Query enumeration over Nowhere-Dense databases

Alexandre Vigny

Join work with: Arnaud Durand, Nicole Schweikardt, Luc Segoufin

September 15, 2017

Introduction

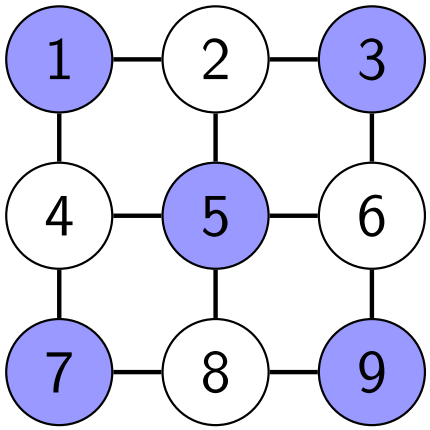
- Query q — *small*
- Database D — *huge*
- Compute $q(D)$ — *gigantic*

Examples :

query q

$$q(x,y) := \exists z(B(x) \wedge E(x,z) \wedge \neg E(y,z))$$

database D



solutions $q(D)$

{(1,2) (1,3) (1,4)
(1,6) (1,7) $\cdots$
(3,1) (3,2) (3,4)
(3,6) (3,7) $\cdots$
$\cdots$ }

Enumeration

Input : $\|D\| := n \quad \& \quad \|q\| := k$ (computation with RAM)

Goal : output solutions one by one (no repetition)

- STEP 1: Preprocessing

  Prepare the enumeration : Database $D \longrightarrow$ Index I

  Preprocessing time : $f(k) \cdot n \rightsquigarrow O(n)$

- STEP 2 : Enumeration

  Enumerate the solutions : Index $I \longrightarrow \overline{x_1}\ ,\ \overline{x_2}\ ,\ \overline{x_3}\ ,\ \overline{x_4}\ ,\ \cdots$

  Delay : $O(f(k)) \rightsquigarrow O(1)$

Constant delay enumeration after linear preprocessing

Example

Input :

- Database $D := \langle \{1, \cdots, n\}; E \rangle$ $\quad\quad \|D\| = |E| \quad (E \subseteq D \times D)$
- Query $q(x, y) := \neg E(x, y)$

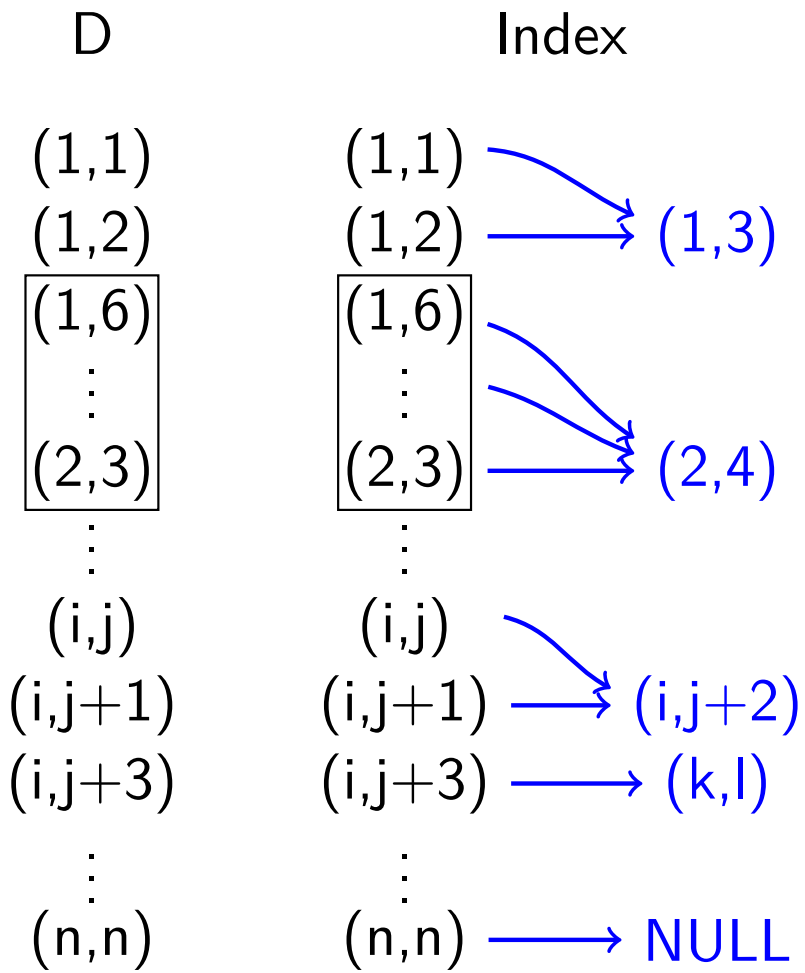
D

(1,1)
(1,2)
(1,6)
⋮
(2,3)
⋮
(i,j)
(i,j+1)
(i,j+3)
⋮
(n,n)

