Constant delay enumeration for FO queries over databases with local bounded expansion

Luc Segoufin ¹  Alexandre Vigny²

¹ENS Ulm, Paris
²Université Paris Diderot, Paris

October 16, 2017
Introduction

- Query $q$
- Database $D$
- Compute $q(D)$

Examples:

$\text{query } q$
$q(x, y) := \exists z (B(x) \land E(x, z) \land \lnot E(y, z))$

$\text{database } D$

$\text{solutions } q(D)$
$(1,2) (1,3) (1,4) (1,6) (1,7) \ldots$
$(3,1) (3,2) (3,4) (3,6) (3,7) \ldots$
$\ldots \}$
Enumeration

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

Goal: output solutions one by one (no repetition)

STEP 1: Preprocessing

Prepare the enumeration: Database $D \rightarrow \text{Index } I$

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

STEP 2: Enumeration

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

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

Constant delay enumeration after linear preprocessing
Example 1

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

\[
D
\]

\[
\begin{array}{cccccc}
(1,1) & (1,2) & (1,6) & \cdots & (i,j) & (i,j+1) \\
(1,1) & (1,2) & (1,6) & \cdots & (i,j) & (i,j+1) \\
(1,1) & (1,3) & (1,4) & (1,5) & (1,6) & (2,4) & (2,5) \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
(1,1) & (1,3) & (1,4) & (1,5) & (1,6) & (2,4) & (2,5) \\
\end{array}
\]
Example 1

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

<table>
<thead>
<tr>
<th>D</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1)</td>
<td>(1,1)</td>
</tr>
<tr>
<td>(1,2)</td>
<td>(1,2)</td>
</tr>
<tr>
<td>(1,6)</td>
<td>(1,6)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>(2,3)</td>
<td>(2,3)</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>(i,j)</td>
<td>(i,j)</td>
</tr>
<tr>
<td>(i,j+1)</td>
<td>(i,j+1)</td>
</tr>
<tr>
<td>(i,j+3)</td>
<td>(i,j+3)</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>(n,n)</td>
<td>(n,n)</td>
</tr>
<tr>
<td></td>
<td>NULL</td>
</tr>
</tbody>
</table>
Example 1

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

<table>
<thead>
<tr>
<th>D</th>
<th>Index</th>
<th>Enum</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1)</td>
<td>(1,1)</td>
<td>(1,1)</td>
</tr>
<tr>
<td>(1,2)</td>
<td>(1,2)</td>
<td></td>
</tr>
<tr>
<td>(1,6)</td>
<td>(1,6)</td>
<td>(1,3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1,4)</td>
</tr>
<tr>
<td>(2,3)</td>
<td>(2,3)</td>
<td>(1,5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1,6)</td>
</tr>
<tr>
<td>(i,j)</td>
<td>(i,j)</td>
<td></td>
</tr>
<tr>
<td>(i,j+1)</td>
<td>(i,j+1)</td>
<td>(i,j+2)</td>
</tr>
<tr>
<td>(i,j+3)</td>
<td>(i,j+3)</td>
<td>(k,l)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(n,n)</td>
<td>(n,n)</td>
<td>NULL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Alexandre Vigny

Enumeration & local-bounded expansion

October 16, 2017 4 / 12
Example 2

Input :
- Database $D := \langle \{1, \cdots, n\}; E_1; E_2 \rangle$ \quad \|D\| = |E_1| + |E_2| \quad (E_i \subseteq D \times D)
- Query $q(x, y) := \exists z, E_1(x, z) \land E_2(z, y)$
Example 2

Input:
- Database \( D := \langle\{1, \cdots, n\}; E_1; E_2\rangle \) \( \|D\| = |E_1| + |E_2| \) \( (E_i \subseteq D \times D) \)
- Query \( q(x, y) := \exists z, E_1(x, z) \land E_2(z, y) \)

\[ B : \text{Adjacency matrix of } E_2 \]
\[
\begin{pmatrix}
E_2(1, 1) & \cdots & E_2(1, y) & \cdots & E_2(1, n) \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
E_2(z, 1) & \cdots & E_2(z, y) & \cdots & E_2(z, n) \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
E_2(n, 1) & \cdots & E_2(n, y) & \cdots & E_2(n, n)
\end{pmatrix}
\]

