Query Evaluation With Constant Delay

Wojciech Kazana

INRIA Saclay, ENS de Cachan

PhD Thesis Defense

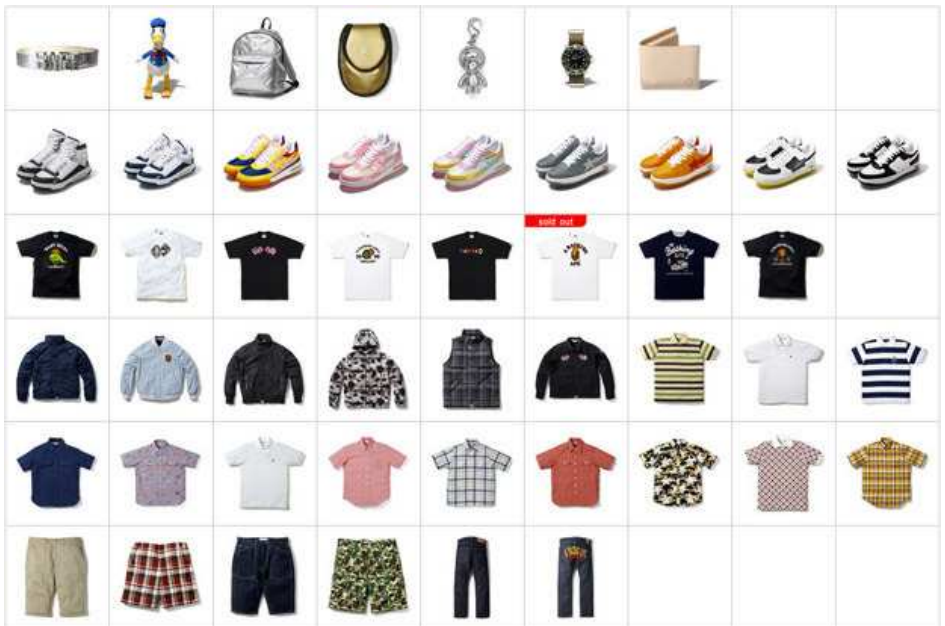
LSV, École normale supérieure de Cachan

September 16, 2013

Introduction – databases

Databases: storage of data and retrieval of information.

1. A store has its list of offered products.
   Can I buy orange shoes?
2. Private collection of photos.
3. Map of a metro system.
4. Social network and its graph.
5. ...



Introduction – databases

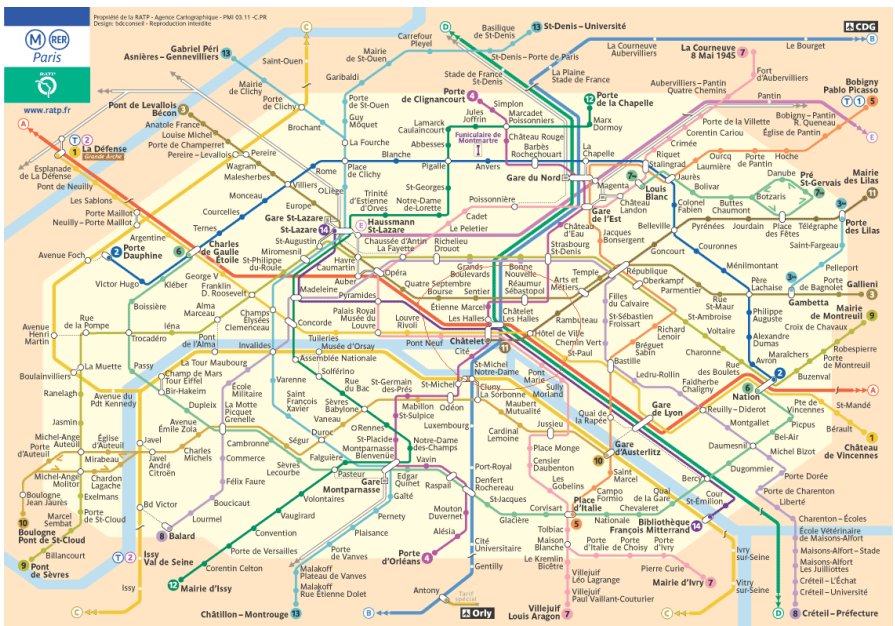
Databases: storage of data and retrieval of information.

1. A store has its list of offered products.
2. Private collection of photos.
   On how many of my photos am I actually present?
3. Map of a metro system.
4. Social network and its graph.
5. ...

Introduction – databases

Databases: storage of data and retrieval of information.