Example

Input :

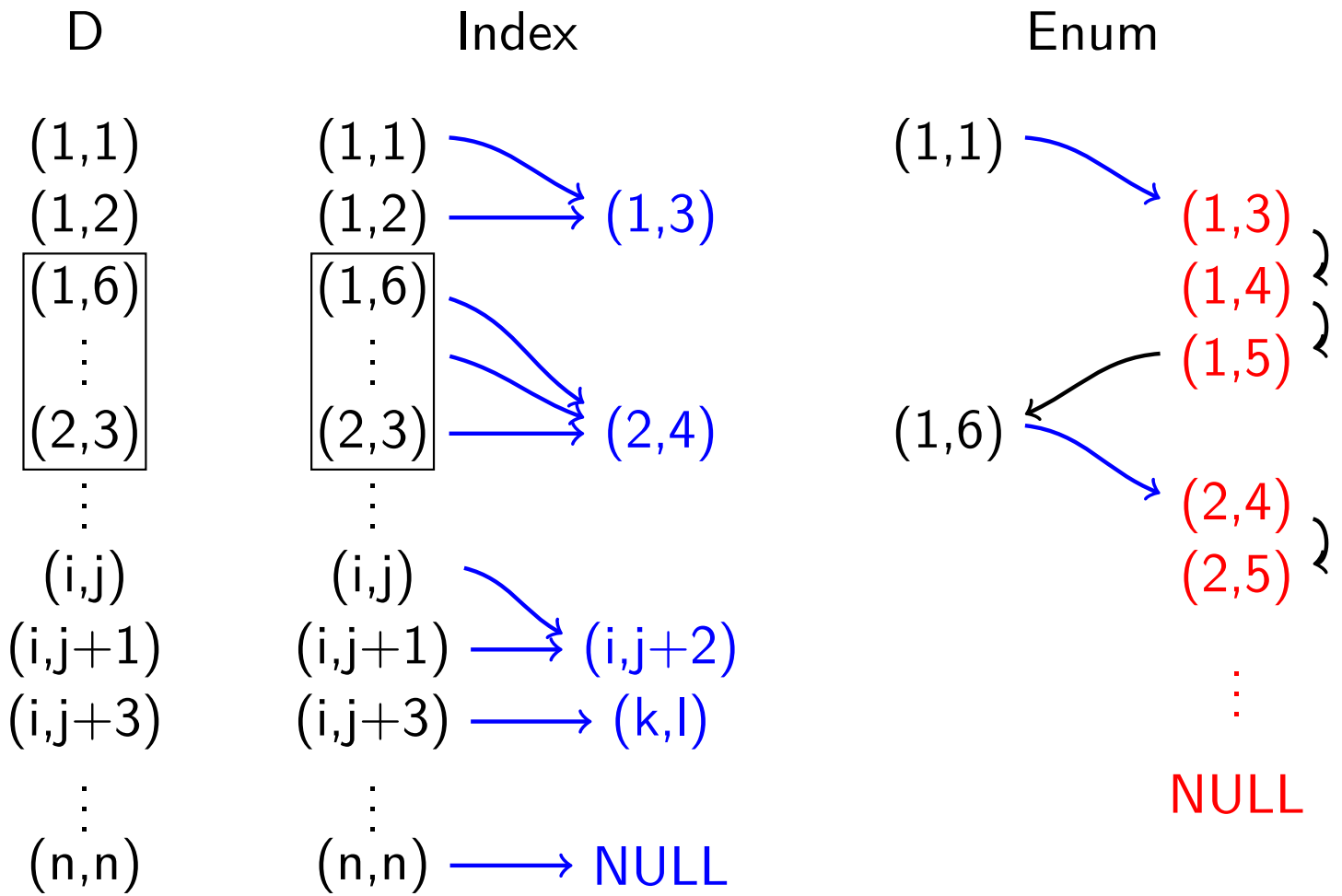
- Database $D := \langle \{1, \cdots, n\}; E \rangle$ $\quad \|D\| = |E| \quad (E \subseteq D \times D)$
- Query $q(x, y) := \neg E(x, y)$