\[ A : \text{Adjacency matrix of } E_1 \]
\[
\begin{pmatrix}
E_1(1, 1) & \cdots & E_1(1, i) & \cdots & E_1(1, n) \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
E_1(x, 1) & \cdots & E_1(x, z) & \cdots & E_1(x, n) \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
E_1(n, 1) & \cdots & E_1(n, z) & \cdots & E_1(n, n)
\end{pmatrix}
\]

\[ C : \text{Result matrix} \]
\[
\begin{pmatrix}
q(1, 1) & \cdots & q(1, y) & \cdots & q(1, n) \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
q(x, 1) & \cdots & q(x, y) & \cdots & q(x, n) \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
q(n, 1) & \cdots & q(n, y) & \cdots & q(n, n)
\end{pmatrix}
\]
Example 2

Input:
- Database $D := \langle \{1, \cdots, n\}; E_1; E_2 \rangle$  \[ \|D\| = |E_1| + |E_2| \quad (E_i \subseteq D \times D) \]
- Query $q(x, y) := \exists z, E_1(x, z) \land E_2(z, y)$

Compute the set of solutions

\[
\begin{pmatrix}
E_2(1, 1) & \cdots & E_2(1, y) & \cdots & E_2(1, n) \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
E_2(z, 1) & \cdots & E_2(z, y) & \cdots & E_2(z, n) \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
E_2(n, 1) & \cdots & E_2(n, y) & \cdots & E_2(n, n)
\end{pmatrix}
\]

$=$

boolean matrix multiplication

\[
\begin{pmatrix}
E_1(1, 1) & \cdots & E_1(1, i) & \cdots & E_1(1, n) \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
E_1(x, 1) & \cdots & E_1(x, z) & \cdots & E_1(x, n) \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
E_1(n, 1) & \cdots & E_1(n, z) & \cdots & E_1(n, n)
\end{pmatrix}
\begin{pmatrix}
q(1, 1) & \cdots & q(1, y) & \cdots & q(1, n) \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
q(x, 1) & \cdots & q(x, y) & \cdots & q(x, n) \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
q(n, 1) & \cdots & q(n, y) & \cdots & q(n, n)
\end{pmatrix}
\]

$A$ : Adjacency matrix of $E_1$  \hfill  $C$ : Result matrix
Example 2

Input:
- Database $D := \{1, \cdots, n\}; E_1; E_2$ \quad \|D\| = |E_1| + |E_2| \quad (E_i \subseteq D \times D)$
- Query $q(x, y) := \exists z, E_1(x, z) \land E_2(z, y)$

\[ A : \text{Adjacency matrix of } E_1 \]
\[ B : \text{Adjacency matrix of } E_2 \]
\[ C : \text{Result matrix} \]

\[ \begin{pmatrix}
E_2(1, 1) & \cdots & E_2(1, y) & \cdots & E_2(1, n) \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
E_2(z, 1) & \cdots & E_2(z, y) & \cdots & E_2(z, n) \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
E_2(n, 1) & \cdots & E_2(n, y) & \cdots & E_2(n, n)
\end{pmatrix} \]

\[ \begin{pmatrix}
E_1(1, 1) & \cdots & E_1(1, i) & \cdots & E_1(1, n) \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
E_1(x, 1) & \cdots & E_1(x, z) & \cdots & E_1(x, n) \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
E_1(n, 1) & \cdots & E_1(n, z) & \cdots & E_1(n, n)
\end{pmatrix} \]

\[ \begin{pmatrix}
q(1, 1) & \cdots & q(1, y) & \cdots & q(1, n) \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
q(x, 1) & \cdots & q(x, y) & \cdots & q(x, n) \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
q(n, 1) & \cdots & q(n, y) & \cdots & q(n, n)
\end{pmatrix} \]

- Linear preprocessing : $O(n^2)$
- Number of solutions : $O(n^2)$
- Algorithm for the boolean matrix multiplication in $O(n^2)$
- Conjecture : "There are no algorithm for the boolean matrix multiplication working in time $O(n^2)$."
Example 2

Input:
- Database $D := \langle \{1, \ldots, n\}; E_1; E_2 \rangle$  $\|D\| = |E_1| + |E_2|$  ($E_i \subseteq D \times D$)
- Query $q(x, y) := \exists z, E_1(x, z) \land E_2(z, y)$

This query cannot be enumerated with constant delay

1. Unless there is a breakthrough with the boolean matrix multiplication.
Example 2

Input:
- Database $D := \langle \{1, \ldots, n\}; E_1; E_2 \rangle$ \quad $\|D\| = |E_1| + |E_2| \quad (E_i \subseteq D \times D)$
- Query $q(x, y) := \exists z, E_1(x, z) \land E_2(z, y)$

This query cannot be enumerated with constant delay$^1$

We need to put restrictions on queries and/or databases.