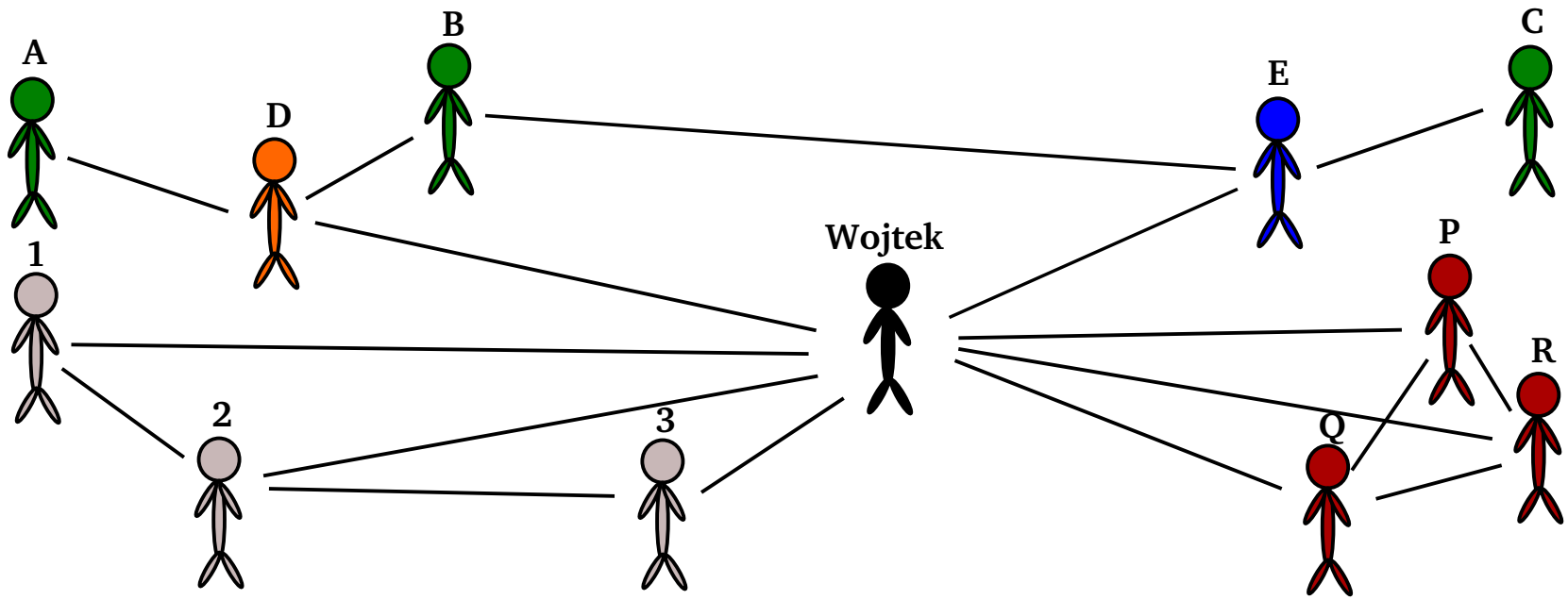
1. A store has its list of offered products.
2. Private collection of photos.
3. Map of a metro system. Can I get from Château d'Eau to Bagneux with just one hop?
4. Social network and its graph.
5. ...