Example

Input :

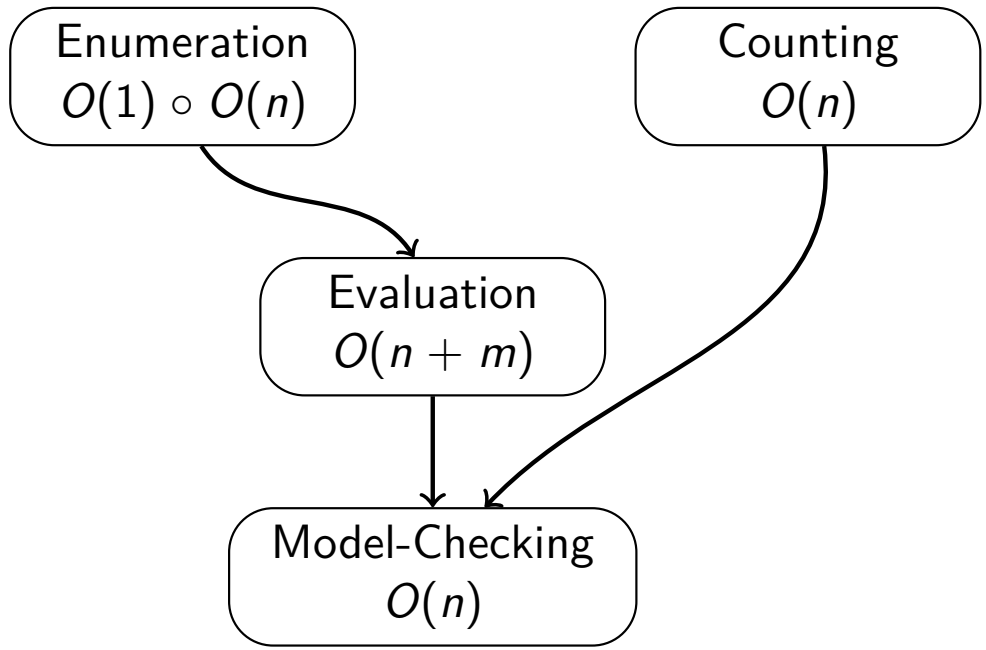
- Database $D := \langle \{1, \cdots, n\}; E \rangle$ $\quad \|D\| = |E| \quad (E \subseteq D \times D)$
- Query $q(x, y) := \neg E(x, y)$



Other problems

For **FO** queries over a class $\mathcal{C}$ of databases.

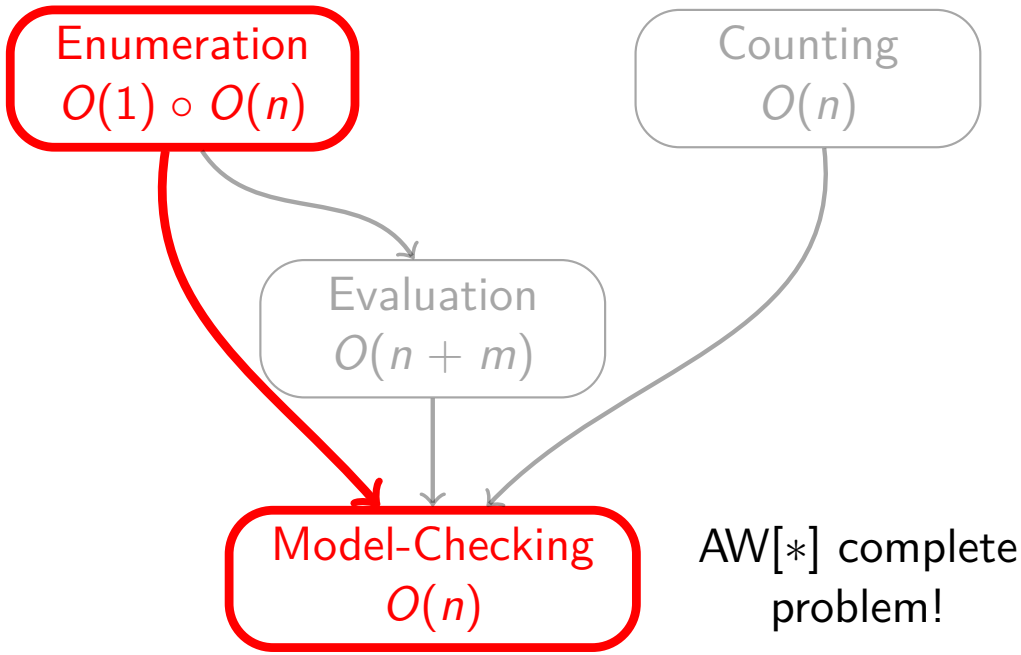
Model-Checking	:	Is this true ?	$O(n)$
Enumeration	:	Enumerate the solutions	$O(1) \circ O(n)$
Counting	:	How many solutions ?	$O(n)$
Evaluation	:	Compute the entire set	$O(n+m)$



