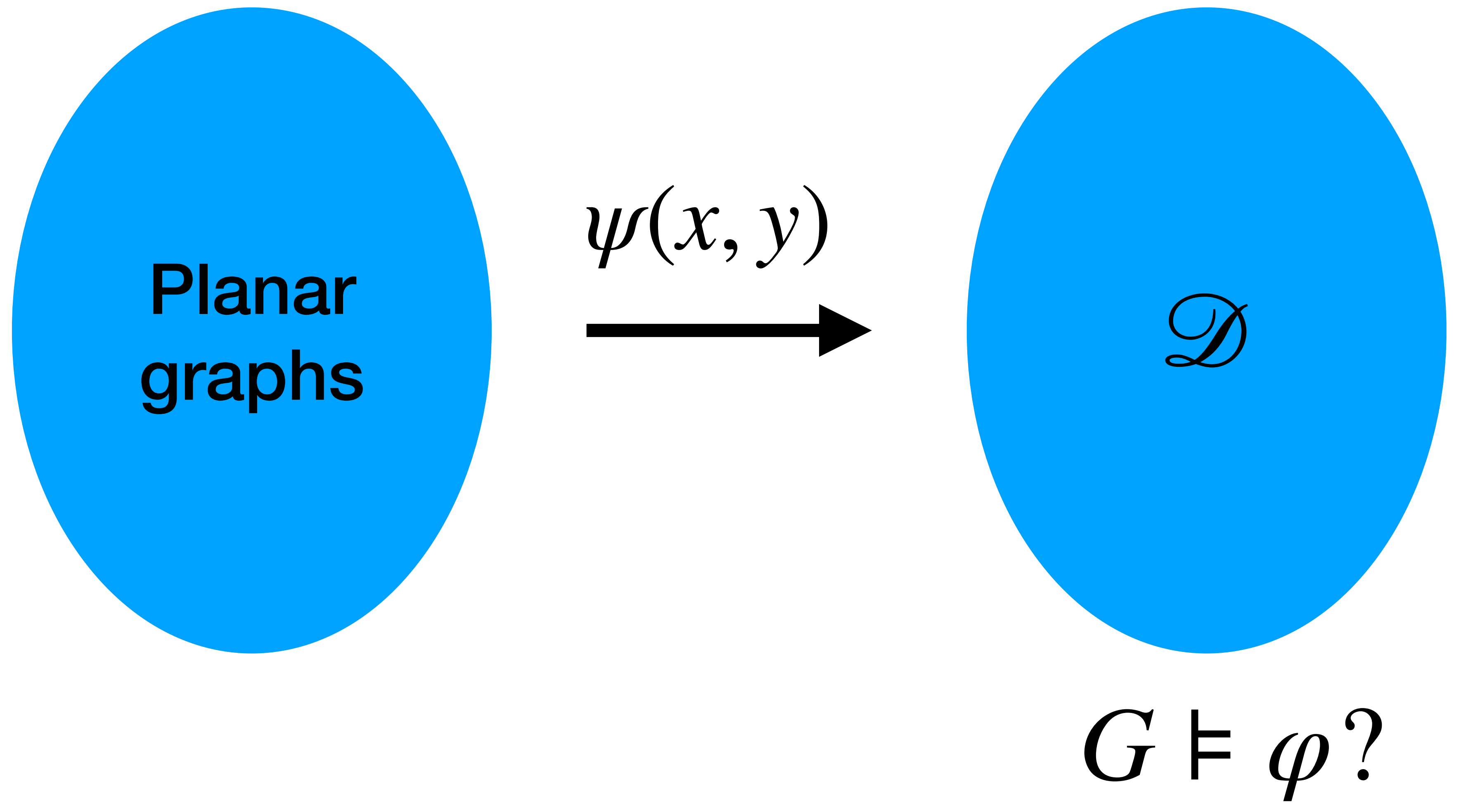


Task: FO model checking on $\mathscr{D}$ — given $G \in \mathscr{D}$ and φ, determine whether $G \models \varphi$.

Why could the conjecture hold?

Setting: Take any sparse graph class $\mathscr{C}$ and any interpretation formula $\psi(x, y)$. Consider $\mathscr{D} = \psi(\mathscr{C})$.

For concreteness:
$\mathscr{C}$ = class of all planar graphs
$\psi(x, y)$ = x and y are at distance at most 4
$\mathscr{D} = \psi(\mathscr{C})$



Task: FO model checking on $\mathscr{D}$ – given $G \in \mathscr{D}$ and φ, determine whether $G \models \varphi$.

For concreteness, let φ express "G has a dominating set of size 3."

Why could the conjecture hold?

$\mathscr{C}$ = class of all planar graphs

$\psi(x, y)$ = x and y are at distance at most 4

$\mathscr{D} = \psi(\mathscr{C})$

Task: given $G \in \mathscr{D}$ and φ, determine whether $G \models \varphi$.

$\varphi :=$ "G has a dominating set of size 3."

G — any graph from $\mathscr{D}$

$G \models \varphi$

Why could the conjecture hold?

$\mathscr{C}$ = class of all planar graphs

$\psi(x, y)$ = x and y are at distance at most 4

$\mathscr{D} = \psi(\mathscr{C})$

Task: given $G \in \mathscr{D}$ and φ, determine whether $G \models \varphi$.

$\varphi :=$ "G has a dominating set of size 3."

G — any graph from $\mathscr{D}$

$G \models \varphi$

G has three vertices such that every vertex is a neighbor of one of them

Why could the conjecture hold?

$\mathscr{C}$ = class of all planar graphs

$\psi(x, y)$ = x and y are at distance at most 4

$\mathscr{D} = \psi(\mathscr{C})$

Task: given $G \in \mathscr{D}$ and φ, determine whether $G \models \varphi$.

φ := "G has a dominating set of size 3."

G – any graph from $\mathscr{D}$

$$H \xrightarrow{\psi} G$$

$G \models \varphi$

G has three vertices such that every vertex is a neighbor of one of them

Why could the conjecture hold?

$\mathscr{C}$ = class of all planar graphs

$\psi(x, y)$ = x and y are at distance at most 4

$\mathscr{D} = \psi(\mathscr{C})$

Task: given $G \in \mathscr{D}$ and φ, determine whether $G \models \varphi$.

$\varphi :=$ "G has a dominating set of size 3."

G – any graph from $\mathscr{D}$

$H \xrightarrow{\psi} G$

H $\qquad$ $G \models \varphi$

H has three vertices such that every vertex at distance at most 4 from one of them $\iff$ G has three vertices such that every vertex is a neighbor of one of them

Why could the conjecture hold?

$\mathscr{C}$ = class of all planar graphs

$\psi(x, y) = x$ and y are at distance at most 4

$\mathscr{D} = \psi(\mathscr{C})$

Task: given $G \in \mathscr{D}$ and φ, determine whether $G \vDash \varphi$.

$\varphi :=$ "G has a dominating set of size 3."

G – any graph from $\mathscr{D}$

$$H \xrightarrow{\psi} G$$

$H \vDash \varphi'$ $\iff$ $G \vDash \varphi$

H has three vertices such that every vertex at distance at most 4 from one of them $\iff$ G has three vertices such that every vertex is a neighbor of one of them

Why could the conjecture hold?

In $\mathscr{C}$ | In $\mathscr{D}$

$$H \xrightarrow{\psi} G$$

$H \vDash \varphi'$ $\iff$ $G \vDash \varphi$

H has three vertices such that every vertex at distance at most 4 from one of them $\iff$ G has three vertices such that every vertex is a neighbor of one of them

General rule: FO-definable propeties about G translate to FO-definable properties about H

Why could the conjecture hold?

In $\mathscr{C}$ — In $\mathscr{D}$

$$H \xrightarrow{\psi} G$$

$H \models \varphi'$ $\iff$ $G \models \varphi$

H has three vertices such that every vertex at distance at most 4 from one of them $\iff$ G has three vertices such that every vertex is a neighbor of one of them

General rule: FO-definable propeties about G translate to FO-definable properties about H

Consequence: FO model checking in $\mathscr{D}$ reduces to FO model checking in $\mathscr{C}$…

Why could the conjecture hold?

In $\mathscr{C}$ $\qquad$ In $\mathscr{D}$

$$H \xrightarrow{\psi} G$$

$H \models \varphi'$ $\iff$ $G \models \varphi$

H has three vertices such that every vertex at distance at most 4 from one of them $\iff$ G has three vertices such that every vertex is a neighbor of one of them

General rule: FO-definable propeties about G translate to FO-definable properties about H

Consequence: FO model checking in $\mathscr{D}$ reduces to FO model checking in $\mathscr{C}$…

…provided that we have access to H.

Why could the conjecture hold?

In $\mathscr{C}$ | In $\mathscr{D}$

$$H \xrightarrow{\psi} G$$

$H \models \varphi'$ $\iff$ $G \models \varphi$

H has three vertices such that every vertex at distance at most 4 from one of them $\iff$ G has three vertices such that every vertex is a neighbor of one of them

General rule: FO-definable propeties about G translate to FO-definable properties about H

Consequence: FO model checking in $\mathscr{D}$ reduces to FO model checking in $\mathscr{C}$…

…provided that we have access to H.

Finding H efficiently — the key algorithmic problem

The interpretation reversal problem

Setting: Take any sparse graph class $\mathscr{C}$ and any interpretation formula $\psi(x, y)$. Consider $\mathscr{D} = \psi(\mathscr{C})$.

Task: Find a polynomial algorithm which, given $G \in \mathscr{D}$ computes $H \in \mathscr{C}$ such that $G = \psi(H)$

$$H \xrightarrow{\quad\quad\psi\quad\quad} G$$

The interpretation reversal problem

Setting: Take any sparse graph class $\mathscr{C}$ and any interpretation formula $\psi(x, y)$. Consider $\mathscr{D} = \psi(\mathscr{C})$.

Task: Find a polynomial algorithm which, given $G \in \mathscr{D}$ computes $H \in \mathscr{C}$ such that $G = \psi(H)$

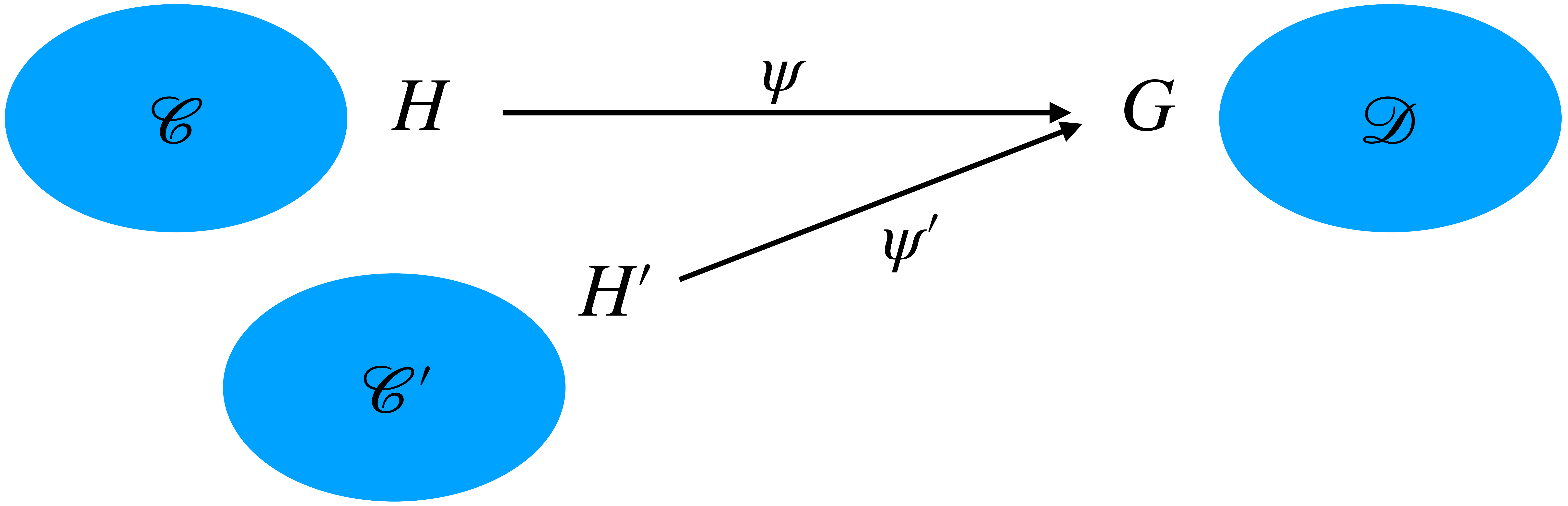
$$H \xrightarrow{\quad\psi\quad} G$$

We do not expect this to work.

Approximate interpretation reversal problem

Setting: Take any sparse graph class $\mathscr{C}$ and any interpretation formula $\psi(x, y)$. Consider $\mathscr{D} = \psi(\mathscr{C})$.

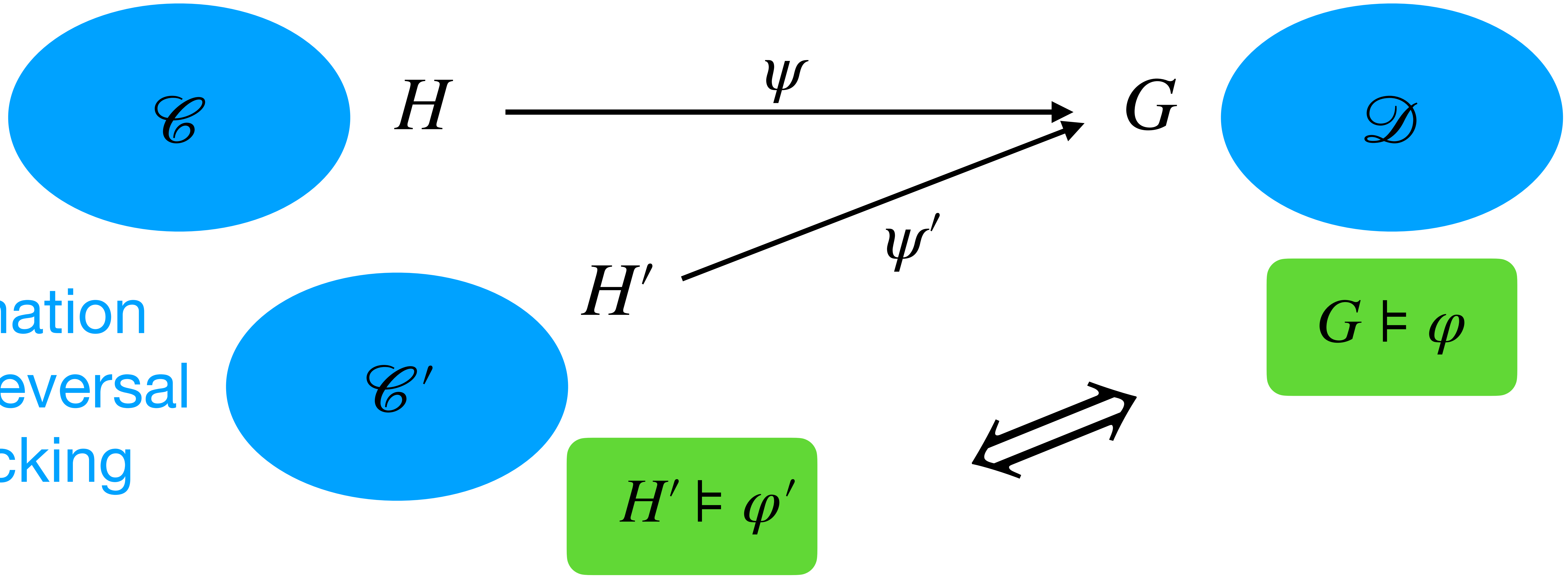
Task: Find a sparse graph class $\mathscr{C}'$, interpretation formula $\psi'(x, y)$ and a polynomial algorithm which given $G \in \mathscr{D}$ computes $H' \in \mathscr{C}$ such that $G = \psi'(H)$



Approximate interpretation reversal problem

Setting: Take any sparse graph class $\mathscr{C}$ and any interpretation formula $\psi(x, y)$. Consider $\mathscr{D} = \psi(\mathscr{C})$.

