Query enumeration over Nowhere-Dense databases

Alexandre Vigny

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

September 15, 2017

Introduction

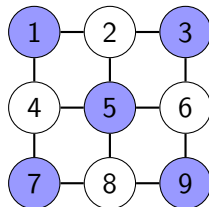- Query q *small*
- Database D *huge*
- Compute $q(D)$ *gigantic*

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 : $\|D\| := n$ & $\|q\| := k$ (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 : $O(f(k)) \rightsquigarrow O(1)$

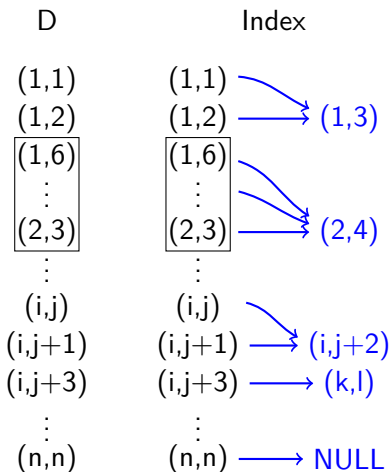**Constant delay enumeration after linear preprocessing**

Example

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

$$D$$

$$(1,1)$$
$$(1,2)$$
$$\boxed{\begin{array}{c} (1,6) \\ \vdots \\ (2,3) \end{array}}$$
$$\vdots$$
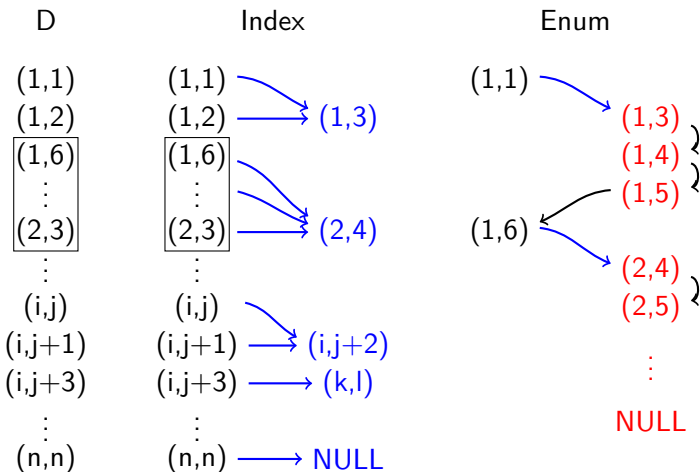$$(i,j)$$
$$(i,j+1)$$
$$(i,j+3)$$
$$\vdots$$
$$(n,n)$$

Example

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

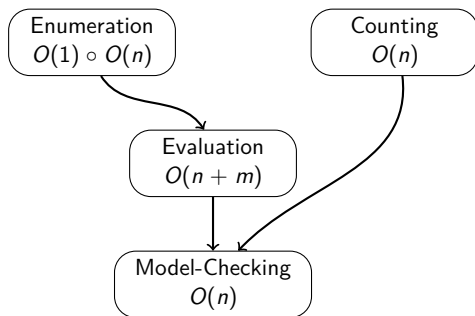| D	Index
(1,1)	(1,1)
(1,2)	(1,2) ⟶ (1,3)
(1,6)	(1,6)
⋮	⋮
(2,3)	(2,3) ⟶ (2,4)
⋮	⋮
(i,j)	(i,j)
(i,j+1)	(i,j+1) ⟶ (i,j+2)
(i,j+3)	(i,j+3) ⟶ (k,l)
⋮	⋮
(n,n)	(n,n) ⟶ NULL

Example

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

Other problems

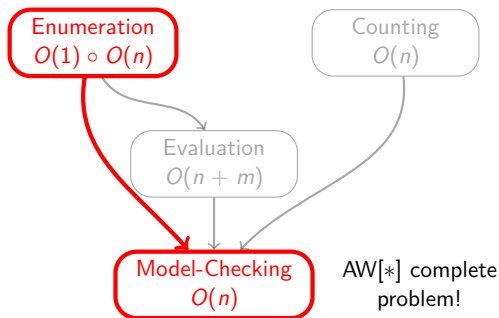For **FO** queries over a class $\mathcal{C}$ of databases.

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.