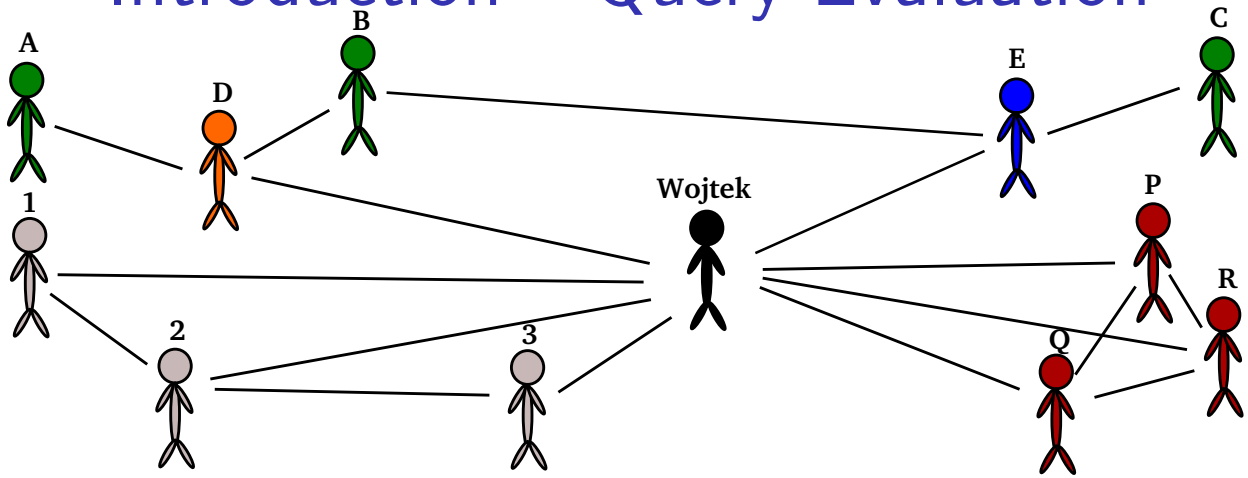
Introduction – databases

Databases: storage of data and retrieval of information.

1. A store has its list of offered products.
2. Private collection of photos.
3. Map of a metro system.
4. Social network and its graph. Which pairs of people are in the 2-handshakes distance from each other?
5. ...



Introduction – Query Evaluation



Query Evaluation Problem:

Input:

- query $q(\bar{x})$
- database $\mathbf{D}$ of size $\|\mathbf{D}\|$

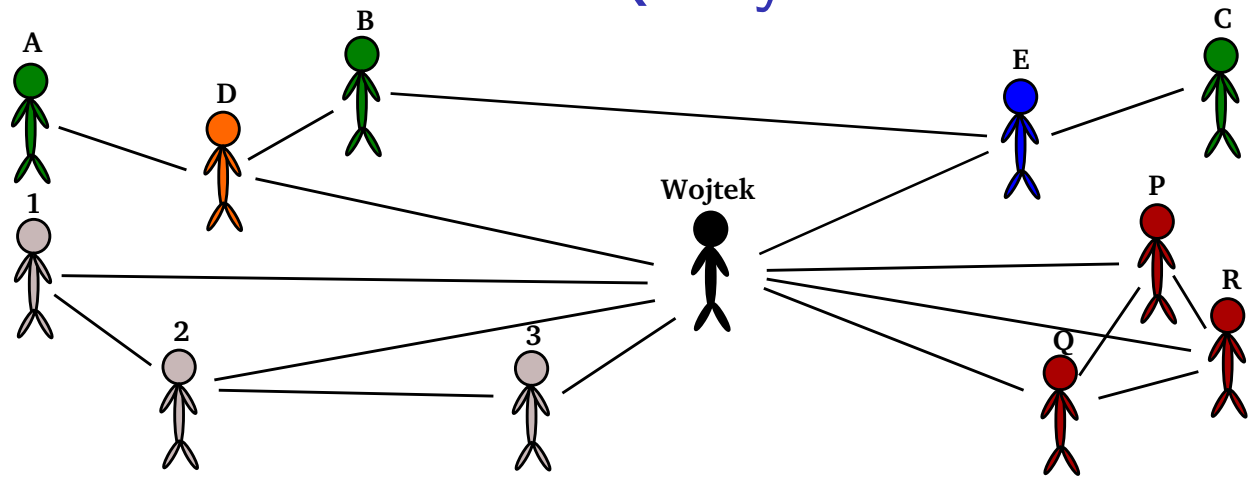
Output:

$q(\mathbf{D})$

Which pairs of people are in the 2-handshakes distance from each other?

$(1,2),(1,3),(1,\text{Wojtek}),(1,D),(1,E),(1,P),(1,Q),(1,R)$
$(2,1),(2,3),(2,\text{Wojtek}),(2,D),(2,E),(2,P),(2,Q),(2,R)$
$(3,1),(3,2),(3,\text{Wojtek}),(3,D),(3,E),(3,P),(3,Q),(3,R)\ldots$

Introduction – Query Evaluation



Query Evaluation Problem:

Input:

- query $q(\bar{x})$
- database $\mathbf{D}$ of size $\|\mathbf{D}\|$

Output:

$q(\mathbf{D})$

Special case: q boolean = Model Checking Problem.

Are there two green friends? *No*

Introduction – Query Evaluation

Query Evaluation Problem:

Input:
- query $q(\bar{x})$
- database $\mathbf{D}$

Output:

$q(\mathbf{D})$

Issue: $|q(\mathbf{D})| = O(\|\mathbf{D}\|^k)$ if q has k free variables.

$\|\mathbf{D}\|^k$ is too big!

Query Enumeration and Related Problems

Input:

- query $q(\bar{x})$
- database $\mathbf{D}$

Enumeration:

- compute first solution quickly,
- compute the rest with minimal *delay* between consecutive ones.