Task: Find a sparse graph class $\mathscr{C}'$, interpretation formula $\psi'(x, y)$ and a polynomial algorithm which given $G \in \mathscr{D}$ computes $H' \in \mathscr{C}$ such that $G = \psi'(H)$

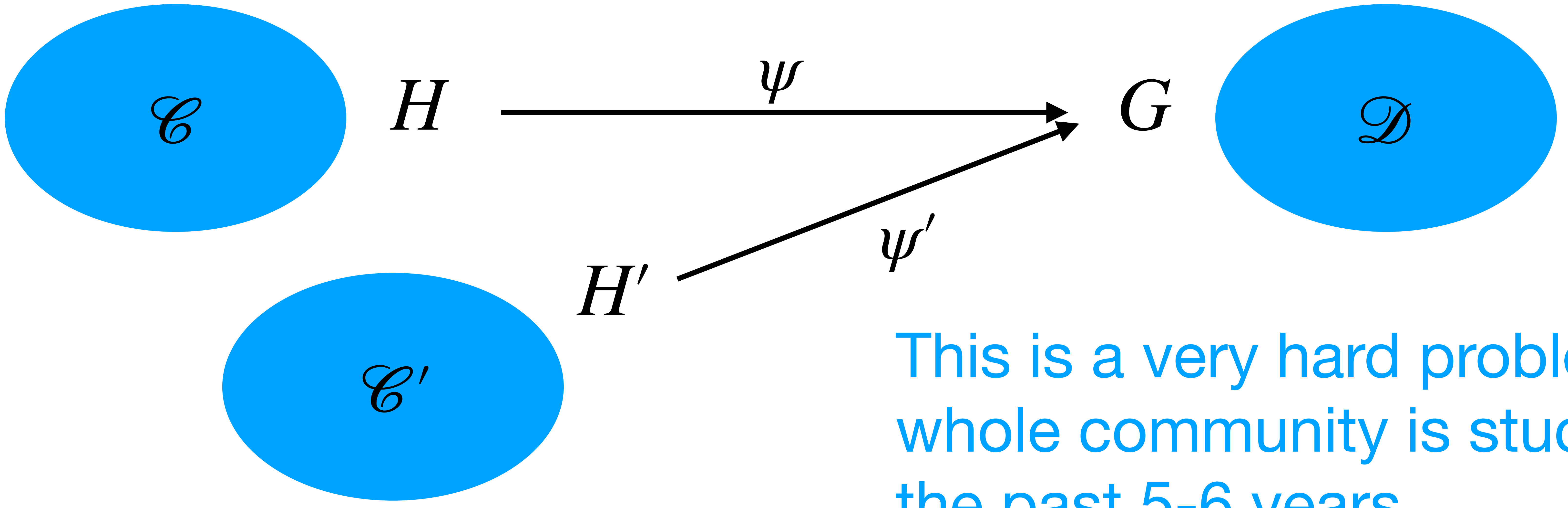


From approximation interpretation reversal the model checking follows.

Approximate interpretation reversal problem

Setting: Take any sparse graph class $\mathscr{C}$ and any interpretation formula $\psi(x, y)$. Consider $\mathscr{D} = \psi(\mathscr{C})$.

Task: Find a sparse graph class $\mathscr{C}'$, interpretation formula $\psi'(x, y)$ and a polynomial algorithm which given $G \in \mathscr{D}$ computes $H' \in \mathscr{C}$ such that $G = \psi'(H)$



This is a very hard problem, and the whole community is stuck on that for the past 5-6 years.

What we know?

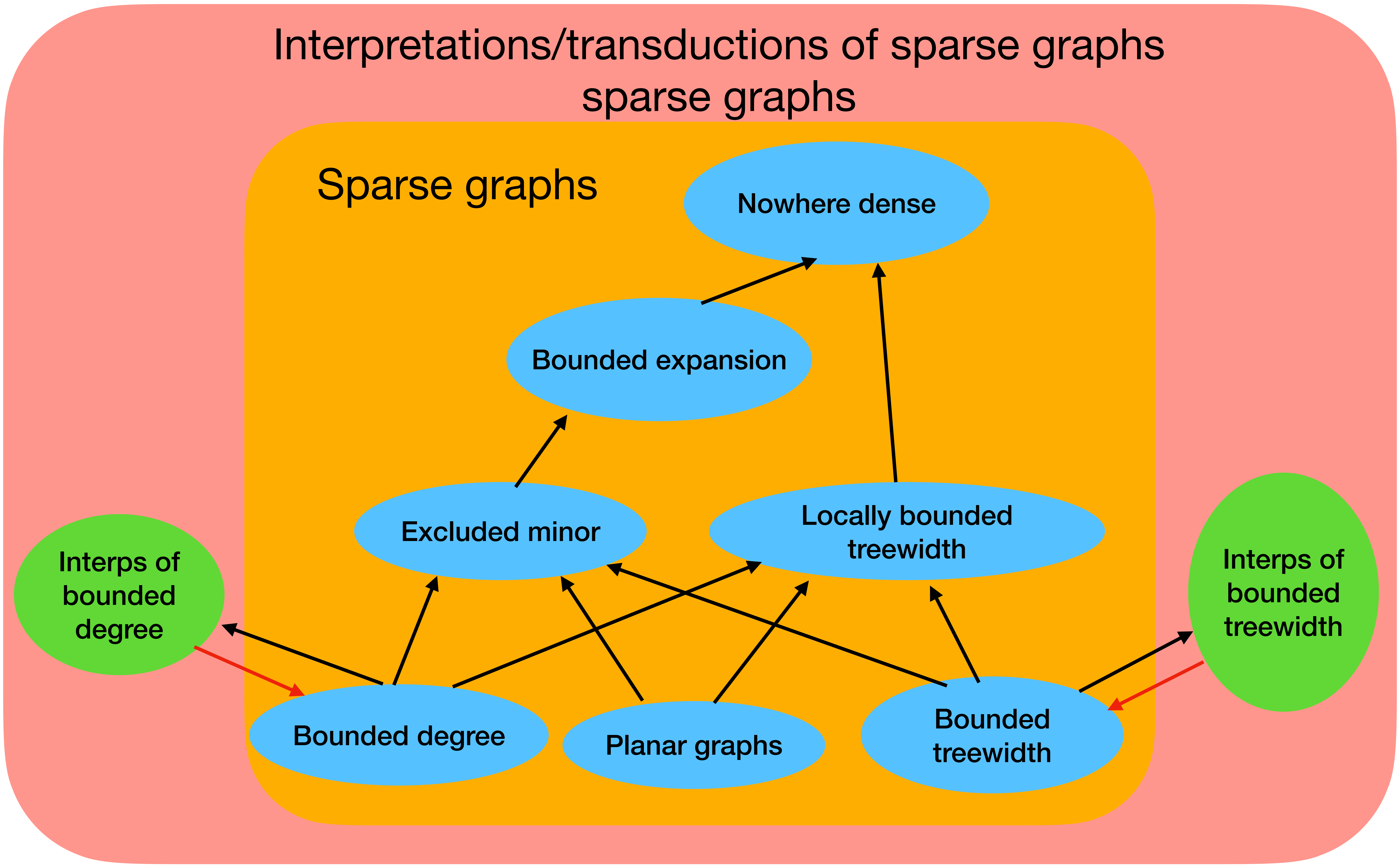
We know how to do this:

- When $\mathscr{C}$ is a class of bounded degree (G., Hliněný, Lokshtanov, Ramanujan, Obdržálek; LICS 2016)
  (Here $\mathscr{C}'$ has also bounded degree, but larger than $\mathscr{C}$)

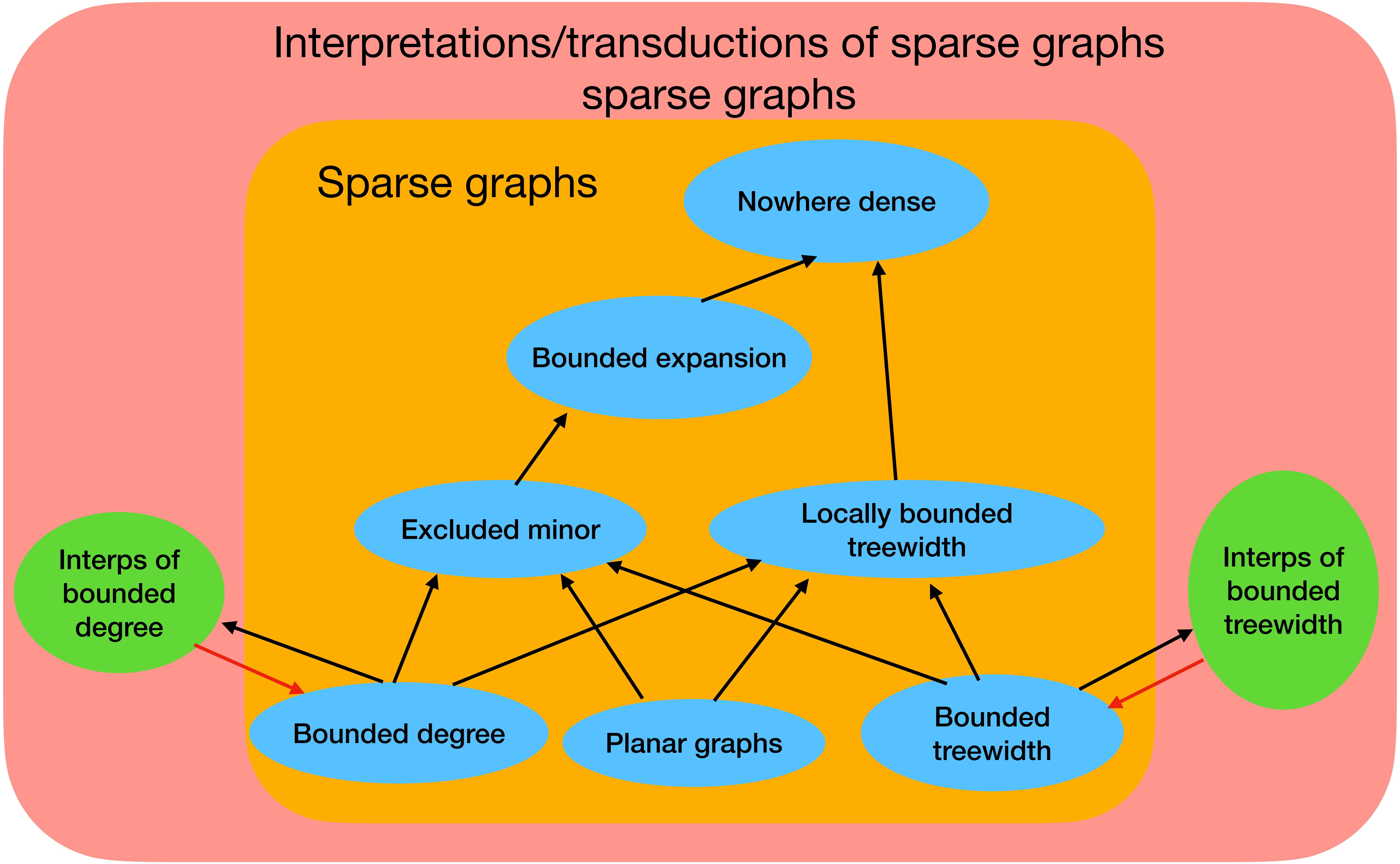
- When $\mathscr{C}$ is a class of bounded pathwidth (Nešetřil, Ossona de Mendez, Rabinovich, Siebertz; SODA 2020)
  (Here $\mathscr{C}'$ has also bounded pathwidth, but larger than $\mathscr{C}$)

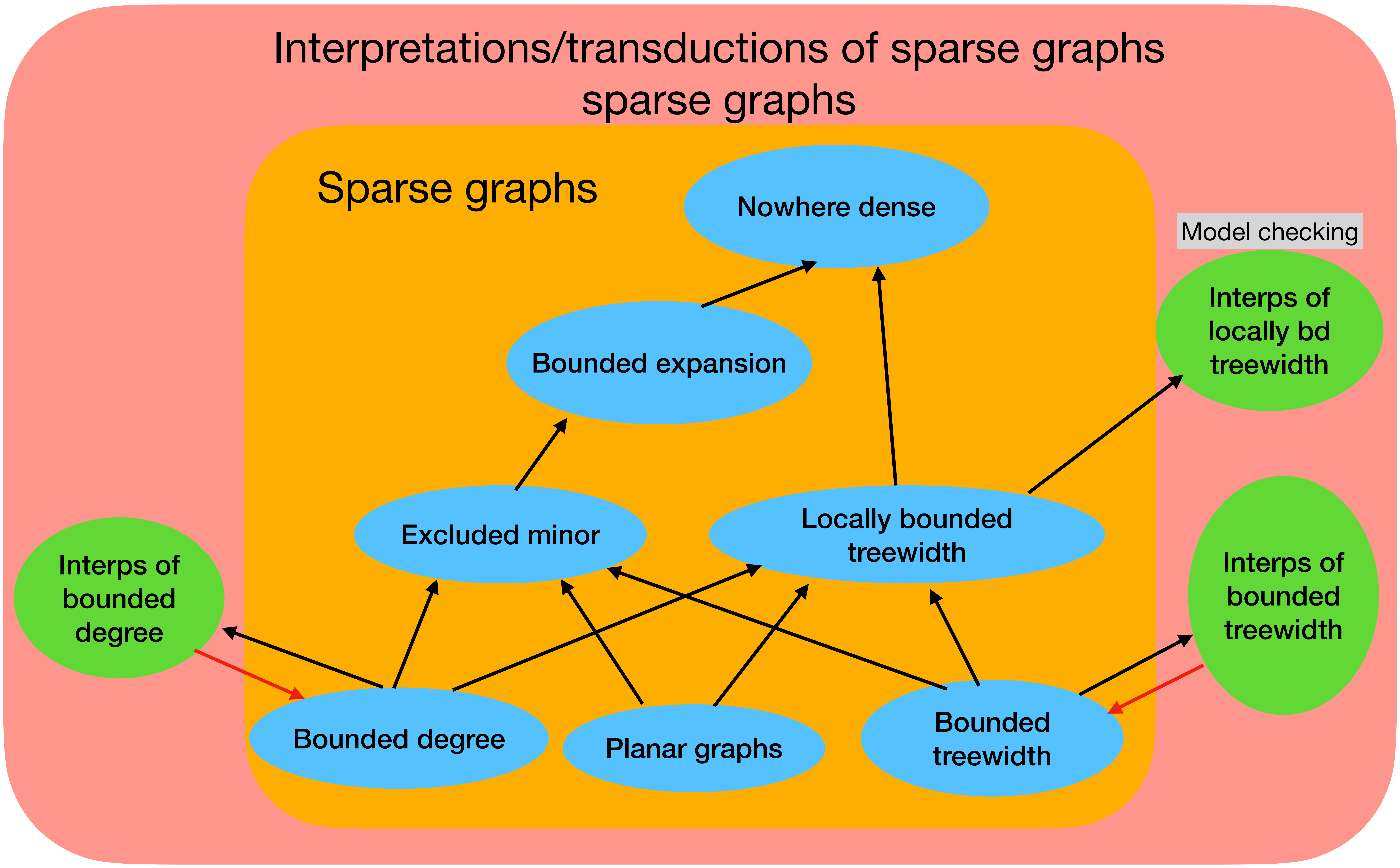
- When $\mathscr{C}$ is a class of bounded treewidth (Nešetřil, Ossona de Mendez, Mi. Pilipczuk, Rabinovich, Siebertz; SODA 2021)
  (Here $\mathscr{C}'$ has also bounded pathwidth, but larger than $\mathscr{C}$)