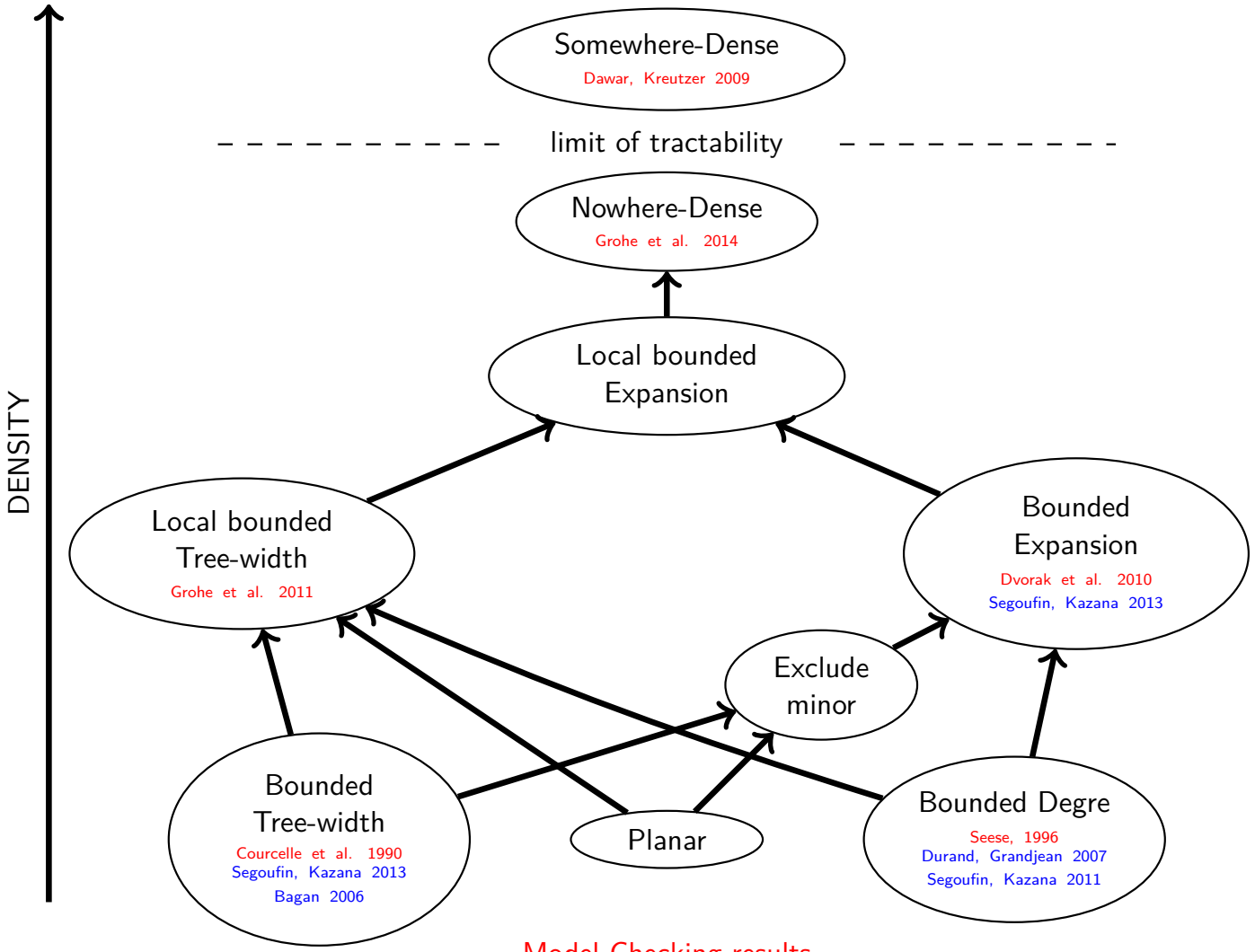
Other problems

For **FO** queries over a class $\mathcal{C}$ of databases.