Aim: First solution in $O(\|\mathbf{D}\|)$, $O(1)$ delay $\rightarrow$ $\textsc{Constant-Delay}_{lin}$

Query Enumeration and Related Problems

Input:
- query $q(\bar{x})$
- database $\mathbf{D}$

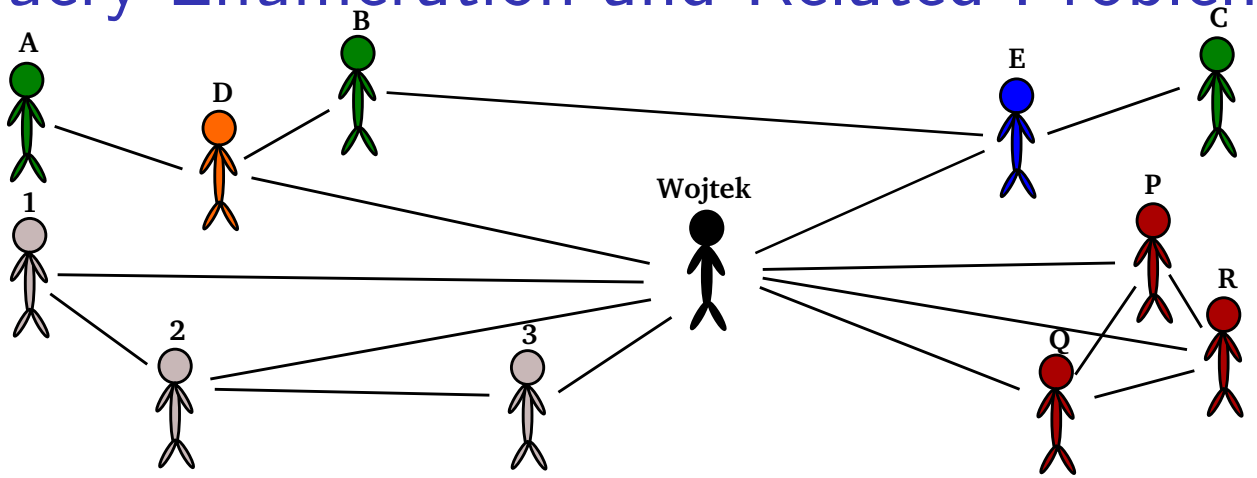
Enumeration:
- compute first solution quickly,
- compute the rest with minimal *delay* between consecutive ones.

Aim: First solution in $O(\|\mathbf{D}\|)$, $O(1)$ delay $\rightarrow$ $\textsc{Constant-Delay}_{lin}$

In practice:

- the $O(\|\mathbf{D}\|)$ *preprocessing* is a linear refactorization of the input database (usually adding to it some additional navigational power),
- the refactorized database can then be traversed efficiently, producing new solutions after only constant delays.

Query Enumeration and Related Problems



Query ~~Evaluation~~ Problem:

Input:

- query $q(\bar{x})$
- database $\mathbf{D}$

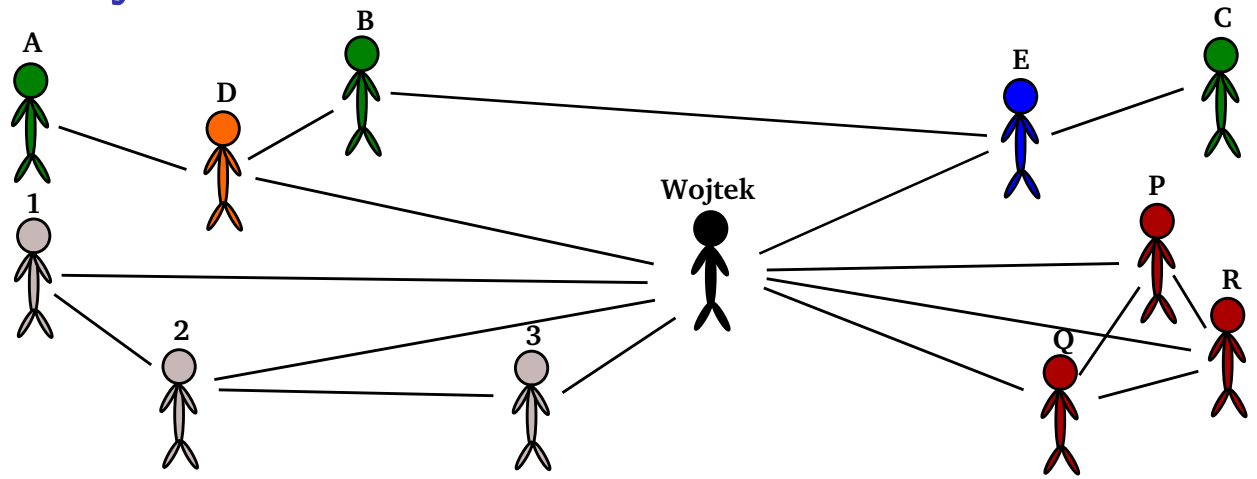
Output:

$q(\mathbf{D})$

Which pairs of people are in the 2-handshakes distance from each other?