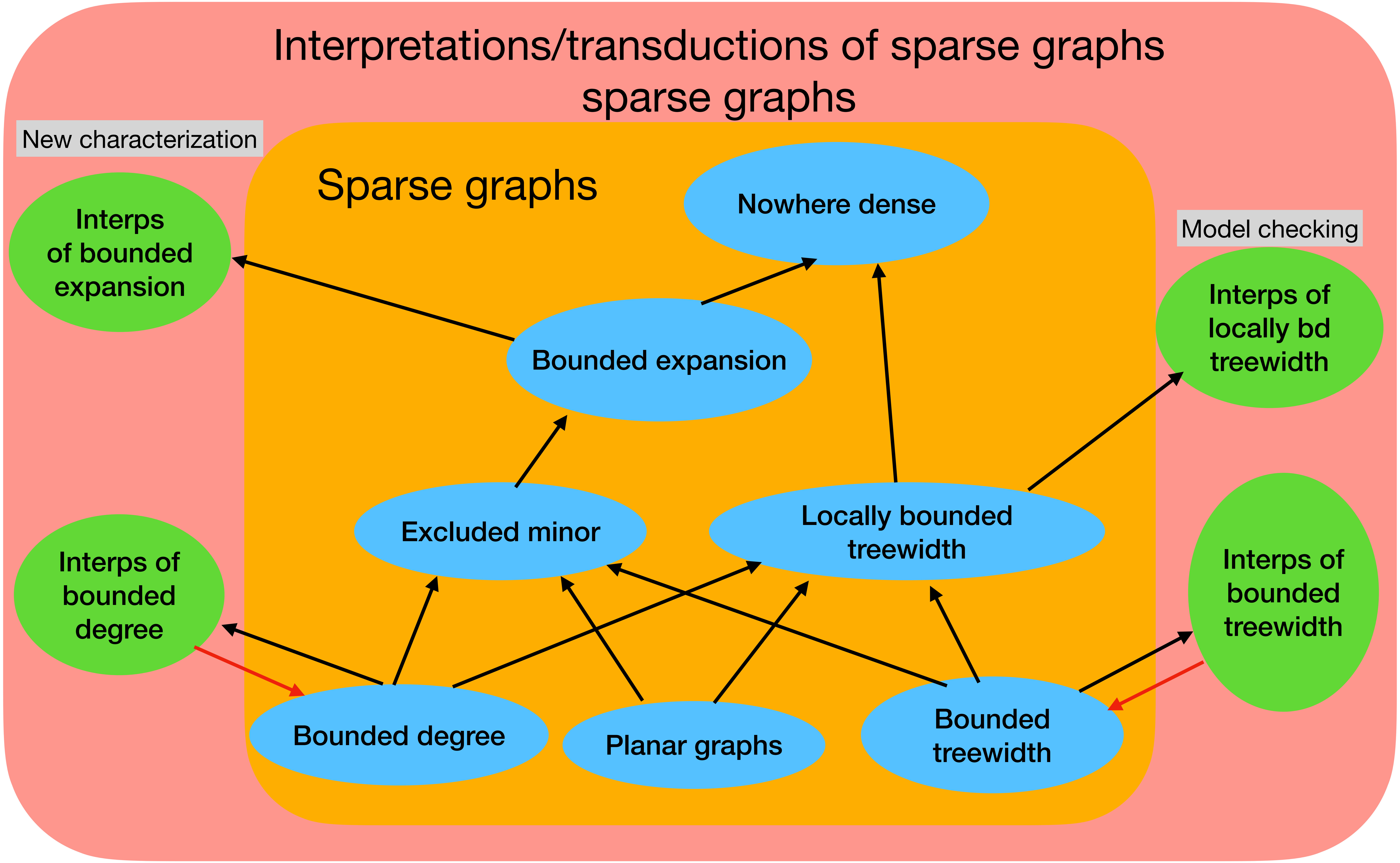
Everything else is open.

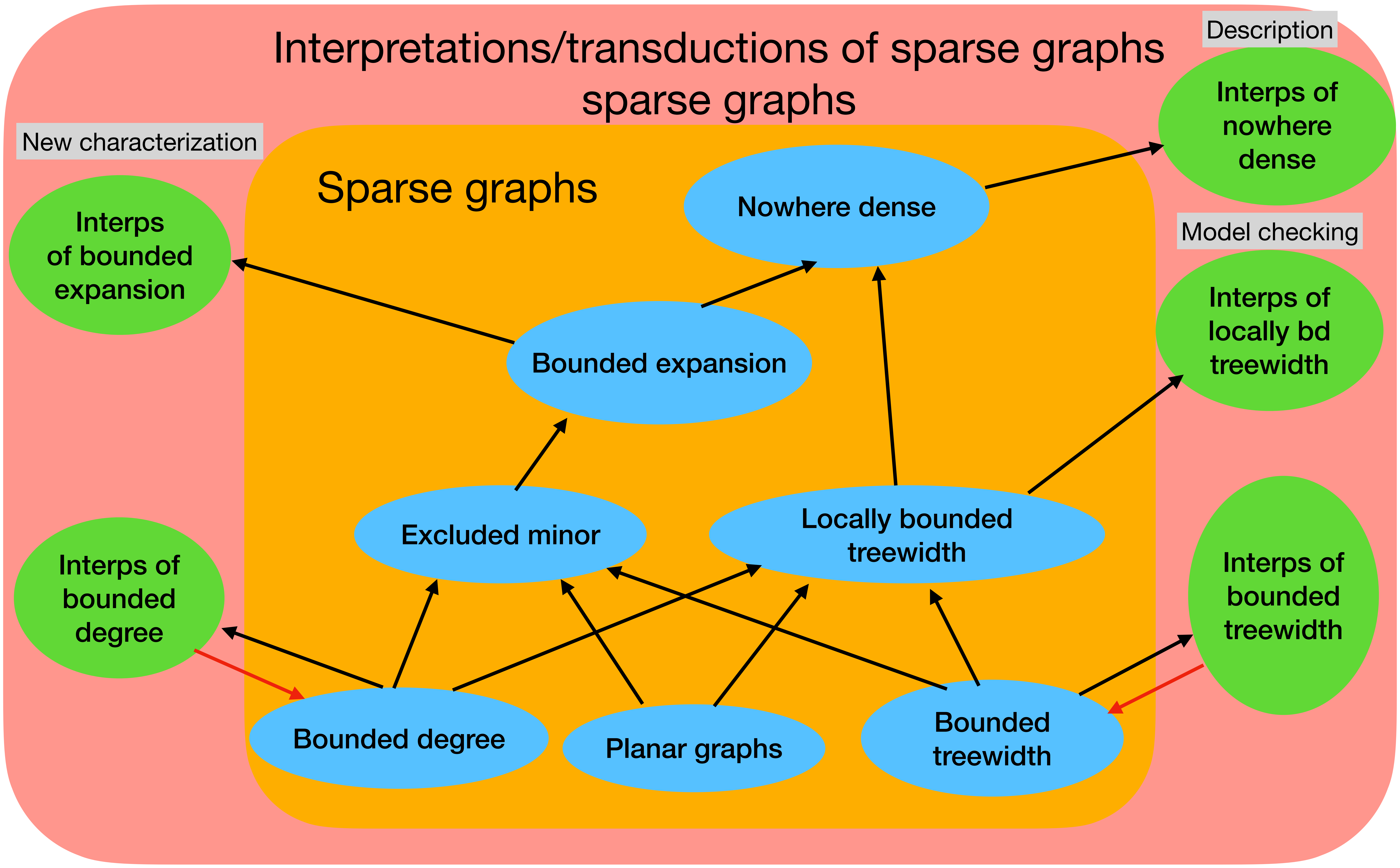

Interpretations/transductions of sparse graphs
sparse graphs
Sparse graphs
Nowhere dense
Bounded expansion
Excluded minor
Locally bounded treewidth
Interps of bounded degree
Interps of bounded treewidth
Bounded degree
Planar graphs
Bounded treewidth

What did we do?


Interpretations/transductions of sparse graphs
sparse graphs
Sparse graphs
Nowhere dense
Bounded expansion
Excluded minor
Locally bounded treewidth
Interps of bounded degree
Interps of bounded treewidth
Bounded degree
Planar graphs
Bounded treewidth


Interpretations/transductions of sparse graphs
sparse graphs
Sparse graphs
Nowhere dense
Model checking
Bounded expansion
Interps of locally bd treewidth
Excluded minor
Locally bounded treewidth
Interps of bounded degree
Interps of bounded treewidth
Bounded degree
Planar graphs
Bounded treewidth


Interpretations/transductions of sparse graphs
sparse graphs
New characterization
Sparse graphs
Nowhere dense
Model checking
Interps of bounded expansion
Bounded expansion
Interps of locally bd treewidth
Excluded minor
Locally bounded treewidth
Interps of bounded degree
Interps of bounded treewidth
Bounded degree
Planar graphs
Bounded treewidth


Interpretations/transductions of sparse graphs
sparse graphs
Description
Interps of nowhere dense
New characterization
Sparse graphs
Nowhere dense
Interps of bounded expansion
Model checking
Interps of locally bd treewidth
Bounded expansion
Excluded minor
Locally bounded treewidth
Interps of bounded degree
Interps of bounded treewidth
Bounded degree
Planar graphs
Bounded treewidth

Our results

Theorem (Bonnet, Dreier, G., Kreutzer, Mählmann, Simon, Toruńczyk; LICS 2022):
Let $\mathscr{D}$ be a graph class interpretable in planar graphs. Then the FO model checking problem is in FPT on $\mathscr{D}$.

Theorem (Dreier, G., Kiefer, Mi. Pilipczuk, Toruńczyk; LICS 2022):
A class $\mathscr{D}$ of graphs is interpretable in a class $\mathscr{C}$ of bounded expansion if and only if there exists a class $\mathscr{B}$ of *bushes* of bounded height and bounded expansion representing $\mathscr{D}$.

Theorem (Dreier, G., Kiefer, Mi. Pilipczuk, Toruńczyk; LICS 2022):
Let $\mathscr{D}$ be a graph class interpretable in a nowhere dense class of graphs $\mathscr{C}$. Then there exists a class $\mathscr{B}$ of quasi-bushes of bounded height which is almost nowhere dense and which represents $\mathscr{D}$.

Model checking on interpretations of planar graphs

Model checking on interpretations of planar graphs

Theorem (Bonnet, Dreier, G., Kreutzer, Mählmann, Simon, Toruńczyk; LICS 2022):
Let $\mathscr{D}$ be a graph class interpretable in planar graphs. Then the FO model checking problem is in FPT on $\mathscr{D}$.

Not proven by approximate interpretation reversals, but somewhat similarly.

Model checking on interpretations of planar graphs

Theorem (Bonnet, Dreier, G., Kreutzer, Mählmann, Simon, Toruńczyk; LICS 2022):
Let $\mathscr{D}$ be a graph class interpretable in planar graphs. Then the FO model checking problem is in FPT on $\mathscr{D}$.

Not proven by approximate interpretation reversals, but somewhat similarly.

Before we explain the algorithm, we need to explain the model checking algorithms for (some) classes of sparse graphs based on Gaifman's locality theorem.

Interlude:
Gaifman’s locality theorem and model checking on sparse graphs

Gaifman's locality theorem

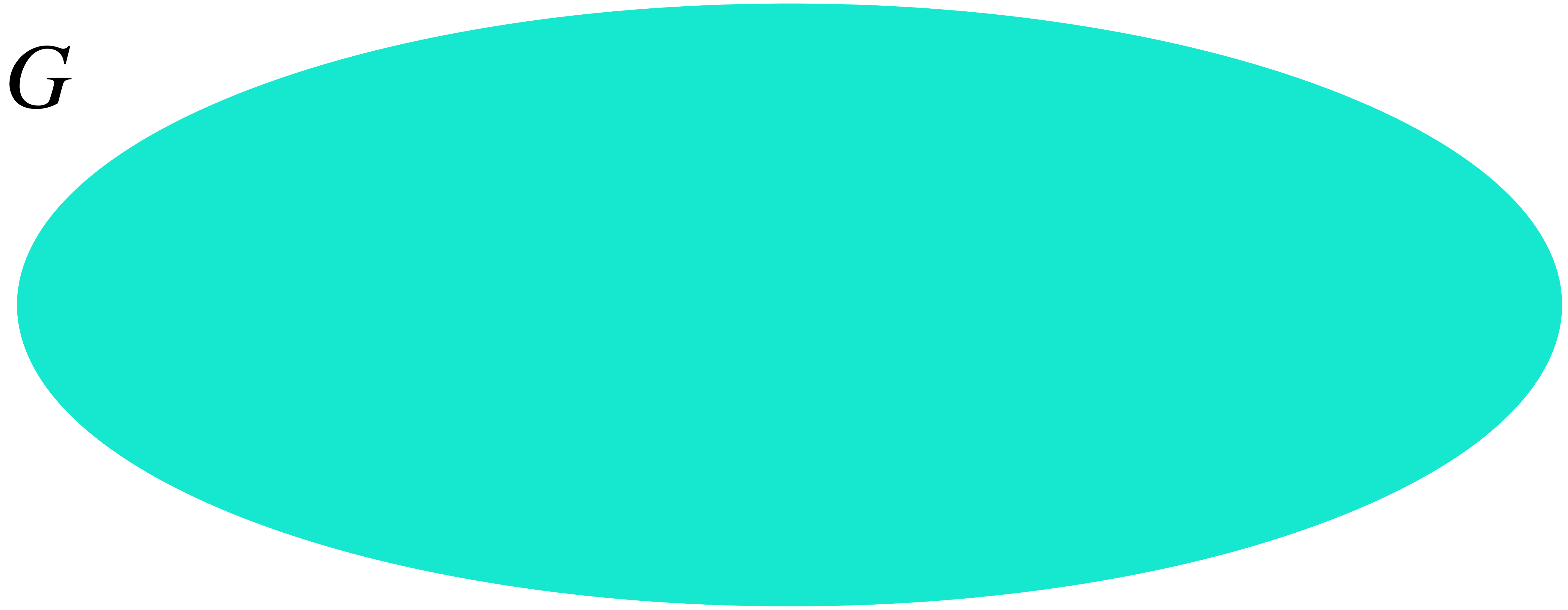
For every FO sentence φ there exist r, m and formulas $\gamma_1(x), \ldots, \gamma_m(x)$ such that to evaluate φ on any graph G one needs to:

1. For every $v \in V(G)$ look at $G[N_r(v)]$
2. Evaluate all formulas $\gamma_1(x), \ldots, \gamma_m(x)$ on $G[N_r(v)]$ and store the results
3. After doing this for all $v \in V(G)$ combine the results together

Gaifman's locality theorem

For every FO sentence φ there exist r, m and formulas $\gamma_1(x), \ldots, \gamma_m(x)$ such that to evaluate φ on any graph G one needs to:

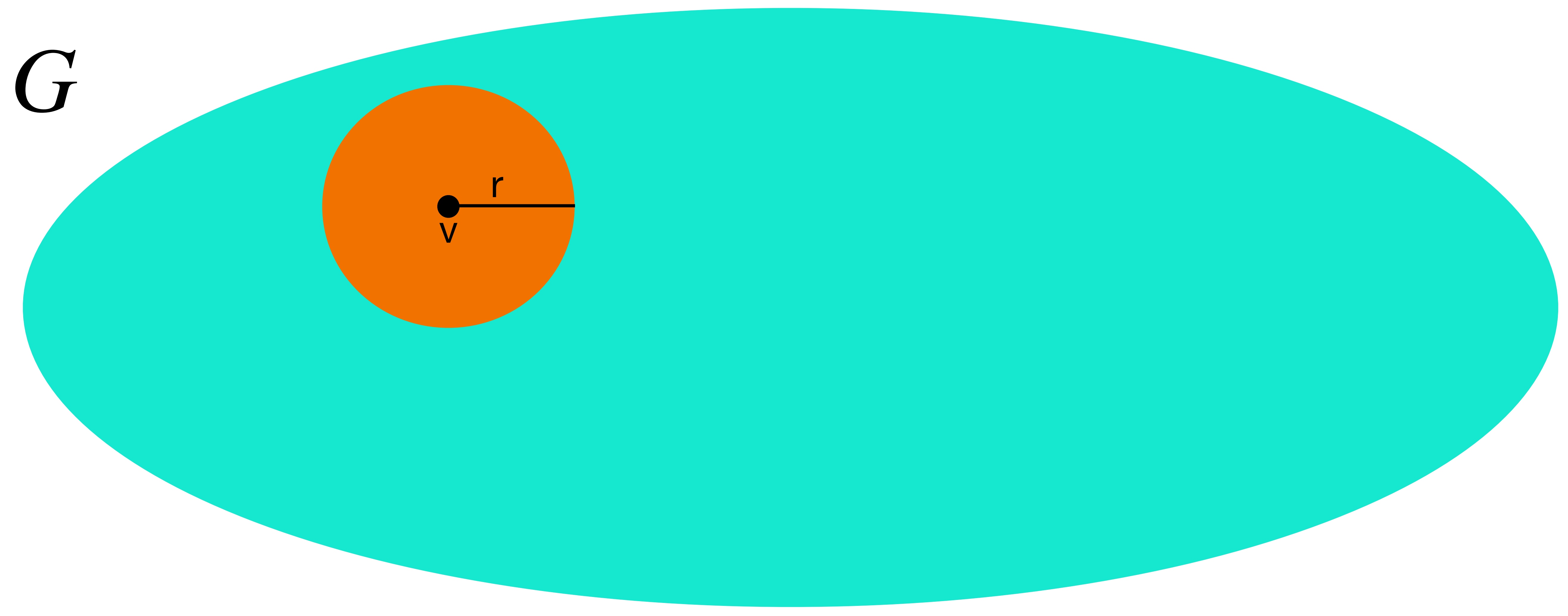
1. For every $v \in V(G)$ look at $G[N_r(v)]$
2. Evaluate all formulas $\gamma_1(x), \ldots, \gamma_m(x)$ on $G[N_r(v)]$ and store the results
3. After doing this for all $v \in V(G)$ combine the results together