---

$^1$ Unless there is a breakthrough with the boolean matrix multiplication.
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 $O(n + m)$

AW[∗] complete problem!
Classes of graphs closed under taking sub-graphs

- Bounded Tree-width
  - Courcelle et al. 1990

- Bounded Degree
  - Seese, 1996

Model-Checking results

- Durand, Grandjean 2007
- Segoufin, Kazana 2011

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

- Grohe et al. 2011
- Dvorak et al. 2010
- Segoufin, Kazana 2013

- Grohe et al. 2014

- Dawar, Kreutzer 2009
- Somewhere-Dense
  - Limit of tractability
  - Lack of enumerationalgorithm!

Nowhere-Dense
  - Grohe et al. 2014
  - Local bounded Expansion

Planar
  - Excludeminor

- ICDT '17
- With: Luc Segoufin

Alexandre Vigny
Enumeration & local-bounded expansion
October 16, 2017
Classes of graphs closed under taking sub-graphs

Model-Checking results

- Bounded Degree
  - Seese, 1996

- Excludeminer
  - Dvorak et al. 2010

- Bounded Expansion
  - Dvorak et al. 2010
  - Segoufin, Kazana 2013

- Local bounded Expansion

- Local bounded Tree-width
  - Courcelle et al. 1990
  - Grohe et al. 2011

- Bounded Tree-width
  - Courcelle et al. 1990
  - Grohe et al. 2011

- Planar

- Nowhere-Dense
  - Grohe et al. 2014

- Bounded Expansion
  - Dvorak et al. 2010

- Somewhere-Dense
  - Dawar, Kreutzer 2009

Limit of tractability
Lack of enumerationalgorithm!

With: Luc Segoufin

Model-Checking results
Classes of graphs closed under taking sub-graphs

Model-Checking results

- Somewhere-Dense
  - Dawar, Kreutzer 2009

- Nowhere-Dense
  - Grohe et al. 2014

- Limit of tractability

- Local bounded Expansion

- Somewhere-Dense
  - Dawar, Kreutzer 2009

- Nowhere-Dense
  - Grohe et al. 2014

- Local bounded Expansion

- Bounded Expansion
  - Dvorak et al. 2010

- Local bounded Tree-width
  - Grohe et al. 2011

- Bounded Tree-width
  - Courcelle et al. 1990

- Bounded Expansion
  - Dvorak et al. 2010

- Bounded Degre
  - Seese, 1996

- Planar

- Exclude minor

Model-Checking results

- Bounded Degre
  - Seese, 1996

- Local bounded Tree-width
  - Grohe et al. 2011

- Bounded Tree-width
  - Courcelle et al. 1990

- Planar

DENSITY
Classes of graphs closed under taking sub-graphs

- Somewhere-Dense
  - Dawar, Kreutzer 2009
  - limit of tractability

- Nowhere-Dense
  - Grohe et al. 2014

- Local bounded Expansion

- Local bounded Tree-width
  - Grohe et al. 2011

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

- Bounded Expansion
  - Dvorak et al. 2010
  - Segoufin, Kazana 2013

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

- Planar

- Exclude minor

Model-Checking results
Enumeration results
Classes of graphs closed under taking sub-graphs

Model-Checking results
Enumeration results

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

Nowhere-Dense
Grohe et al. 2014

Somewhere-Dense
Dawar, Kreutzer 2009

Local bounded Expansion

Bounded Expansion
Dvorak et al. 2010
Segoufin, Kazana 2013

Excludeminor

Local bounded Tree-width
Grohe et al. 2011

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

Planar

Lack of enumeration algorithm!