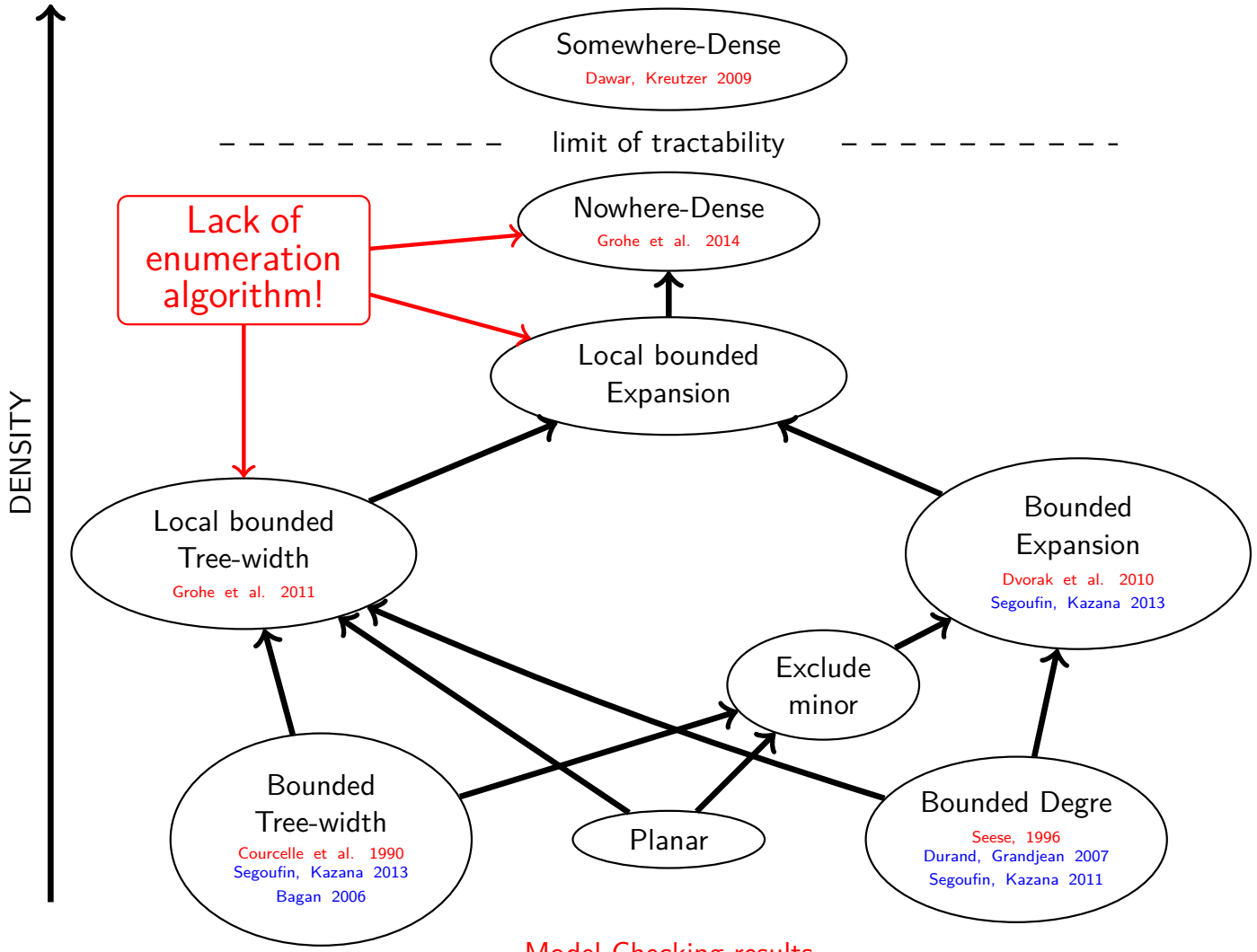
Model-Checking	:	Is this true ?	$O(n)$
Enumeration	:	Enumerate the solutions	$O(1) \circ O(n)$
Counting	:	How many solutions ?	$O(n)$
Evaluation	:	Compute the entire set	$O(n+m)$