Gaifman's locality theorem

For every FO sentence φ there exist r, m and formulas $\gamma_1(x), \ldots, \gamma_m(x)$ such that to evaluate φ on any graph G one needs to:

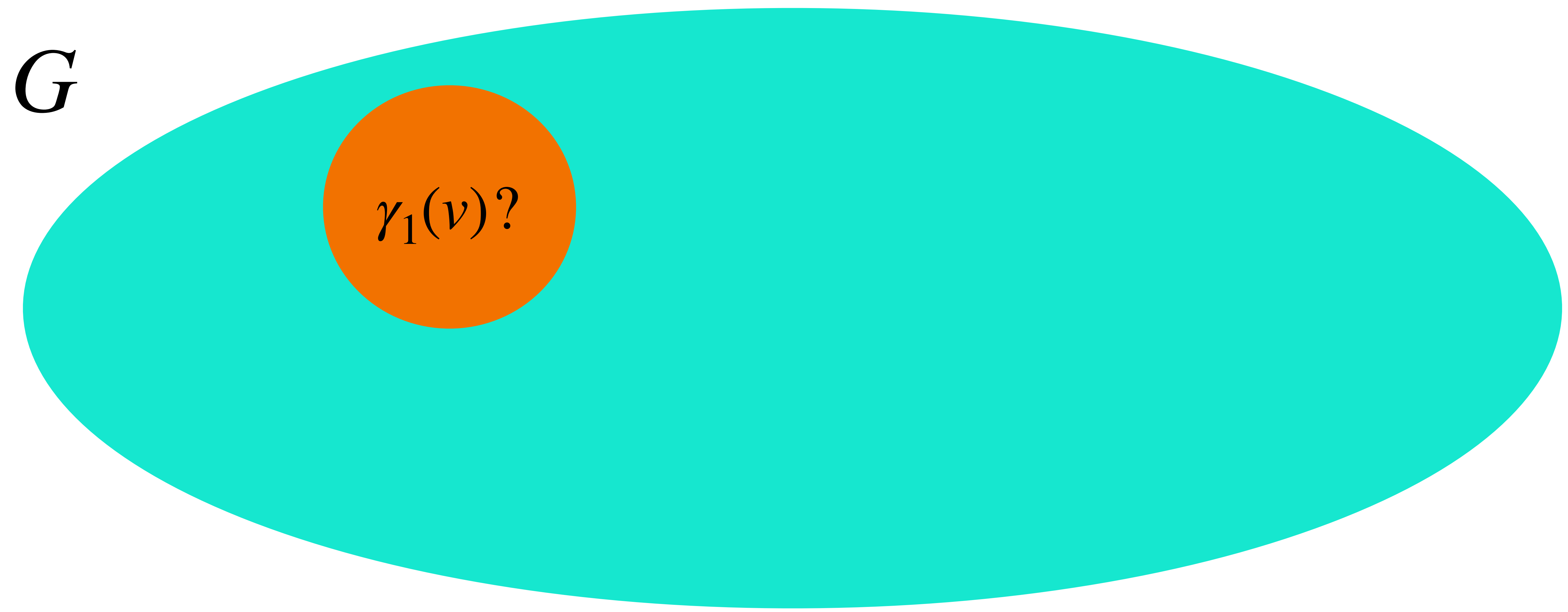
1. For every $v \in V(G)$ look at $G[N_r(v)]$
2. Evaluate all formulas $\gamma_1(x), \ldots, \gamma_m(x)$ on $G[N_r(v)]$ and store the results
3. After doing this for all $v \in V(G)$ combine the results together



Gaifman's locality theorem

For every FO sentence φ there exist r, m and formulas $\gamma_1(x), \ldots, \gamma_m(x)$ such that to evaluate φ on any graph G one needs to:

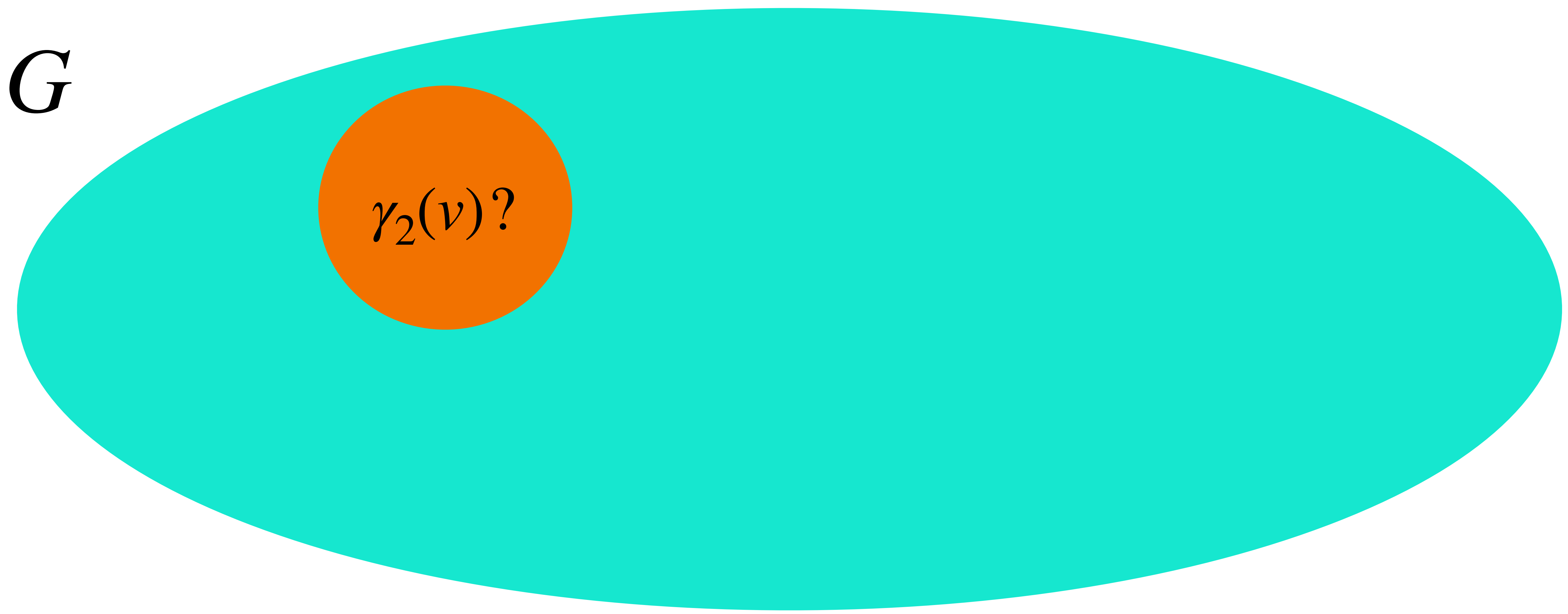
1. For every $v \in V(G)$ look at $G[N_r(v)]$
2. Evaluate all formulas $\gamma_1(x), \ldots, \gamma_m(x)$ on $G[N_r(v)]$ and store the results
3. After doing this for all $v \in V(G)$ combine the results together



Gaifman's locality theorem

For every FO sentence φ there exist r, m and formulas $\gamma_1(x), \ldots, \gamma_m(x)$ such that to evaluate φ on any graph G one needs to:

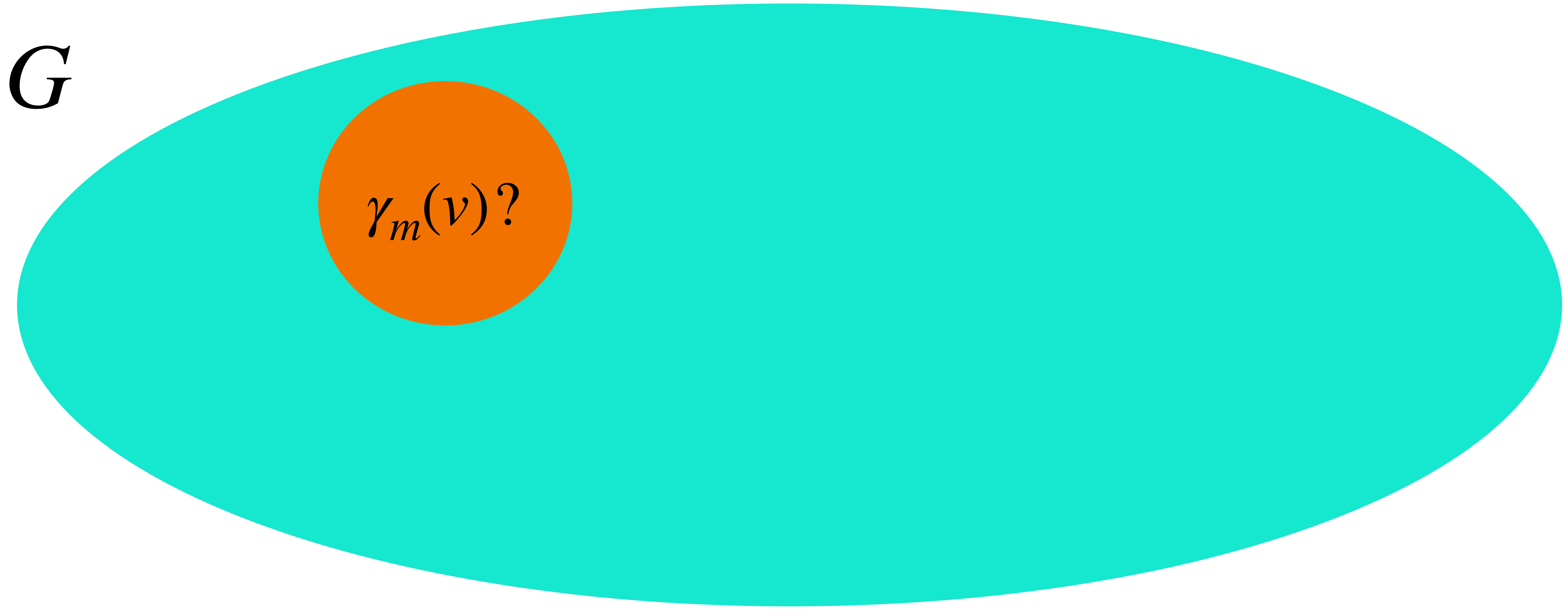
1. For every $v \in V(G)$ look at $G[N_r(v)]$
2. Evaluate all formulas $\gamma_1(x), \ldots, \gamma_m(x)$ on $G[N_r(v)]$ and store the results
3. After doing this for all $v \in V(G)$ combine the results together



Gaifman's locality theorem

For every FO sentence φ there exist r, m and formulas $\gamma_1(x), \ldots, \gamma_m(x)$ such that to evaluate φ on any graph G one needs to:

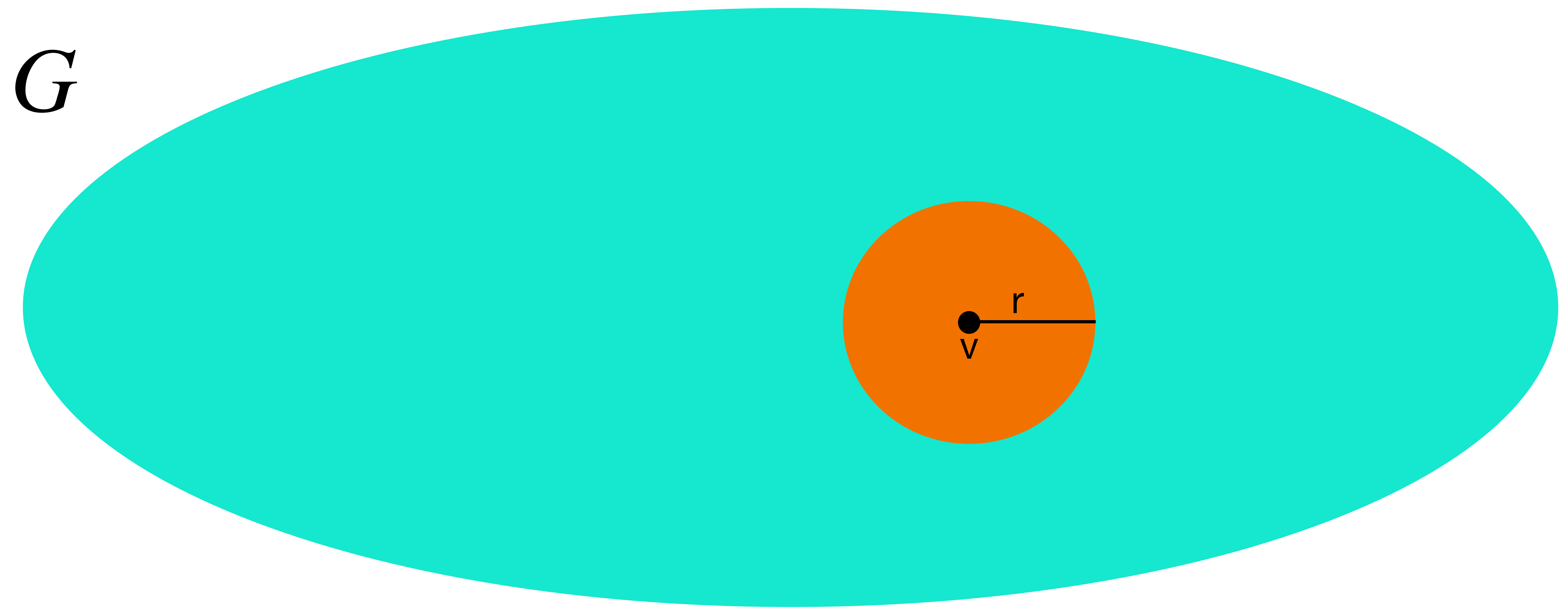
1. For every $v \in V(G)$ look at $G[N_r(v)]$
2. Evaluate all formulas $\gamma_1(x), \ldots, \gamma_m(x)$ on $G[N_r(v)]$ and store the results
3. After doing this for all $v \in V(G)$ combine the results together



Gaifman's locality theorem

For every FO sentence φ there exist r, m and formulas $\gamma_1(x), \ldots, \gamma_m(x)$ such that to evaluate φ on any graph G one needs to:

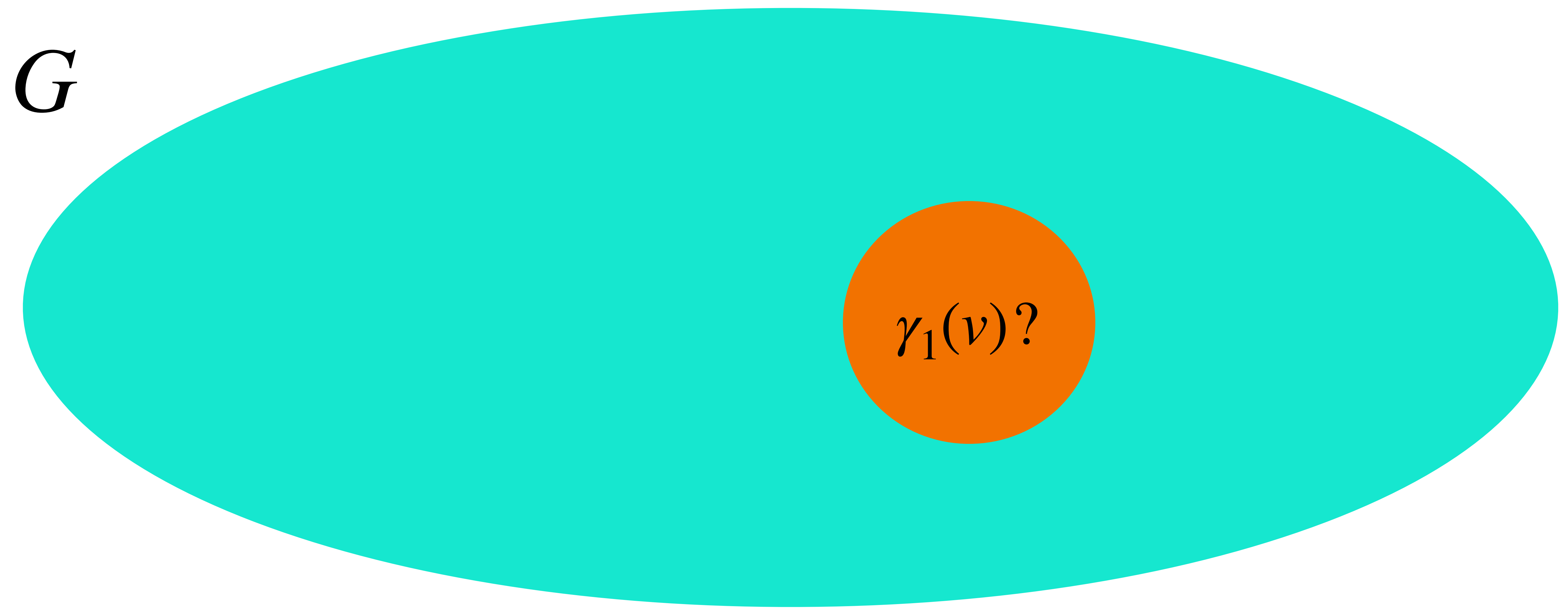
1. For every $v \in V(G)$ look at $G[N_r(v)]$
2. Evaluate all formulas $\gamma_1(x), \ldots, \gamma_m(x)$ on $G[N_r(v)]$ and store the results
3. After doing this for all $v \in V(G)$ combine the results together



Gaifman's locality theorem

For every FO sentence φ there exist r, m and formulas $\gamma_1(x), \ldots, \gamma_m(x)$ such that to evaluate φ on any graph G one needs to:

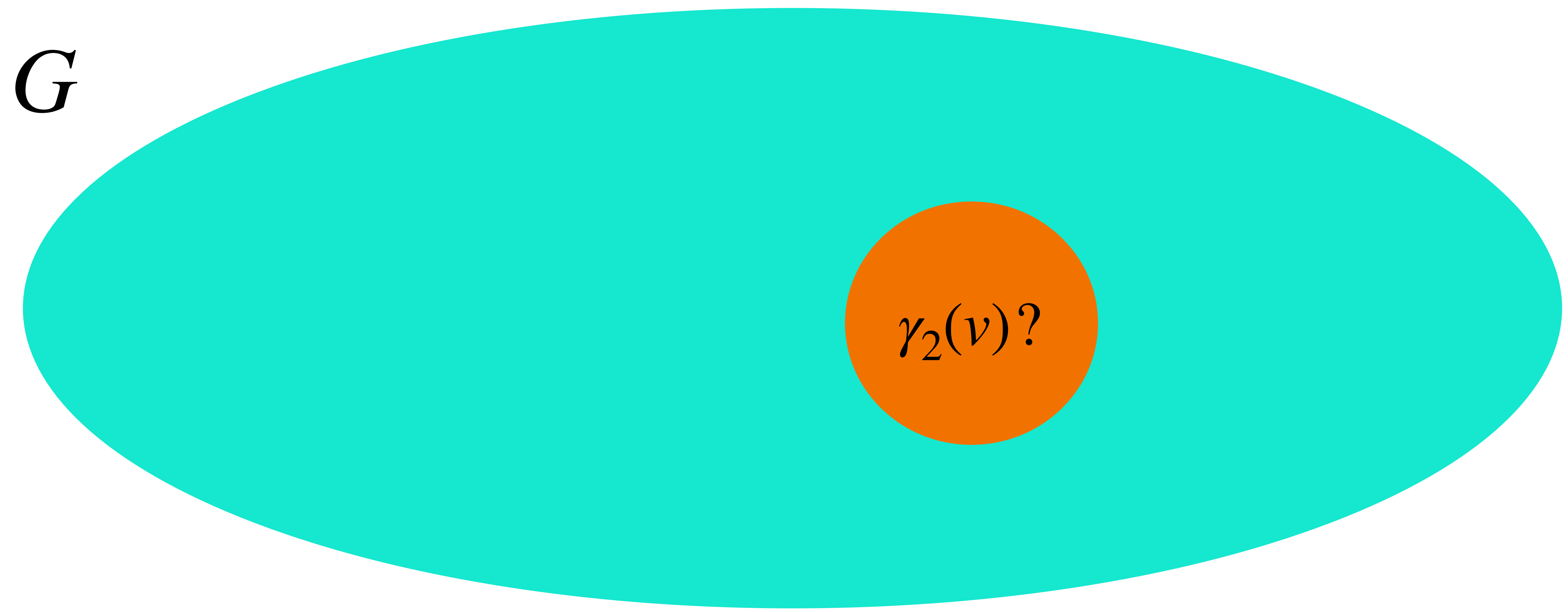
1. For every $v \in V(G)$ look at $G[N_r(v)]$
2. Evaluate all formulas $\gamma_1(x), \ldots, \gamma_m(x)$ on $G[N_r(v)]$ and store the results
3. After doing this for all $v \in V(G)$ combine the results together



Gaifman's locality theorem

For every FO sentence φ there exist r, m and formulas $\gamma_1(x), \ldots, \gamma_m(x)$ such that to evaluate φ on any graph G one needs to:

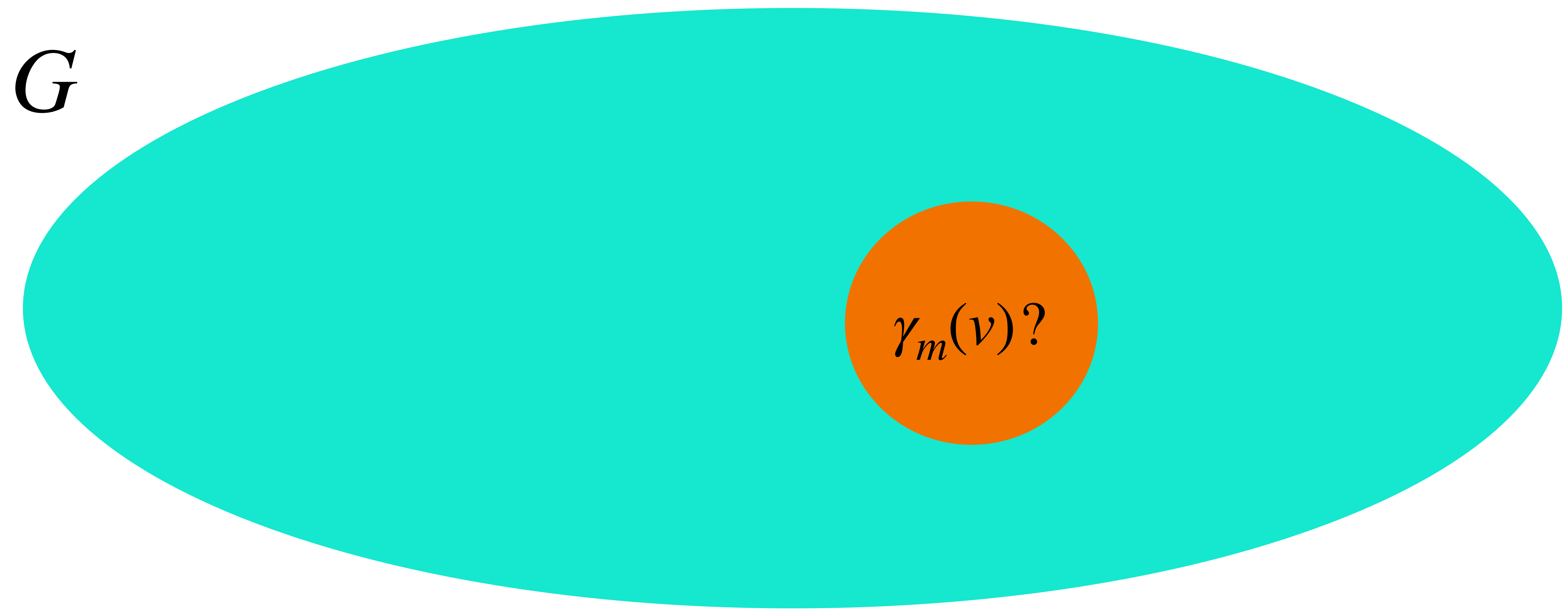
1. For every $v \in V(G)$ look at $G[N_r(v)]$
2. Evaluate all formulas $\gamma_1(x), \ldots, \gamma_m(x)$ on $G[N_r(v)]$ and store the results
3. After doing this for all $v \in V(G)$ combine the results together



Gaifman's locality theorem

For every FO sentence φ there exist r, m and formulas $\gamma_1(x), \dots, \gamma_m(x)$ such that to evaluate φ on any graph G one needs to:

1. For every $v \in V(G)$ look at $G[N_r(v)]$
2. Evaluate all formulas $\gamma_1(x), \dots, \gamma_m(x)$ on $G[N_r(v)]$ and store the results
3. After doing this for all $v \in V(G)$ combine the results together



Gaifman's locality theorem

For every FO sentence φ there exist r, m and formulas $\gamma_1(x), \ldots, \gamma_m(x)$ such that to evaluate φ on any graph G one needs to:

1. For every $v \in V(G)$ look at $G[N_r(v)]$
2. Evaluate all formulas $\gamma_1(x), \ldots, \gamma_m(x)$ on $G[N_r(v)]$ and store the results
3. After doing this for all $v \in V(G)$ combine the results together

Model checking on a class of graphs of degree at most d:

Input: graph G of degree at most d, FO sentence φ

Gaifman's locality theorem

For every FO sentence φ there exist r, m and formulas $\gamma_1(x), \ldots, \gamma_m(x)$ such that to evaluate φ on any graph G one needs to:

1. For every $v \in V(G)$ look at $G[N_r(v)]$
2. Evaluate all formulas $\gamma_1(x), \ldots, \gamma_m(x)$ on $G[N_r(v)]$ and store the results
3. After doing this for all $v \in V(G)$ combine the results together

Model checking on a class of graphs of degree at most d:

Input: graph G of degree at most d, FO sentence φ

1. For every $v \in V(G)$ look at $G[N_r(v)]$ (this has size at most $d^r + 1$)
2. Evaluate all formulas $\gamma_1(x), \ldots, \gamma_m(x)$ on $G[N_r(v)]$ and store the results
3. After doing this for all $v \in V(G)$ combine the results together (easy)

Gaifman's locality theorem

Definition:
Class $\mathscr{C}$ has locally bounded treewidth if there exists a function $f : \mathbb{N} \to \mathbb{N}$ such that: For any $G \in \mathscr{C}$ and any $v \in V(G)$, the graph induced by $N_r(v)$ has treewidth at most $f(r)$.

Gaifman's locality theorem

Definition:
Class $\mathscr{C}$ has locally bounded treewidth if there exists a function $f : \mathbb{N} \to \mathbb{N}$ such that: For any $G \in \mathscr{C}$ and any $v \in V(G)$, the graph induced by $N_r(v)$ has treewidth at most $f(r)$.

Important example: planar graphs

Gaifman's locality theorem

Definition:
Class $\mathscr{C}$ has locally bounded treewidth if there exists a function $f : \mathbb{N} \to \mathbb{N}$ such that: For any $G \in \mathscr{C}$ and any $v \in V(G)$, the graph induced by $N_r(v)$ has treewidth at most $f(r)$.

Important example: planar graphs

Model checking on a class of graphs of locally bounded treewidth:
Input: graph G of degree at most d, FO sentence φ

1. For every $v \in V(G)$ look at $G[N_r(v)]$ (this has treewidth at most $f(r)$)
2. Evaluate all formulas $\gamma_1(x), \ldots, \gamma_m(x)$ on $G[N_r(v)]$ and store the results
3. After doing this for all $v \in V(G)$ combine the results together (easy)

End of interlude

Our algorithm

Setting:

We work with a graph class $\mathscr{D}$ interpretable in planar graphs using formula $\psi(x, y)$.

We are given $G \in \mathscr{D}$ and φ as input.

Our algorithm

Setting:

We work with a graph class $\mathscr{D}$ interpretable in planar graphs using formula $\psi(x, y)$.

We are given $G \in \mathscr{D}$ and φ as input.

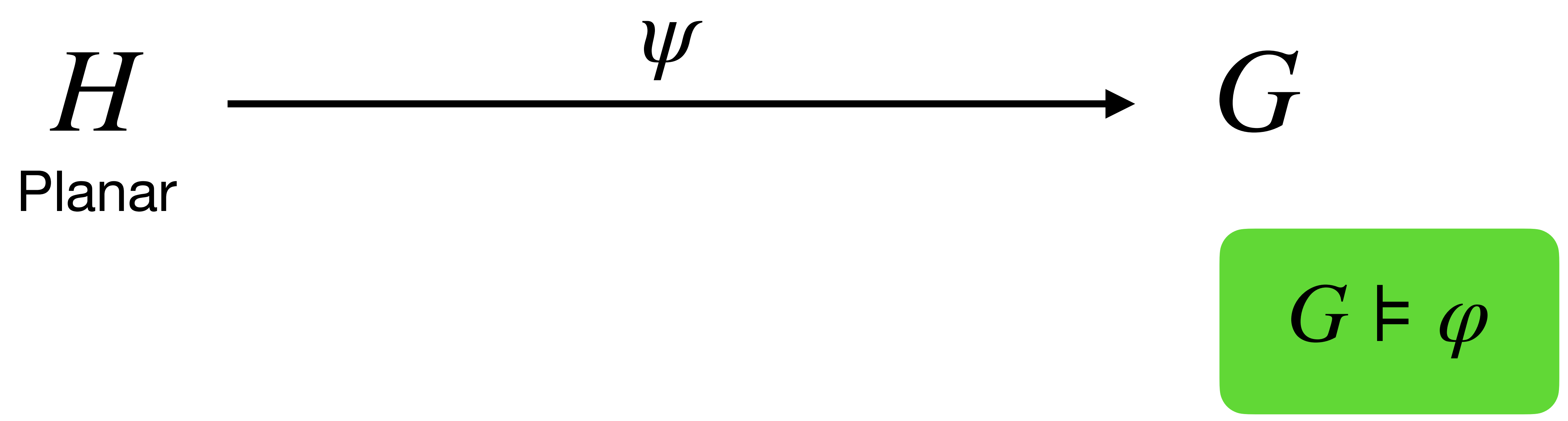
$G \models \varphi$

Our algorithm

Setting:

We work with a graph class $\mathscr{D}$ interpretable in planar graphs using formula $\psi(x, y)$.

We are given $G \in \mathscr{D}$ and φ as input.

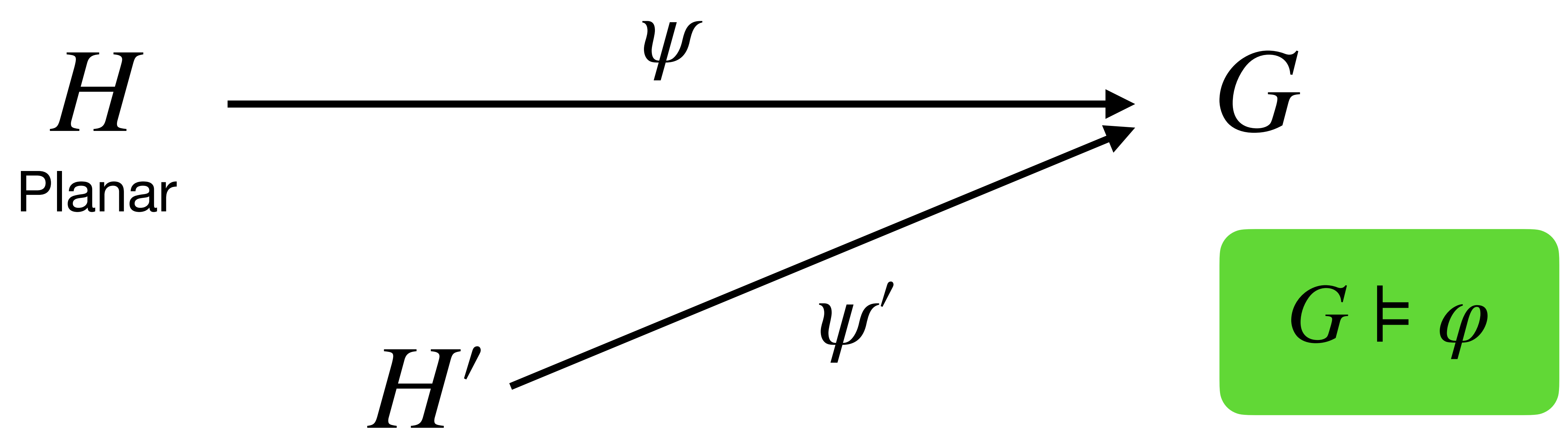


Our algorithm

Setting:

We work with a graph class $\mathscr{D}$ interpretable in planar graphs using formula $\psi(x, y)$.