DENSITY

Model-Checking results
Enumeration results

Alexandre Vigny
Enumeration & local-bounded expansion
October 16, 2017 7 / 12
Classes of graphs closed under taking sub-graphs

- Somewhere-Dense
  - Dawar, Kreutzer 2009

- Nowhere-Dense
  - Grohe et al. 2014

- Local bounded Expansion

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

- Bounded Expansion
  - Dvorak et al. 2010
  - Segoufin, Kazana 2013

- Bounded Degree
  - Seese, 1996
  - Durand, Grandjean 2007
  - Segoufin, Kazana 2011

- Planar

- Exclude minor

- Limit of tractability

- Model-Checking results
  - Segoufin, Kazana 2011

- Enumeration results
  - Courcelle et al. 1990
  - Segoufin, Kazana 2013

- Lack of enumeration algorithm!

- ICDT '17 With : Luc Segoufin

DENSITY
Local bounded expansion?

Definition : Class of r-neighborhoods

Let $\mathcal{C}$ be a class of graphs, $r \in \mathbb{N}$, $\mathcal{C}_r := \{ N_r^G(a) \mid G \in \mathcal{C}, a \in G \}$

Definition : Local bounded expansion

$\mathcal{C}$ has locally bounded expansion if for all $r$, $\mathcal{C}_r$ as bounded expansion.
### Local bounded expansion?

**Definition : Class of r-neighborhoods**

Let $\mathcal{C}$ be a class of graphs, $r \in \mathbb{N}$, $\mathcal{C}_r := \{N_r^G(a) \mid G \in \mathcal{C}, \ a \in G\}$

**Definition : Local bounded expansion**

$\mathcal{C}$ has locally bounded expansion if for all $r$, $\mathcal{C}_r$ as bounded expansion.

---

**Bounded expansion?**
Local bounded expansion?

Definition: Class of r-neighborhoods
Let \( \mathcal{C} \) be a class of graphs, \( r \in \mathbb{N} \), \( \mathcal{C}_r := \{ N^G_r(a) \mid G \in \mathcal{C}, \ a \in G \} \)

Definition: Local bounded expansion
\( \mathcal{C} \) has locally bounded expansion if for all \( r \), \( \mathcal{C}_r \) as bounded expansion.

Bounded expansion?

Examples
Planar graphs, graphs with bounded degree, bounded tree width, ...

Properties
Bounded in-degree, linear number of edges, nice coloring, ...
Local bounded expansion?

**Definition: Class of r-neighborhoods**

Let $\mathcal{C}$ be a class of graphs, $r \in \mathbb{N}$, $\mathcal{C}_r := \{ N_r^G(a) | G \in \mathcal{C}, a \in G \}$

**Definition: Local bounded expansion**

$\mathcal{C}$ has locally bounded expansion if for all $r$, $\mathcal{C}_r$ has bounded expansion.

Can have a non linear number of edges!

Bounded expansion?

**Examples**

Planar graphs, graphs with bounded degree, bounded tree width, ...

**Properties**

Bounded in-degree, linear number of edges, nice coloring, ...
Our results

Theorem (Segoufin, V. 17’)

Over classes of graphs with *local bounded expansion*, for every FO query, after a pseudo-linear preprocessing, we can:

- enumerate with constant delay every solution.
- test in constant time whether a given tuple is a solution.
- compute in constant time the number of solutions.
A function $f$ is pseudo linear if and only if:

$$\forall \epsilon > 0, \ \exists N_\epsilon \in \mathbb{N}, \ \forall n \in \mathbb{N}, \ n > N_\epsilon \implies f(n) \leq n^{1+\epsilon}$$

$$n \ll n \log^i(n) \ll \text{pseudo-linear} \ll n^{1.0001} \ll n^{\sqrt{n}}$$

“Pseudo-linear $\approx n \log^i(n)$”

“Pseudo-constant $\approx \log^i(n)$”
Tools used

We use:

- Gaifman normal form for FO queries.
- Neighbourhood cover.\(^1\)
- Enumeration for graphs with \textbf{Bounded expansion}.\(^2\)
- New short-cut pointers dedicated to the enumeration.

---

1. Grohe, Kreutzer, Siebertz ’14
2. Segoufin, Kazana. ’13
Future/Current work

- The nowhere-dense case!

- 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/Current work

- The nowhere-dense case!

- 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?