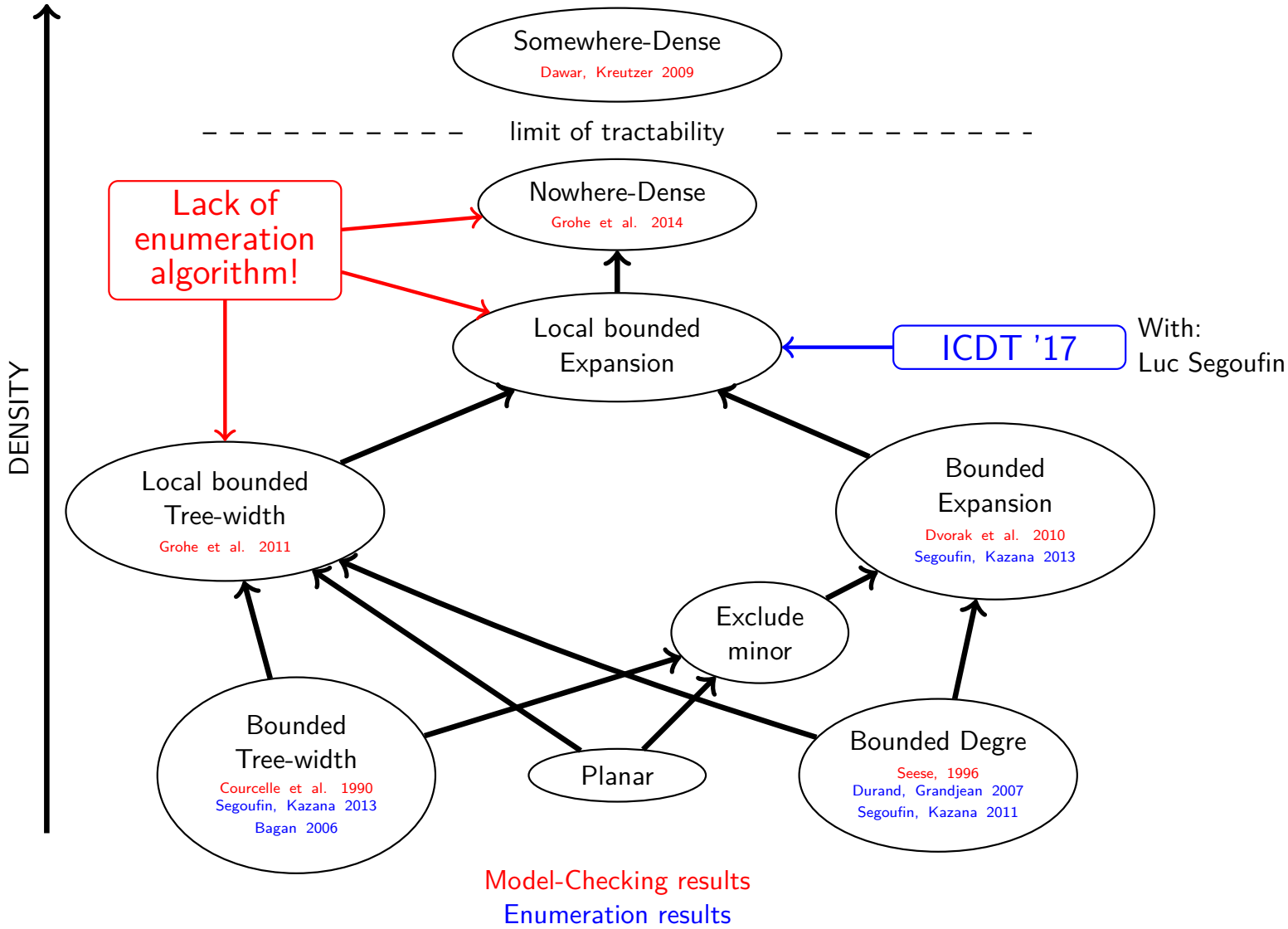
Classes of graphs closed under taking sub-graphs