We are given $G \in \mathscr{D}$ and φ as input.

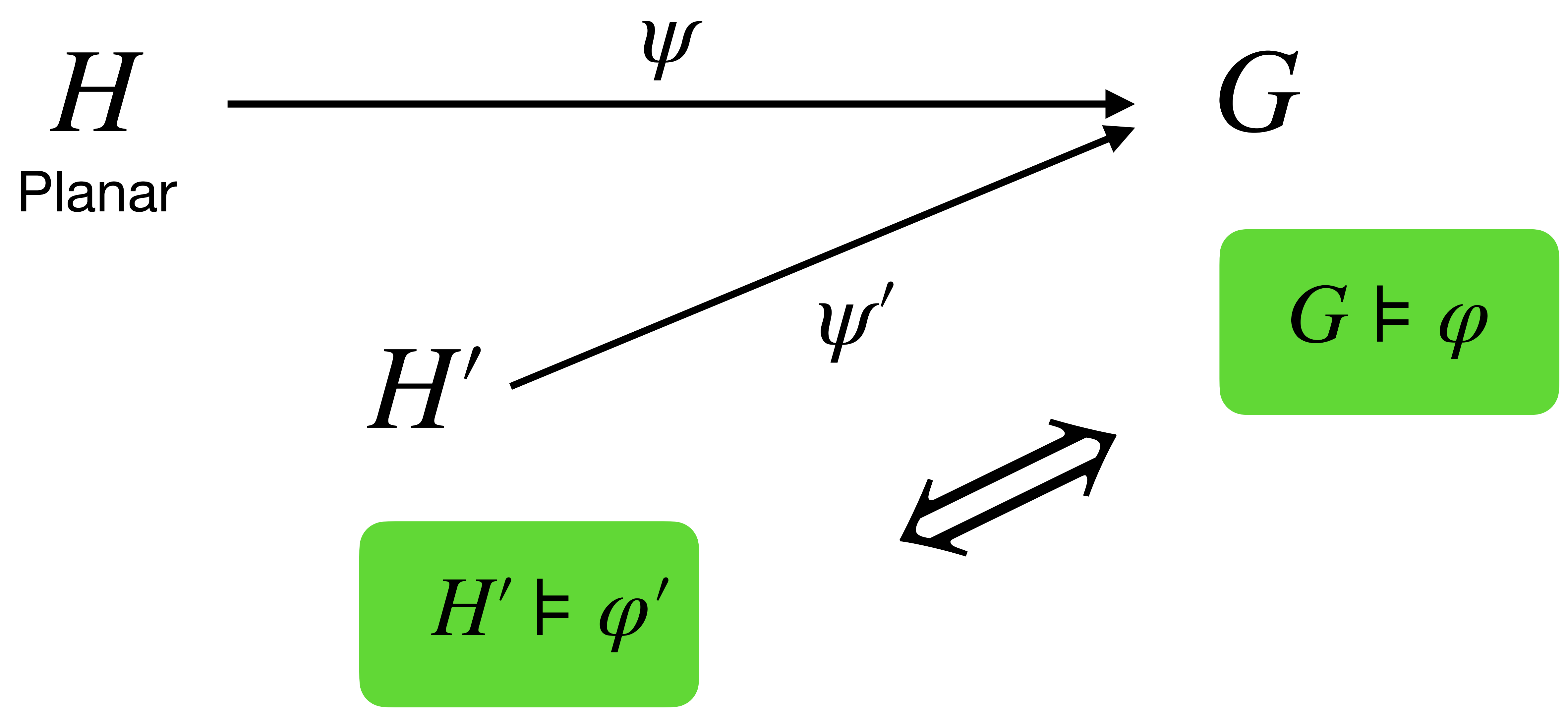


Our algorithm

Setting:

We work with a graph class $\mathscr{D}$ interpretable in planar graphs using formula $\psi(x, y)$.

We are given $G \in \mathscr{D}$ and φ as input.

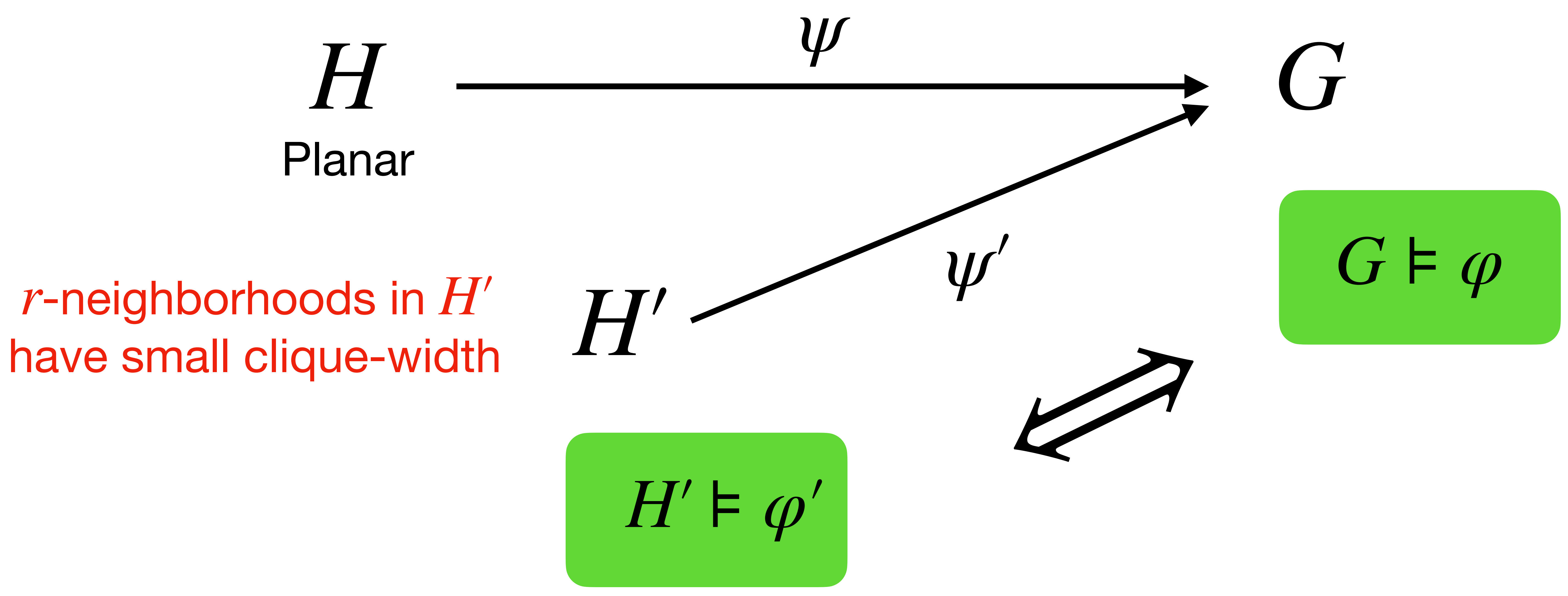


Our algorithm

Setting:

We work with a graph class $\mathscr{D}$ interpretable in planar graphs using formula $\psi(x, y)$.

We are given $G \in \mathscr{D}$ and φ as input.

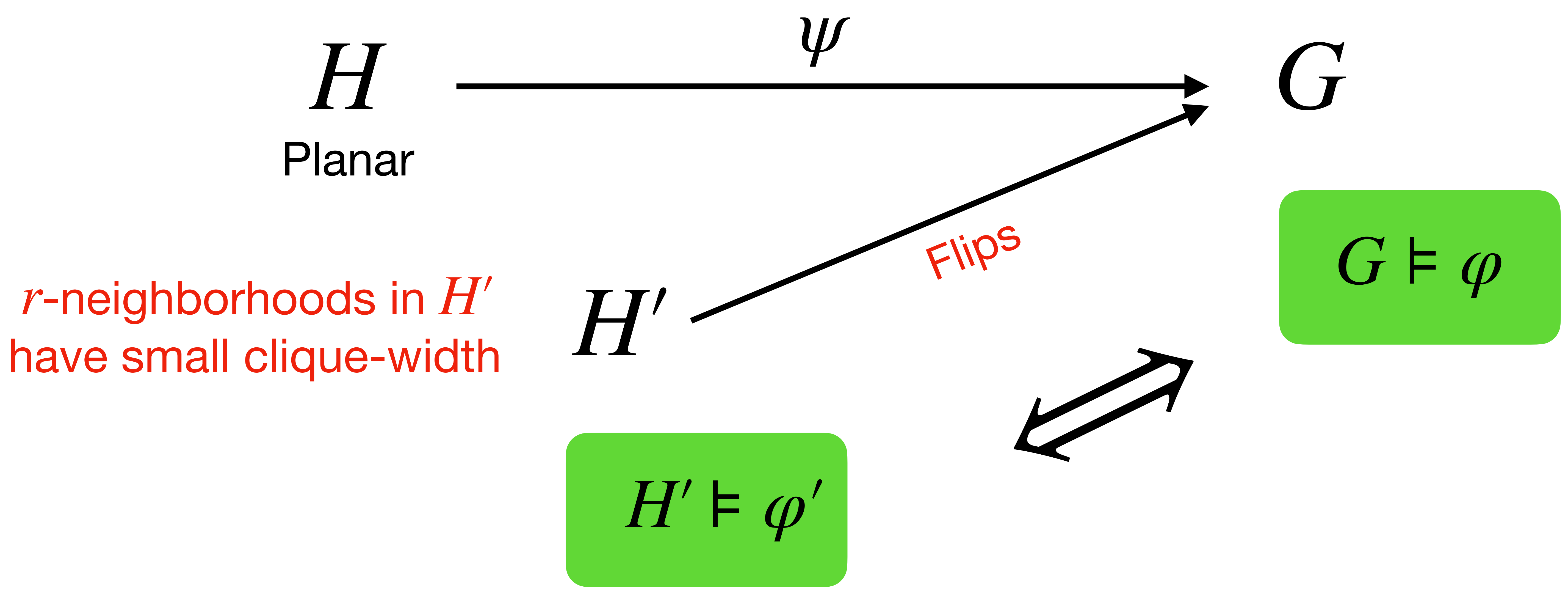


Our algorithm

Setting:

We work with a graph class $\mathscr{D}$ interpretable in planar graphs using formula $\psi(x, y)$.

We are given $G \in \mathscr{D}$ and φ as input.

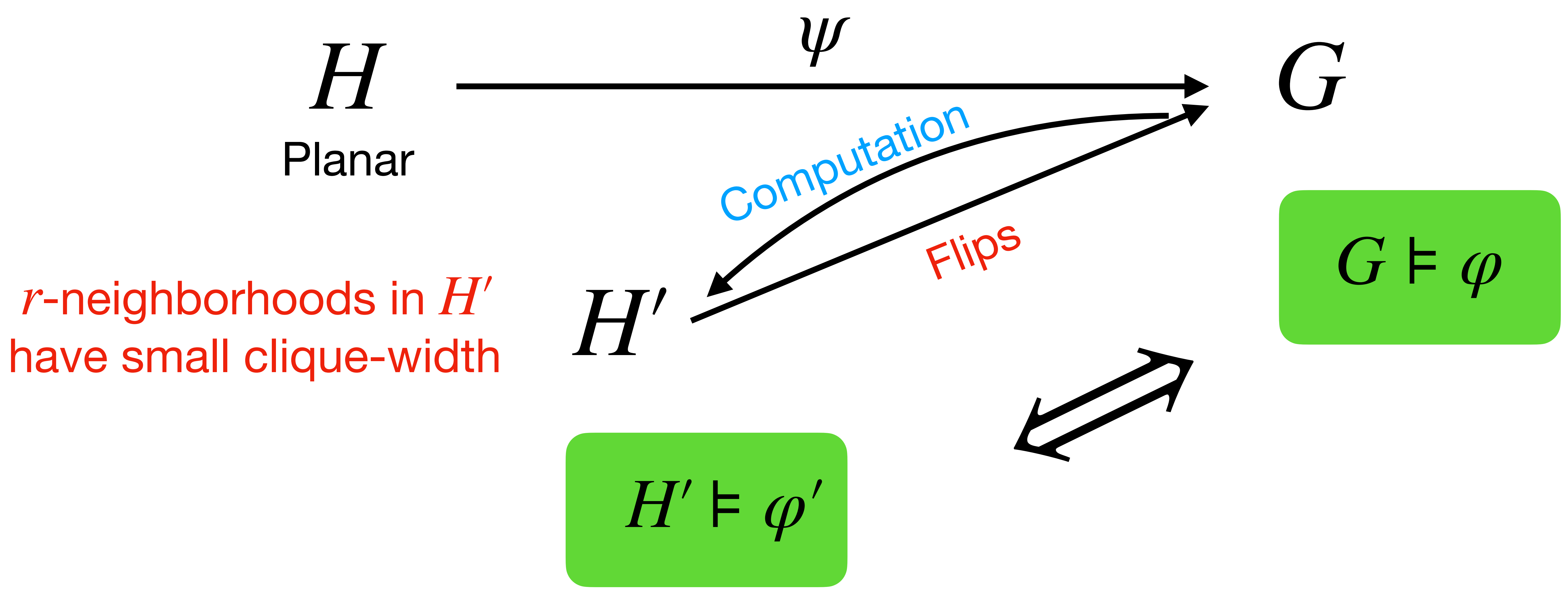


Our algorithm

Setting:

We work with a graph class $\mathscr{D}$ interpretable in planar graphs using formula $\psi(x, y)$.

We are given $G \in \mathscr{D}$ and φ as input.

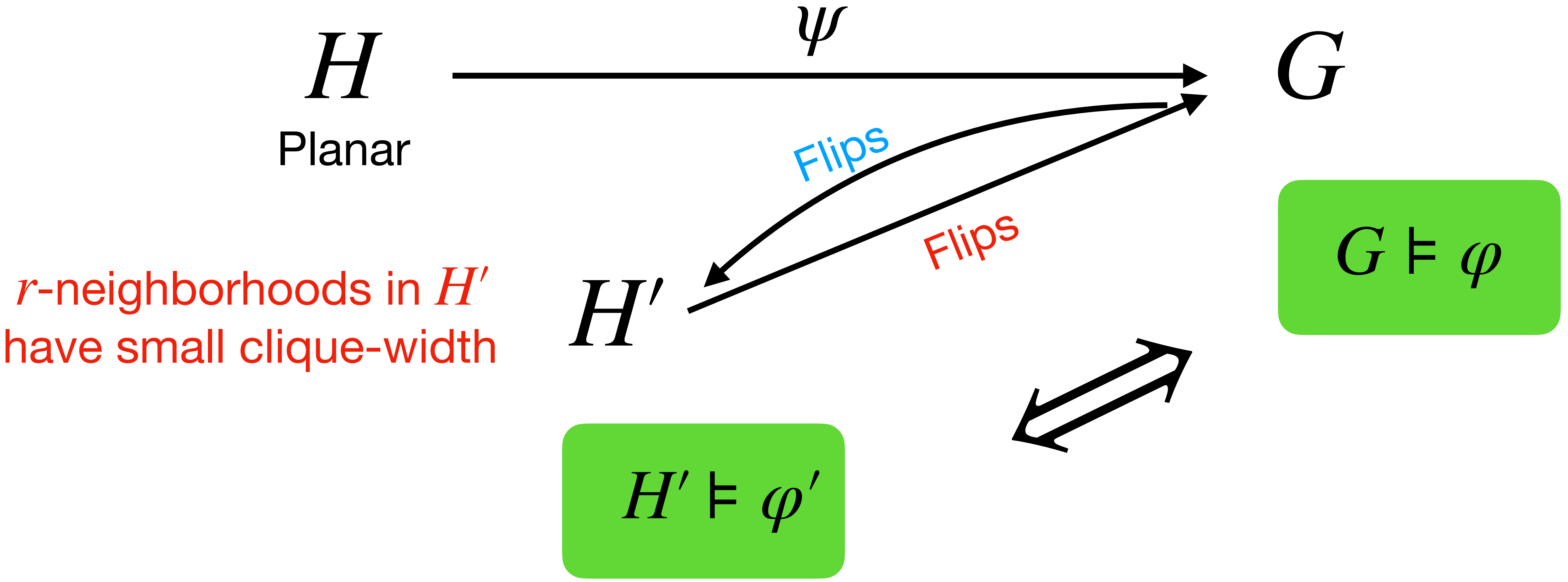


Our algorithm

Setting:

We work with a graph class $\mathscr{D}$ interpretable in planar graphs using formula $\psi(x, y)$.