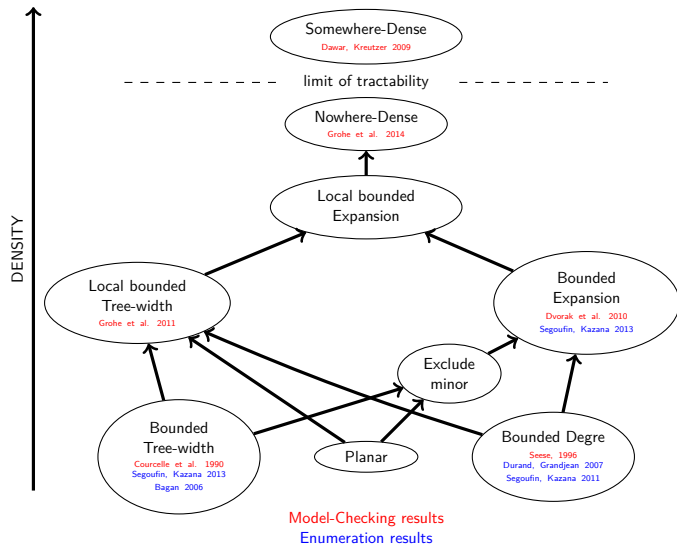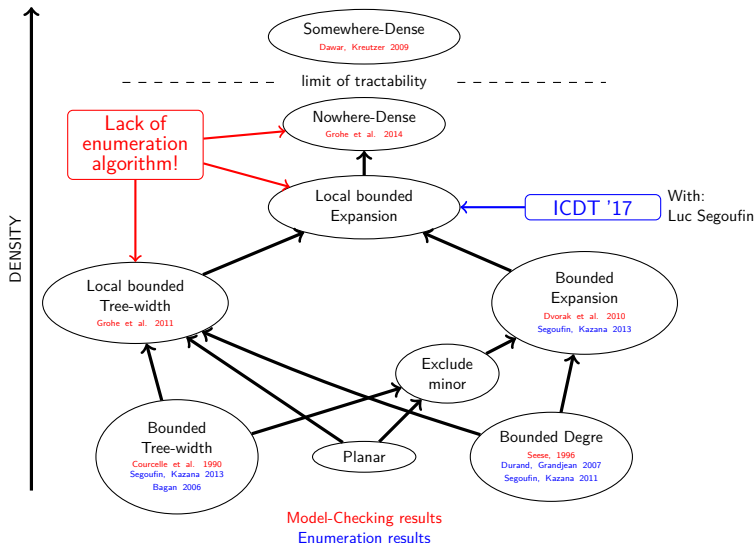|			
Model-Checking	:	Is this true ?	$O(n)$
Enumeration	:	Enumerate the solutions	$O(1) \circ O(n)$
Counting	:	How many solutions ?	$O(n)$
Evaluation	:	Compute the entire set	$O(n + m)$

Enumeration
$O(1) \circ O(n)$

Counting
$O(n)$

Evaluation
$O(n + m)$

Model-Checking
$O(n)$

AW[∗] complete problem!

Classes of graphs closed under taking sub-graphs

Model-Checking results
Enumeration results

Classes of graphs closed under taking sub-graphs

Classes of graphs closed under taking sub-graphs

DENSITY

Somewhere-Dense
Dawar, Kreutzer 2009

- - - - - - - - - limit of tractability - - - - - - - - -

Nowhere-Dense
Grohe et al. 2014

Lack of
enumeration
algorithm!

Local bounded
Expansion

ICDT '17   With:
Luc Segoufin

Local bounded
Tree-width
Grohe et al. 2011

Bounded
Expansion
Dvorak et al. 2010
Segoufin, Kazana 2013

Exclude
minor

Bounded
Tree-width
Courcelle et al. 1990
Segoufin, Kazana 2013
Bagan 2006

Planar

Bounded Degre
Seese, 1996
Durand, Grandjean 2007
Segoufin, Kazana 2011

Model-Checking results
Enumeration results

Classes of graphs closed under taking sub-graphs
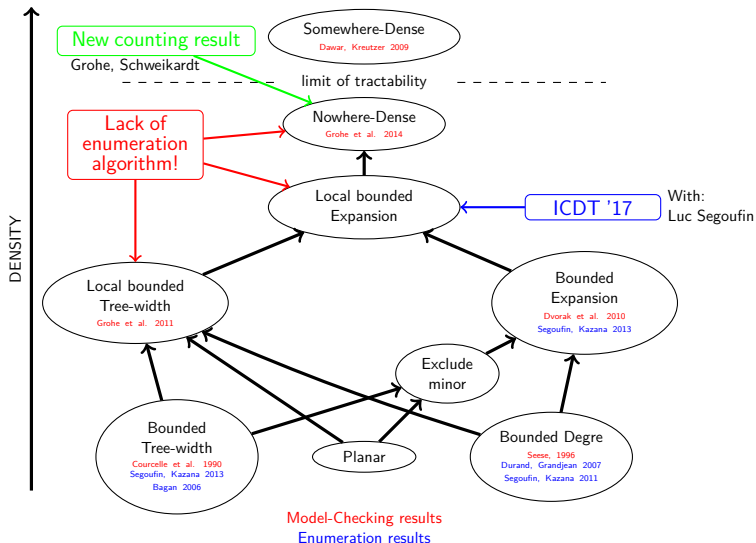
Classes of graphs closed under taking sub-graphs

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 ?