Query enumeration
&
Nowhere-dense graphs

AGGREG 2016

Alexandre Vigny

July 11-12, 2016

About me

- Alexandre Vigny
- I started my Ph.D in October 2015
- My advisors are :
  - Arnaud Durand (IMJ-PRG, Paris 7)
  - Luc Segoufin (LSV, ENS Cachan)

Introduction

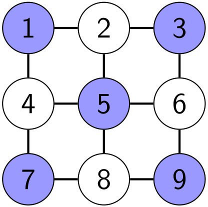
- Query q *small*
- Database D *huge*
- Compute $q(D)$ *gigantic*

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 : $\|D\| := n \quad \& \quad \|q\| := k \qquad (k \ll n)$

Goal : output solutions one by one

- 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 $I \longrightarrow \overline{x_1}\ , \overline{x_2}\ , \overline{x_3}\ , \overline{x_4}\ , \cdots$

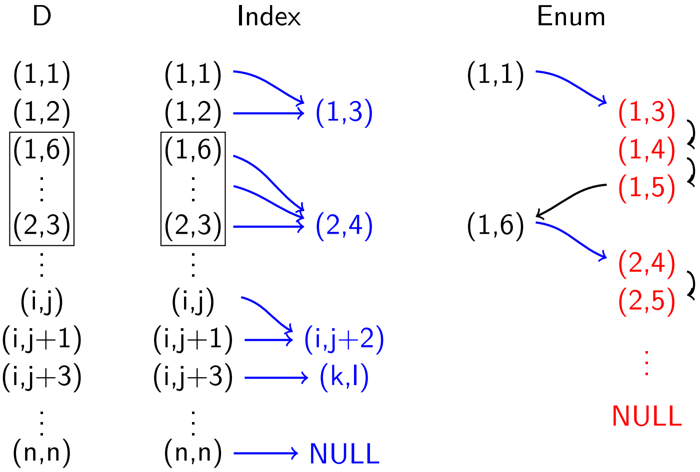
  Delay : $O(f(k)) \rightsquigarrow O(1)$

Constant delay enumeration after linear preprocessing $(CD \circ Lin)$

Example

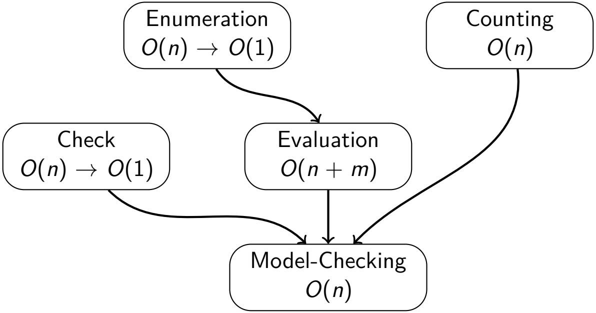
Database $D := \langle\{1,\cdots,n\}; E\rangle \qquad \|D\| = |E| \quad (E \subseteq D \times D)$

Query $q(x,y) := \neg E(x,y)$



Other problems

Model-Checking	:	Is there a solution ?	$O(n)$
Counting	:	How many solutions ?	$O(n)$
Check	:	Is this tuple a solution ?	$O(n) \to O(1)$
Evaluation	:	Compute the entire set	$O(n+m)$



Restrictions are needed

Constant-delay Enumeration $\Longrightarrow$ Linear Model-Checking

Under some complexity hypothesis, the Model-Checking is not doable in polynomial time.

$\Downarrow$

Restricted databases or/and queries

Bonded degree, planar $\cdots$	MSO, quantifier free $\cdots$
Nowhere-dense	**First Order**

Nowhere-dense

- What is it ?

- Is it robust ?

- Why Nowhere-dense ?

Nowhere-dense

- What is it ?
  - Definition by J. Nešetřil & P. Ossona de Mendez.[1]
  - Very large class of graphs :
    Planar, bounded degree, bounded tree width, bounded expansion ...
- Is it robust ?
- Why Nowhere-dense ?

[1] On nowhere dense graphs, Eur. J. Comb. 2011.

Nowhere-dense

- What is it ?
  - Definition by J. Nešetřil & P. Ossona de Mendez.[1]
  - Very large class of graphs :
    Planar, bounded degree, bounded tree width, bounded expansion ...
- Is it robust ?   YES !
  - 10 definitions with asymptotic ratio (edges)/(vertices).
  - Local exclusion of minors.
  - Algorithmic definitions (tree width colouring, augmentation ...).
  - Wining strategy in a destruction/construction game.
- Why Nowhere-dense ?

[1] On nowhere dense graphs, Eur. J. Comb. 2011.