We are given $G \in \mathscr{D}$ and φ as input.



The algorithm for computing H'

Given: G, φ

Computing H':

The algorithm for computing H'

Given: G, φ

Computing H':

For suitably chosen k, go through all k-tuples of vertices and for each k-tuple S:

The algorithm for computing H'

Given: G, φ

Computing H':

For suitably chosen k, go through all k-tuples of vertices and for each k-tuple S:

- Partition $V(G)$ based on the adjacency to S (giving $p := 2^k + k$) classes $V_1, \ldots, V_p$

The algorithm for computing H'

Given: G, φ

Computing H':

For suitably chosen k, go through all k-tuples of vertices and for each k-tuple S:

- Partition $V(G)$ based on the adjacency to S (giving $p := 2^k + k$) classes $V_1, \ldots, V_p$
- For each pair of classes V_i, V_j either complement (flip) the edges between V_i, V_j or not.

The algorithm for computing H'

Given: G, φ

Computing H':

For suitably chosen k, go through all k-tuples of vertices and for each k-tuple S:

- Partition $V(G)$ based on the adjacency to S (giving $p := 2^k + k$) classes $V_1, \ldots, V_p$
- For each pair of classes V_i, V_j either complement (flip) the edges between V_i, V_j or not.
- Check whether $G[N_r(v)]$ has small clique-width for each v.

The algorithm for computing H'

Given: G, φ

Computing H':

For suitably chosen k, go through all k-tuples of vertices and for each k-tuple S:

- Partition $V(G)$ based on the adjacency to S (giving $p := 2^k + k$) classes $V_1, \ldots, V_p$
- For each pair of classes V_i, V_j either complement (flip) the edges between V_i, V_j or not.
- Check whether $G[N_r(v)]$ has small clique-width for each v.

Runtime: $|G|^k$ guesses of S, for each of them $2^{(2^k+k)^2}$ choices for flips, and for each of them we check clique-width: $|G|^k \cdot 2^{(2^k+k)^2} \cdot f(cw) \cdot |G|^3$

The algorithm for computing H'

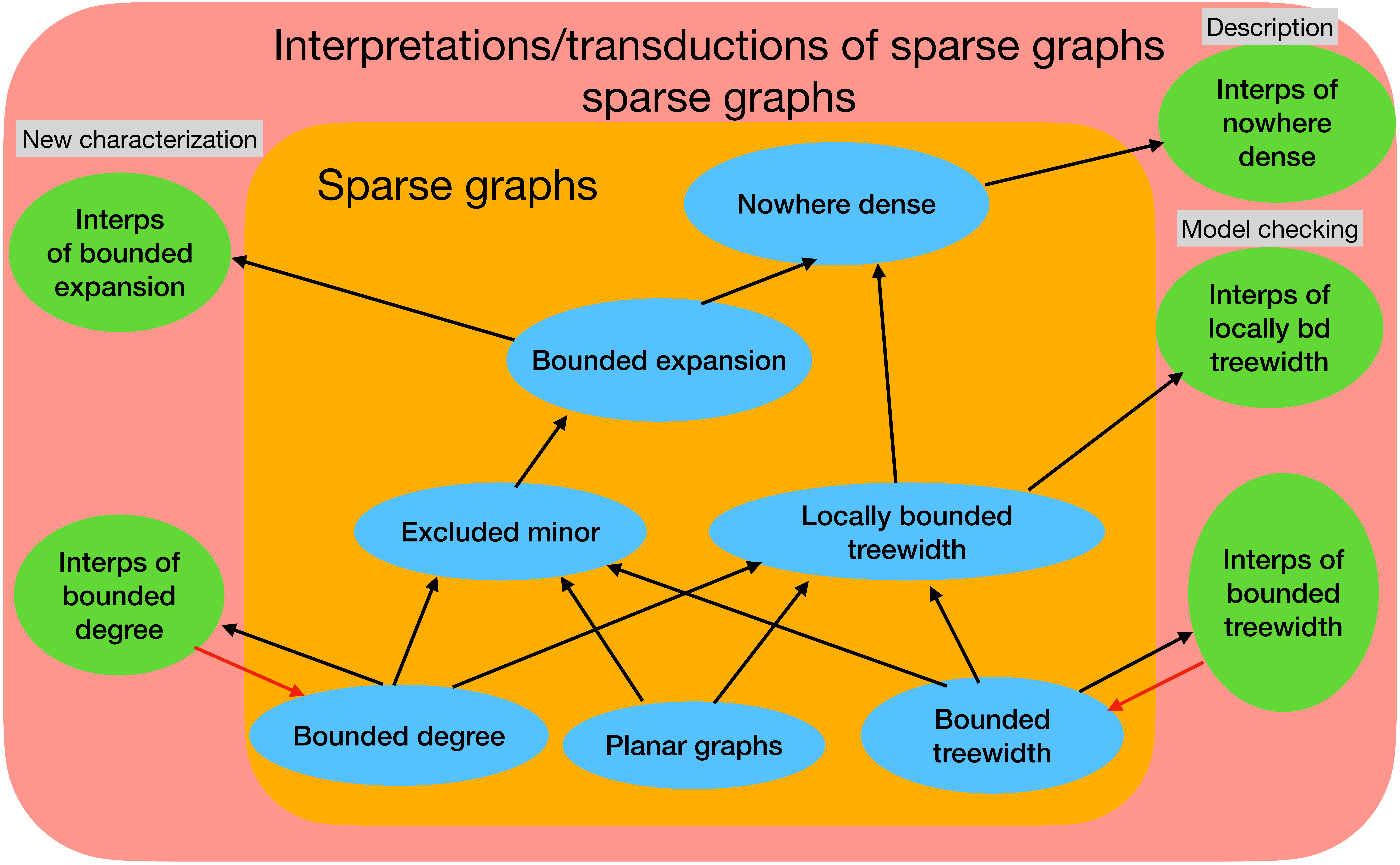
Given: G, φ

Computing H':

For suitably chosen k, go through all k-tuples of vertices and for each k-tuple S:

- Partition $V(G)$ based on the adjacency to S (giving $p := 2^k + k$) classes $V_1, \ldots, V_p$
- For each pair of classes V_i, V_j either complement (flip) the edges between V_i, V_j or not.
- Check whether $G[N_r(v)]$ has small clique-width for each v.

Technical core of the paper — proving that for some choice of S and some choices of flips the resulting graph will have locally small clique-width.


Interpretations/transductions of sparse graphs
sparse graphs
Description
Interps of nowhere dense
New characterization
Sparse graphs
Nowhere dense
Interps of bounded expansion
Model checking
Interps of locally bd treewidth
Bounded expansion
Excluded minor
Locally bounded treewidth
Interps of bounded degree
Interps of bounded treewidth
Bounded degree
Planar graphs
Bounded treewidth

Characterising interpretations of graph classes of bounded expansion and describing interpretations of nowhere dense graph classes

Our results

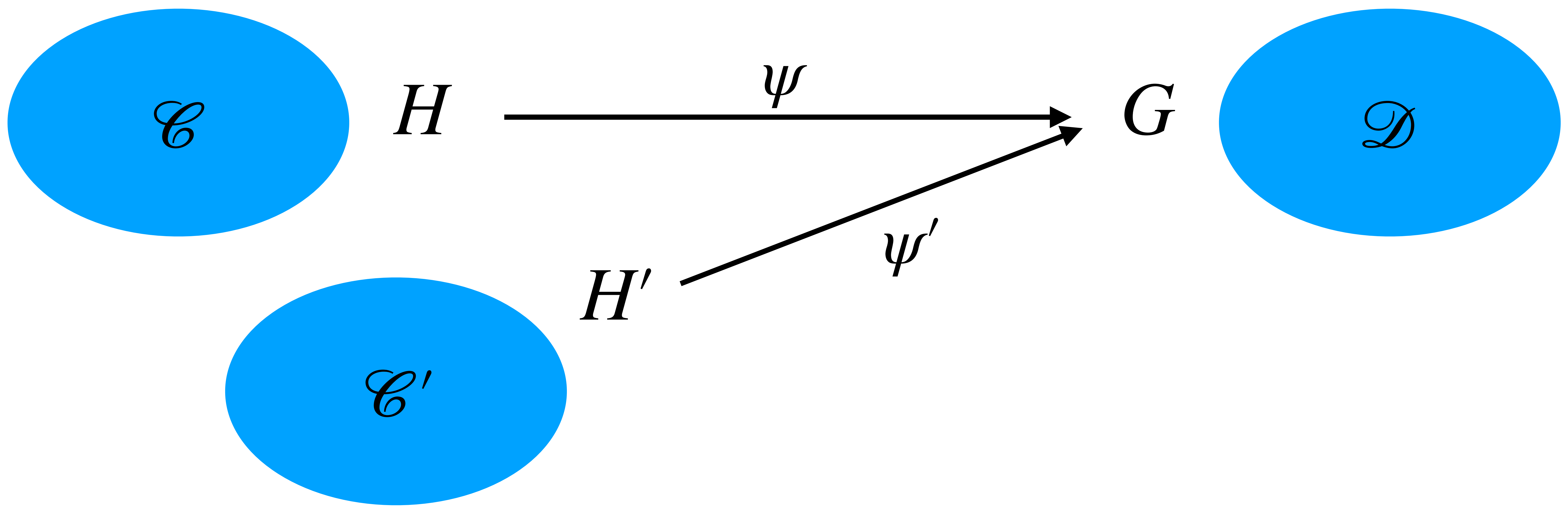
Theorem (Dreier, G., Kiefer, Mi. Pilipczuk, Toruńczyk; LICS 2022):
A class $\mathscr{D}$ of graphs is interpretable in a class $\mathscr{C}$ of bounded expansion if and only if there exists a class $\mathscr{B}$ of *bushes* of bounded height and bounded expansion representing $\mathscr{D}$.

Our results

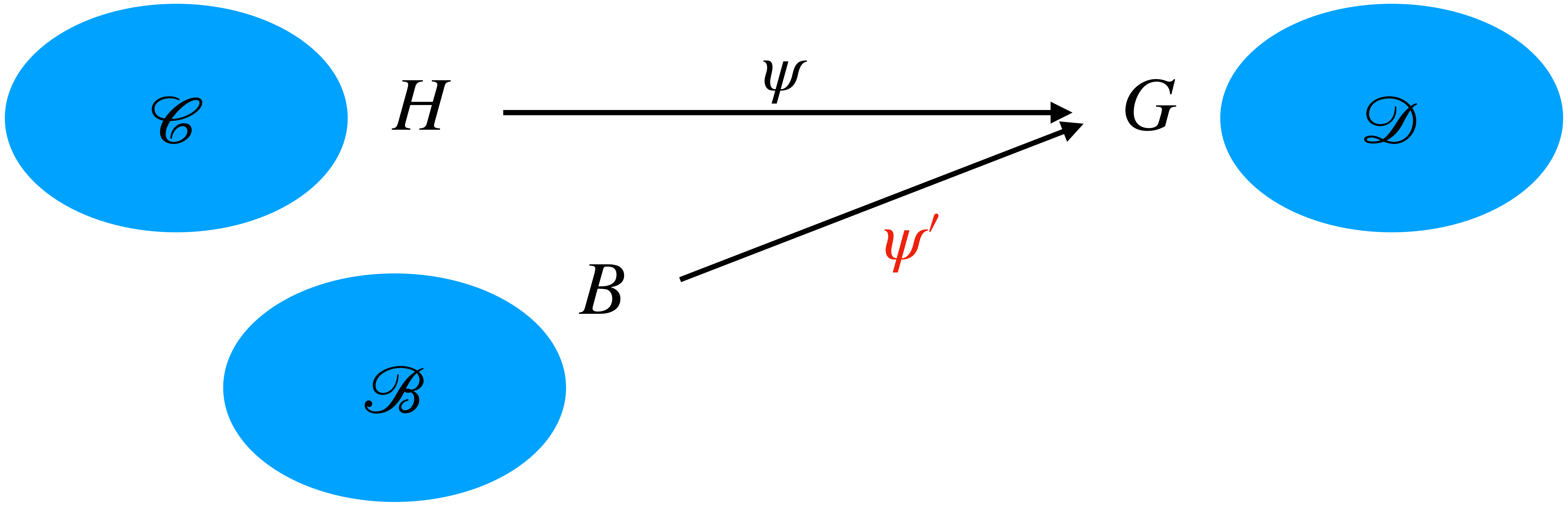
Theorem (Dreier, G., Kiefer, Mi. Pilipczuk, Toruńczyk; LICS 2022):
A class $\mathscr{D}$ of graphs is interpretable in a class $\mathscr{C}$ of bounded expansion if and only if there exists a class $\mathscr{B}$ of *bushes* of bounded height and bounded expansion representing $\mathscr{D}$.

Theorem (Dreier, G., Kiefer, Mi. Pilipczuk, Toruńczyk; LICS 2022):
Let $\mathscr{D}$ be a graph class interpretable in a nowhere dense class of graphs $\mathscr{C}$. Then there exists a class $\mathscr{B}$ of quasi-bushes of bounded height which is almost nowhere dense and which represents $\mathscr{D}$.

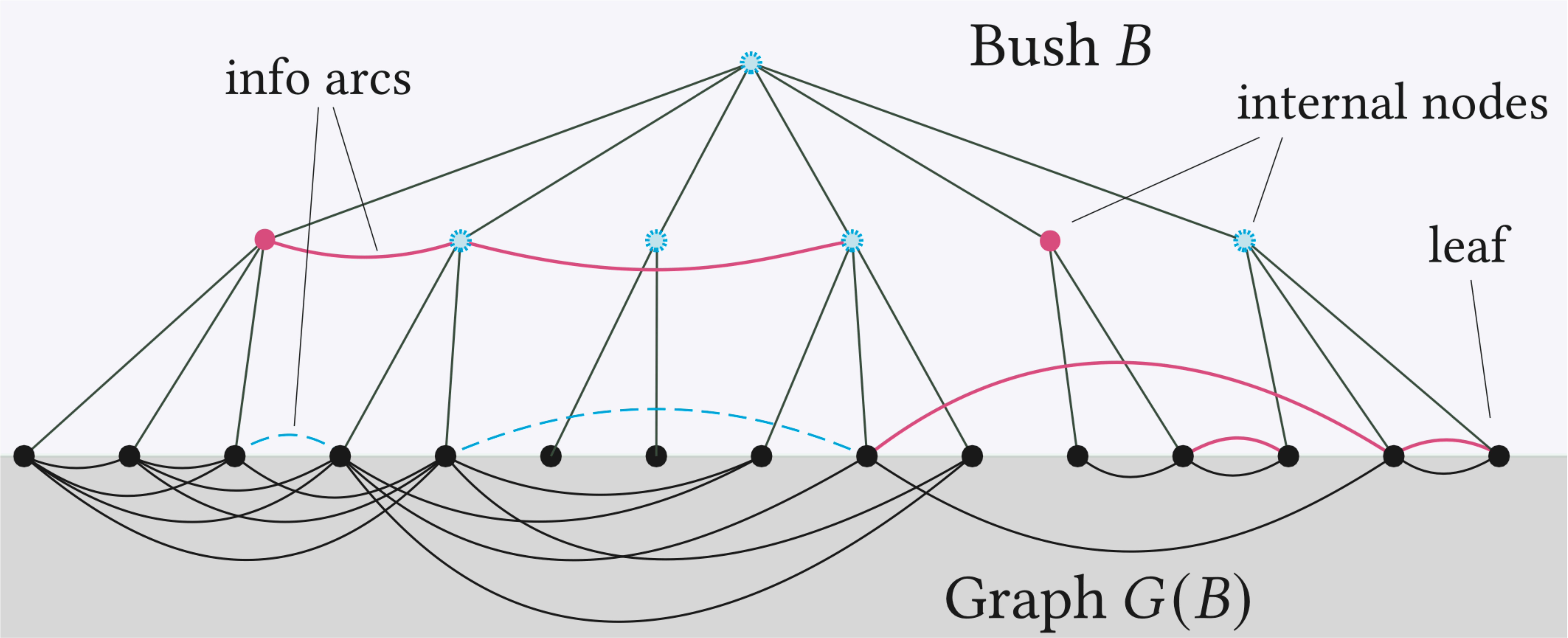
What is it good for?