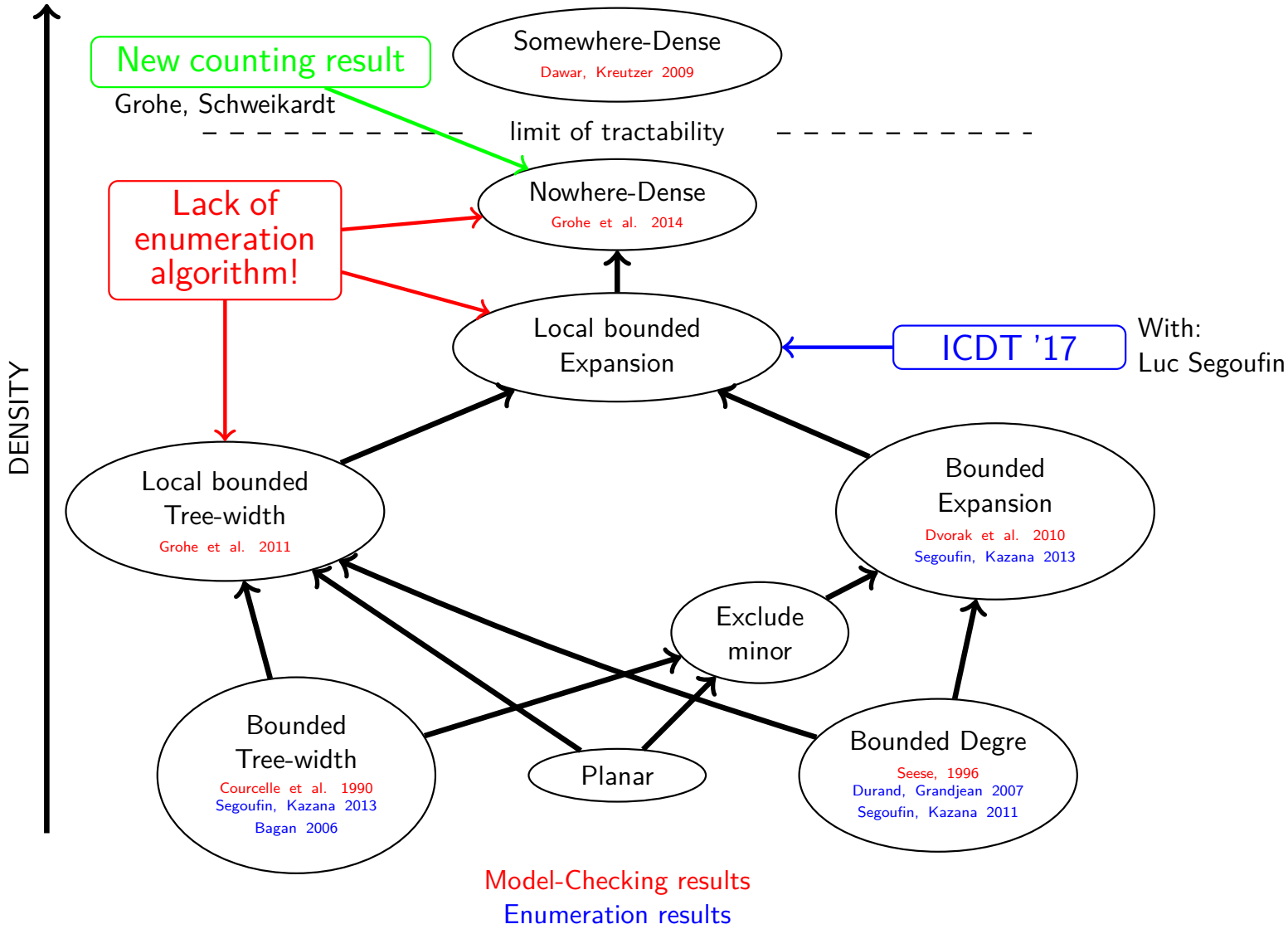
Classes of graphs closed under taking sub-graphs