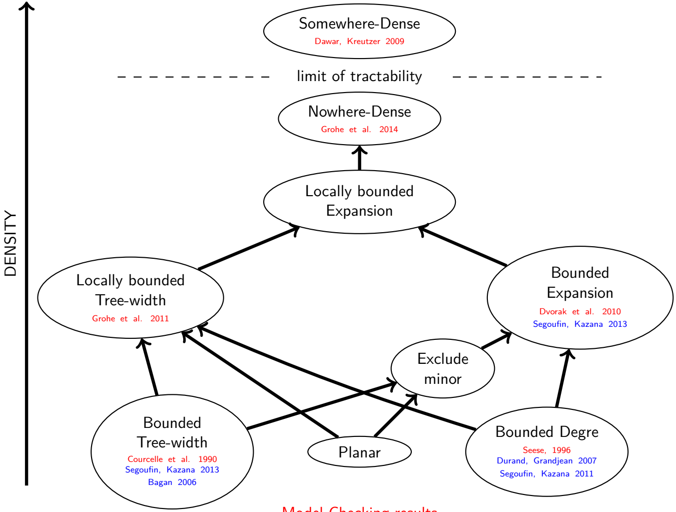
Nowhere-dense

- What is it ?
  - Definition by J. Nešetřil & P. Ossona de Mendez.[1]
  - Very large class of graphs :
    Planar, bounded degree, bounded tree width, bounded expansion ...
- Is it robust ? YES !
  - 10 definitions with asymptotic ratio (edges)/(vertices).
  - Local exclusion of minors.
  - Algorithmic definitions (tree width colouring, augmentation ...).
  - Wining strategy in a destruction/construction game.
- Why Nowhere-dense ?
  - Generalize previous work.
  - The model-checking is linear, Grohe, et al.[2]

[1] On nowhere dense graphs, Eur. J. Comb. 2011.

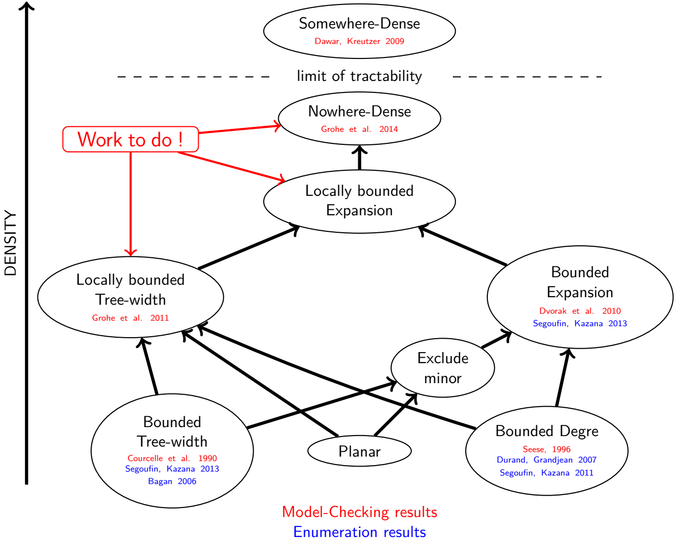
[2] Deciding first-order properties of nowhere dense graphs, STOC 2014.

Classes of graphs closed under taking sub-graphs

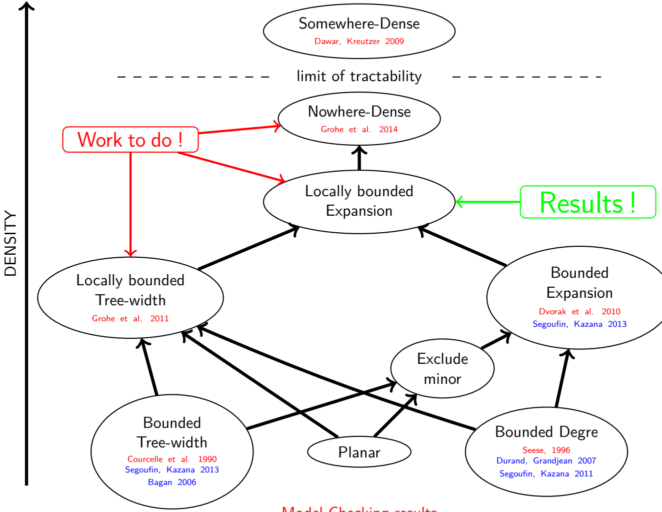


Model-Checking results
Enumeration results

Classes of graphs closed under taking sub-graphs



Classes of graphs closed under taking sub-graphs



Model-Checking results

Enumeration results

Tools

- Gaifman theorem.
- Neighbourhood cover.
- Enumeration for graphs with **Bounded expansion**. W.Kazana, L.Segoufin.[1]
- Short-cut pointers dedicated to the enumeration.

[1] Enumeration of first-order queries on classes of structures with bounded expansion, PODS 2013

First results

$G \in \mathcal{C}$ a class with locally bounded expansion, φ a first-order query.

Theorem 1

New proof that the model checking problem is doable in pseudo-linear time.

Theorem 2

After an pseudo-linear preprocessing, we can, for every tuple $\overline{a}$, compute $NEXT(\overline{a}) := \min\{\overline{b} \in G^k \mid G \models \varphi(\overline{b}) \wedge \overline{a} \leq \overline{b}\}$ in constant time.

This implies the enumeration and the check problem.

Theorem 3

The counting problem is doable in pseudo-linear time.

Future work

- Generalize the Nowhere-Dense case.
- Enumeration with update.

Thank you !