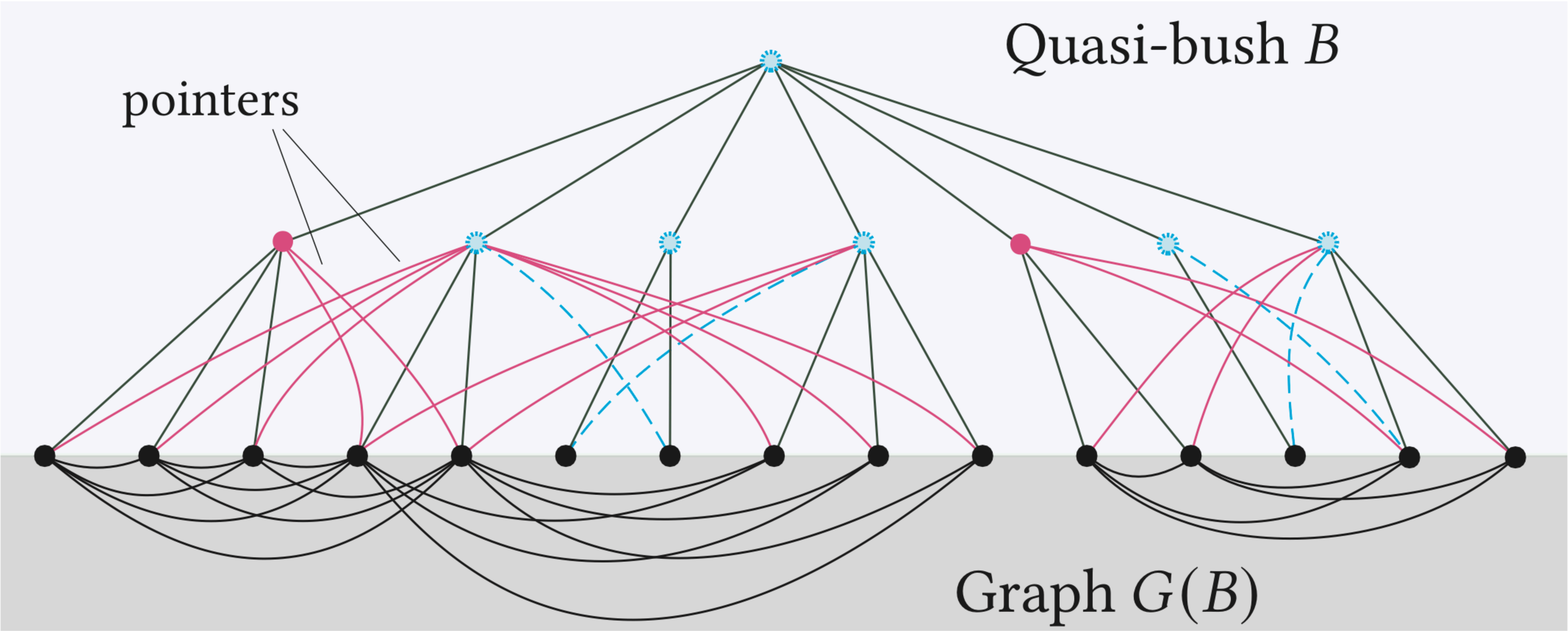
What is it good for?



Bushes



Quasi-bushes

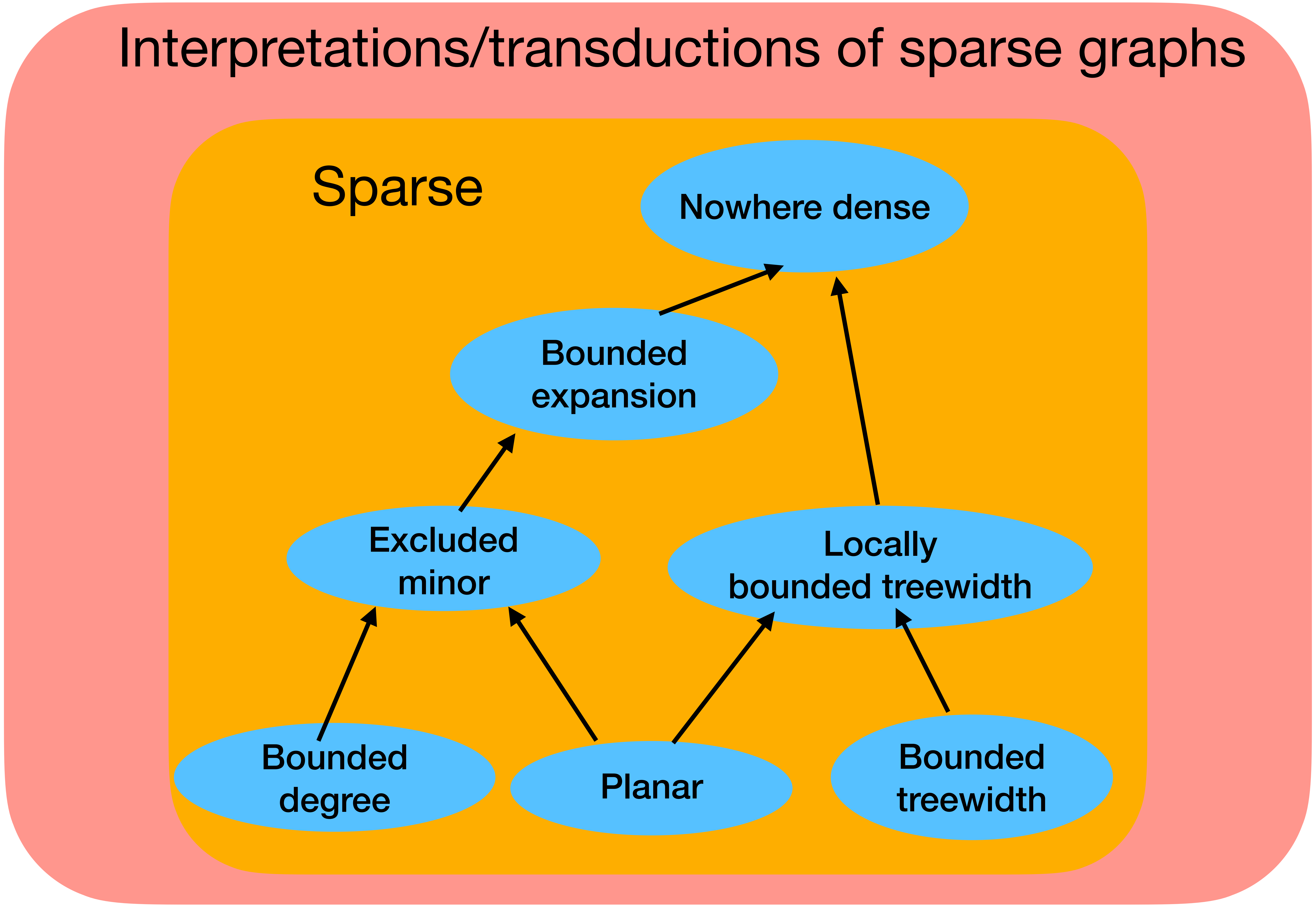


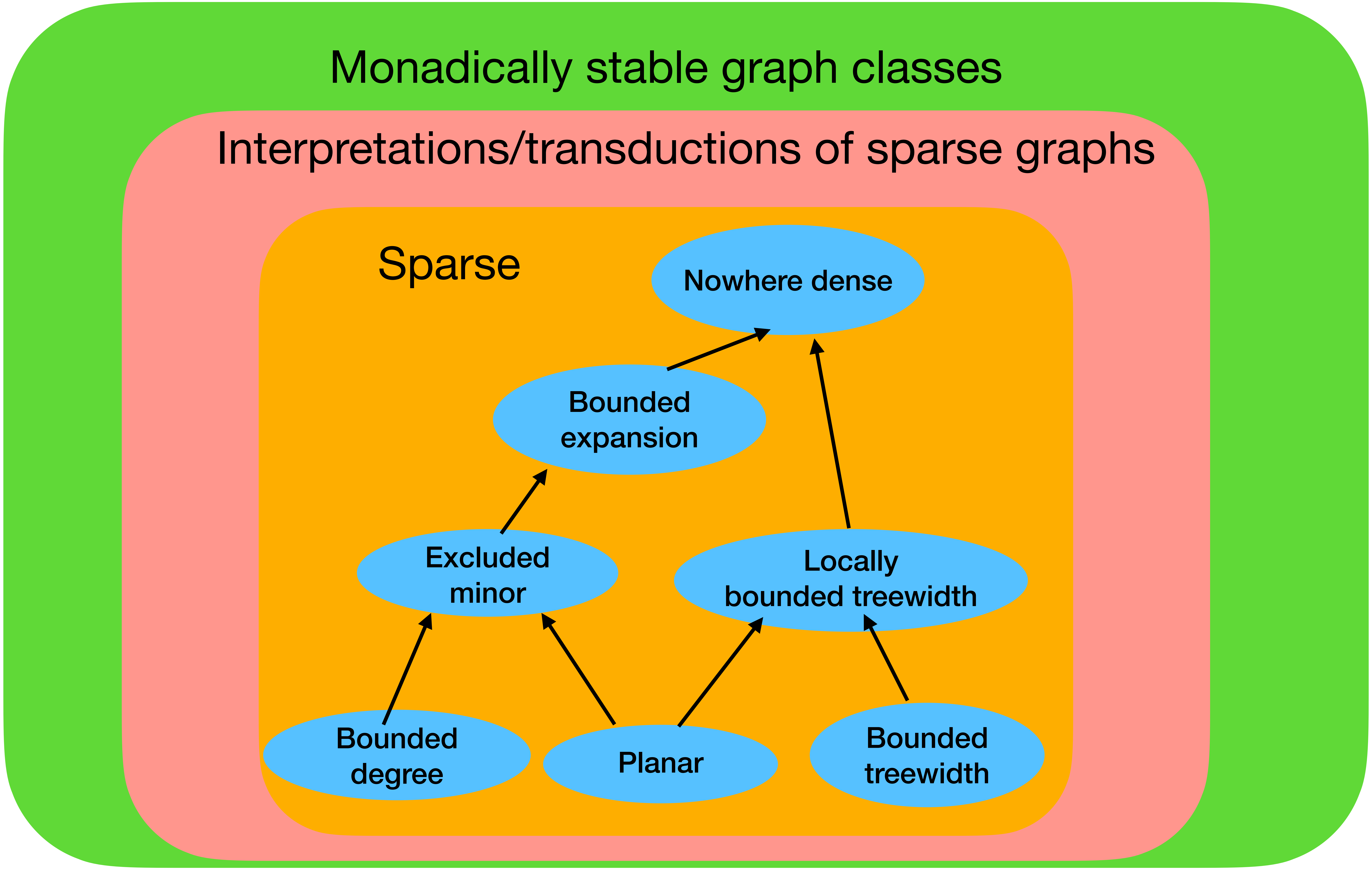
Open problems

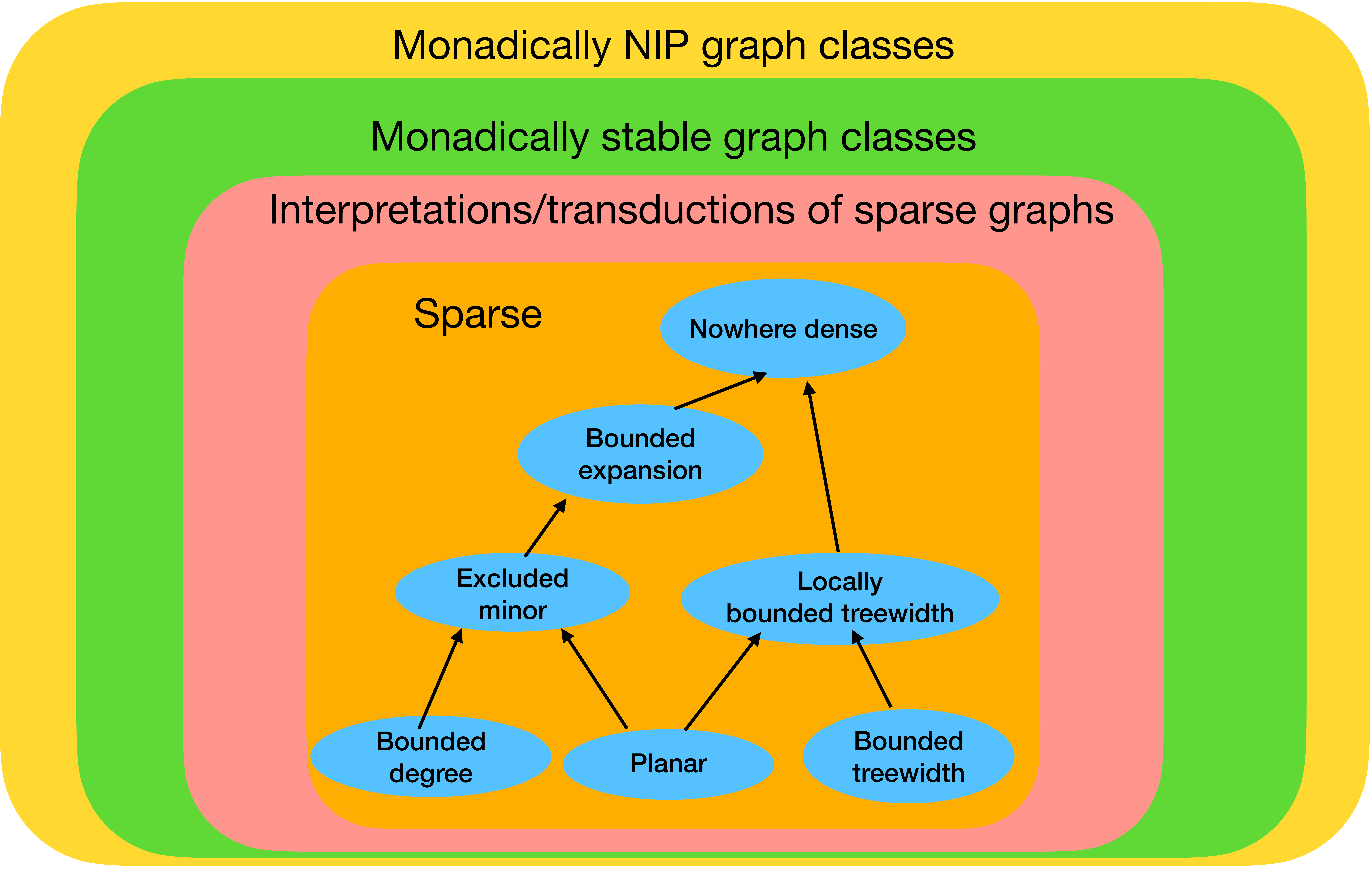
Model checking algorithms for interpretations of other classes of sparse graphs.

Approximate interpretation reversal for interpretations of classes of sparse graphs.

Conjecture:
Let $\mathscr{C}$ be a class of graphs such that one cannot interpret every graph in $\mathscr{C}$ (for every interpretation formula ψ there exists graph G such that $G \notin \mathscr{C}$). Then $\mathscr{C}$ has an efficient model checking algorithm.


Interpretations/transductions of sparse graphs
Sparse
Nowhere dense
Bounded expansion
Excluded minor
Locally bounded treewidth
Bounded degree
Planar
Bounded treewidth


Monadically stable graph classes
Interpretations/transductions of sparse graphs
Sparse
Nowhere dense
Bounded expansion
Excluded minor
Locally bounded treewidth
Bounded degree
Planar
Bounded treewidth


Monadically NIP graph classes
Monadically stable graph classes
Interpretations/transductions of sparse graphs
Sparse
Nowhere dense
Bounded expansion
Excluded minor
Locally bounded treewidth
Bounded degree
Planar
Bounded treewidth