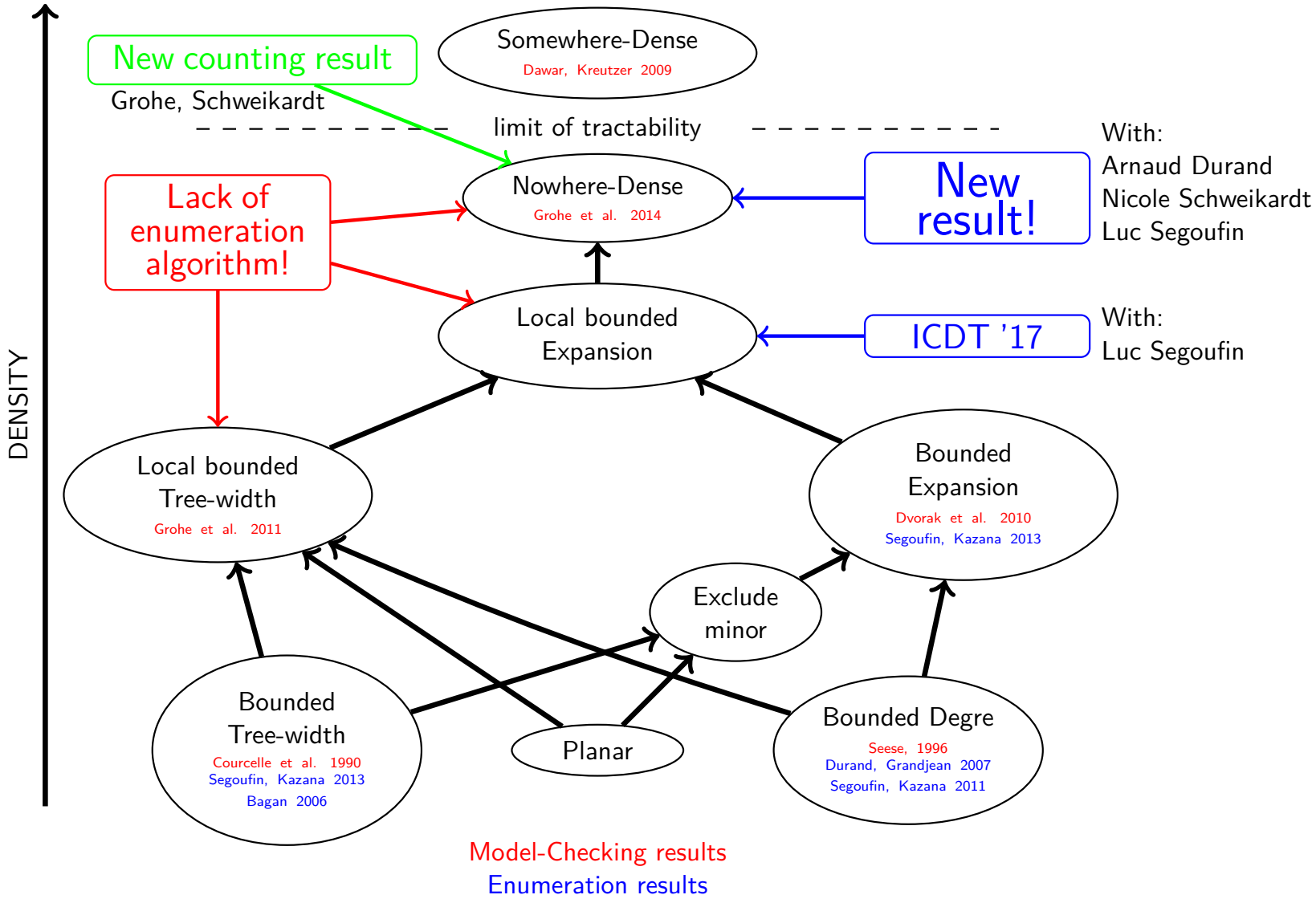
Classes of graphs closed under taking sub-graphs



Classes of graphs closed under taking sub-graphs



Classes of graphs closed under taking sub-graphs



Tools and results (unpublished yet)

We use :

- A new "Hanf like" normal form for FO queries.[1]
- The algorithm for the model checking.[2]
- Neighbourhood cover.[2]
- Game characterization of Nowhere-Dense classes.[2]
- Short-cut pointers dedicated to the enumeration.[3]

We can :

- Enumerate with constant delay after pseudo-linear preprocessing.
- Test in constant time after pseudo-linear preprocessing.

---

[1] Grohe, Schweikardt '17

[2] Grohe, Kreutzer, Siebertz '14

[3] Segoufin, V. '17

Future work

Enumeration with update:
What happens if a small change occurs after the preprocessing ?

Existing results for: words, graphs with bounded tree-width or bounded degree.

Future work

Enumeration with update:
What happens if a small change occurs after the preprocessing ?

Existing results for: words, graphs with bounded tree-width or bounded degree.

Thank you !

Questions ?