$(1,2),(1,3),(1,\text{Wojtek}),(1,D),(1,E),(1,P),(1,Q),(1,R)$
$(2,1),(2,3),(2,\text{Wojtek}),(2,D),(2,E),(2,P),(2,Q),(2,R)$
$(3,1),(3,2),(3,\text{Wojtek}),(3,D),(3,E),(3,P),(3,Q),(3,R)\ldots$

Query Enumeration and Related Problems



Query Enumeration Problem:

Input:

- query $q(\bar{x})$
- database $\mathbf{D}$

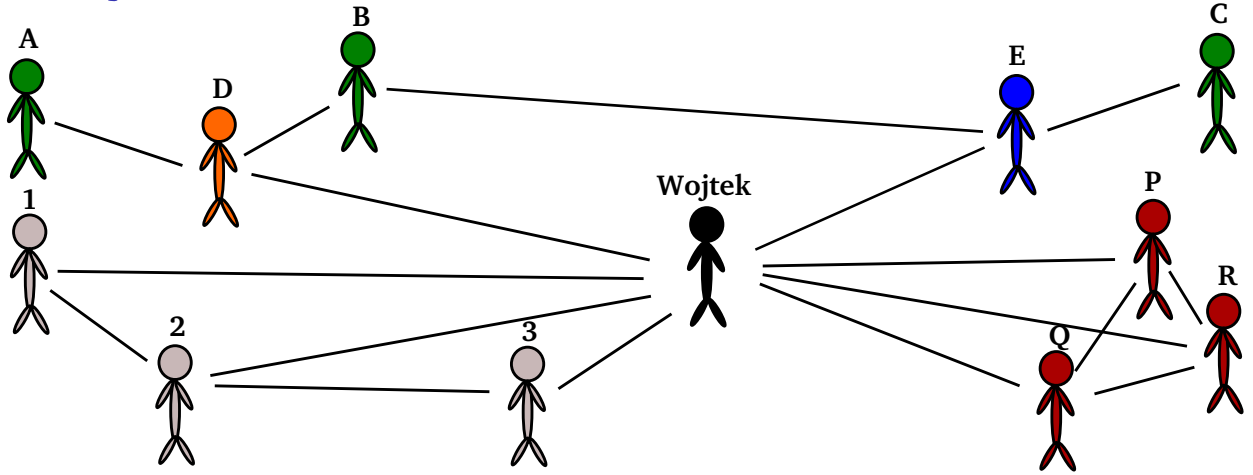
Output:

$q(\mathbf{D})$

Which pairs of people are in the 2-handshakes distance from each other?

$(1, 2)$

Query Enumeration and Related Problems



Query Enumeration Problem:

Input:

- query $q(\bar{x})$
- database $\mathbf{D}$

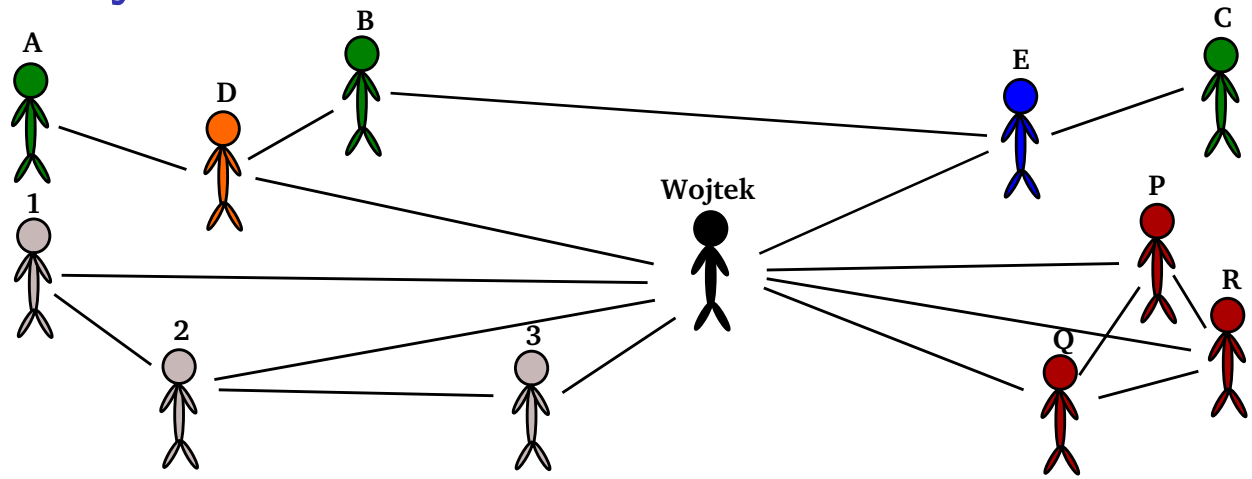
Output:

$q(\mathbf{D})$

Which pairs of people are in the 2-handshakes distance from each other?

$(1, 2), (1, 3)$

Query Enumeration and Related Problems



Query Enumeration Problem:

Input:

- query $q(\bar{x})$
- database $\mathbf{D}$

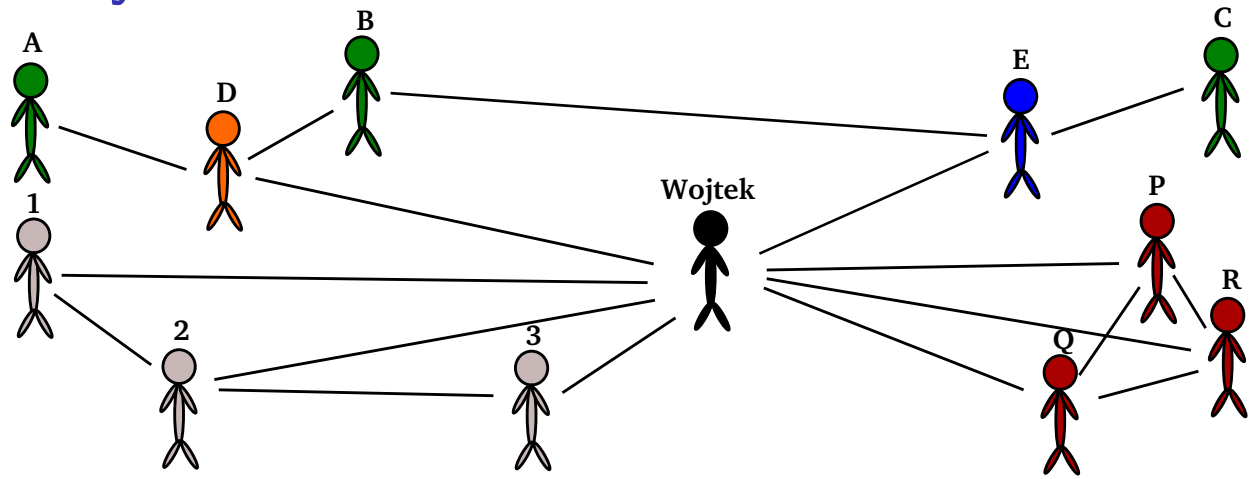
Output:

$q(\mathbf{D})$

Which pairs of people are in the 2-handshakes distance from each other?

$(1, 2), (1, 3), (1, \text{Wojtek})$

Query Enumeration and Related Problems



Query Enumeration Problem:

Input:

- query $q(\bar{x})$
- database $\mathbf{D}$

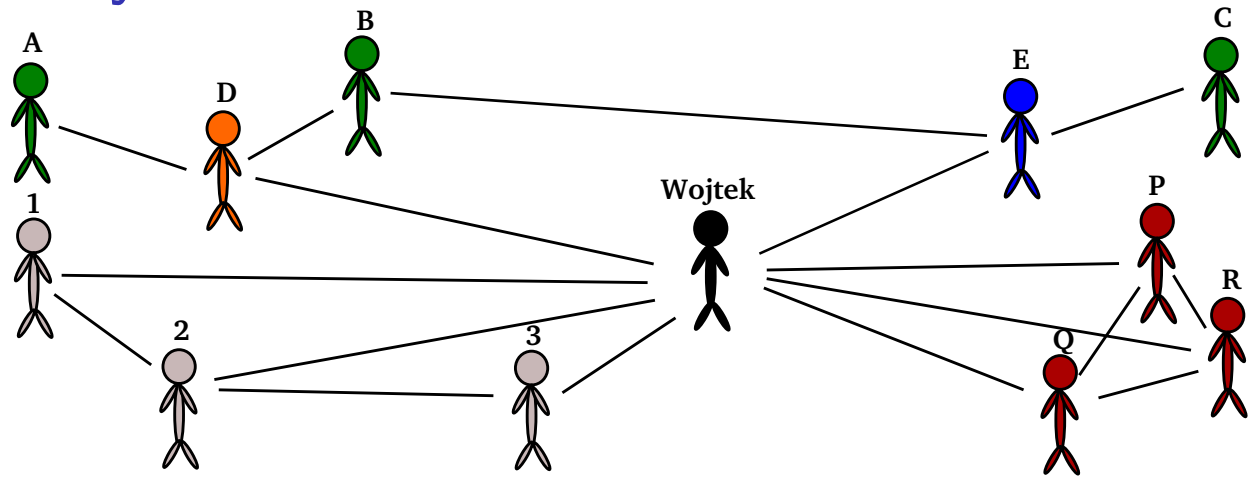
Output:

$q(\mathbf{D})$

Which pairs of people are in the 2-handshakes distance from each other?

$(1, 2), (1, 3), (1, \text{Wojtek}), (1, D)$

Query Enumeration and Related Problems



Query Enumeration Problem:

Input:

- query $q(\bar{x})$
- database $\mathbf{D}$

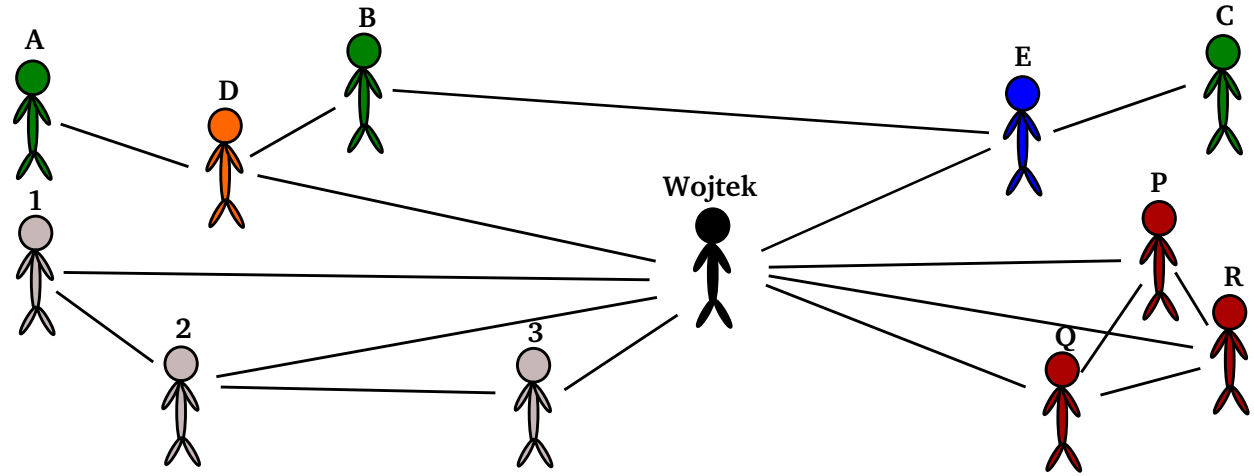
Output:

$q(\mathbf{D})$

Which pairs of people are in the 2-handshakes distance from each other?

$(1,2),(1,3),(1,\text{Wojtek}),(1,D),(1,E),(1,P),(1,Q),(1,R)$
$(2,1),(2,3),(2,\text{Wojtek}),(2,D),(2,E),(2,P),(2,Q),(2,R)$
$(3,1),(3,2),(3,\text{Wojtek}),(3,D),(3,E),(3,P),(3,Q),(3,R)\ldots$

Query Enumeration and Related Problems



Counting Problem:

Input:

- query $q(\bar{x})$
- database $\mathbf{D}$

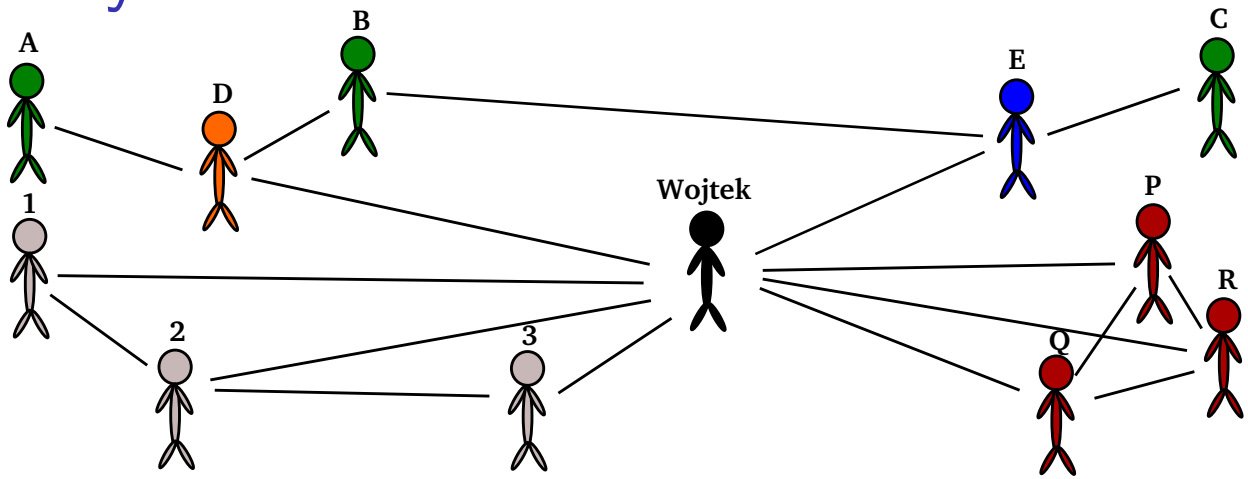
Output:

$|q(\mathbf{D})|$

Aim: $O(\|\mathbf{D}\|)$

How many pairs of people are in the 2-handshakes distance from each other? 78

Query Enumeration and Related Problems



Testing Problem:

Input:

- query $q(\bar{x})$
- database $\mathbf{D}$

Dynamical output:

given $\bar{v}$, answer $\bar{v} \stackrel{?}{\in} q(\mathbf{D})$

Aim:

preprocessing (once, $\bar{v}$ unknown) $O(\|\mathbf{D}\|)$

answering (multiple times) $O(1)$

After preprocessing

Is $(1, P)$ in the 2-handshakes distance? *Yes*

Is (A, E) in the 2-handshakes distance? *No*

Constant-Delay$_{lin}$ vs. Evaluation

Remark 1

Constant-Delay$_{lin}$ *enumeration* $\rightarrow$ $O(\|\mathbf{D}\| + |q(\mathbf{D})|)$ *evaluation.*

Remark 2

Constant-Delay$_{lin}$ *enumeration* $\rightarrow$ $O(\|\mathbf{D}\|)$ *model checking.*

Computational model – RAM machine

- Necessary, since we want to talk about linear time.
- We assume that the elements can be compared in constant time.

$$A <_{\text{lex}} P <_{\text{lex}} \text{Wojtek}$$

In real life: user = (short) e-mail address

- We can sort lexicographically tuples of constant size in linear time.

Radix sort

- We can follow pointers in constant time.

Direct access to the n-th cell of an array.

- Coding of a graph:

List of consecutive edges.

NOT an adjacency matrix!

Introduction

Enumeration

Examples

Results

Conclusions

Example 1: enumeration of edges

Database:
graph $\mathbf{G} = (V, E)$
$\|\mathbf{G}\| = |V| + |E|$

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

- CONSTANT-DELAY$_{lin}$ enumeration is not too difficult.
- $O(\|\mathbf{G}\|)$ counting is not too difficult.
- Testing requires logarithmic time.
  - $O(1)$ testing if $\mathbf{G}$ is a tree.

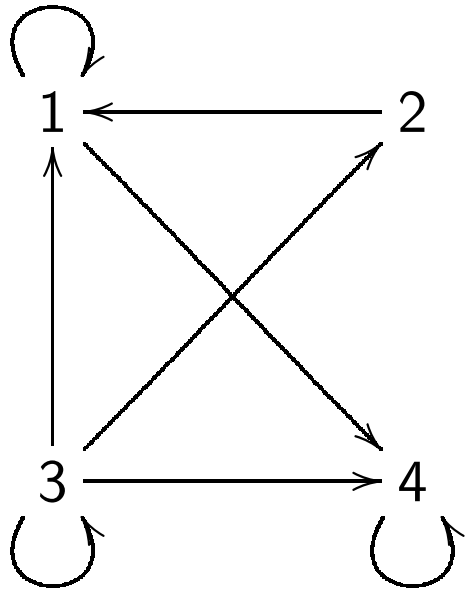
Example 2: complement of a graph

Database:
graph $\mathbf{G} = (V, E)$
$\|\mathbf{G}\| = |V| + |E|$

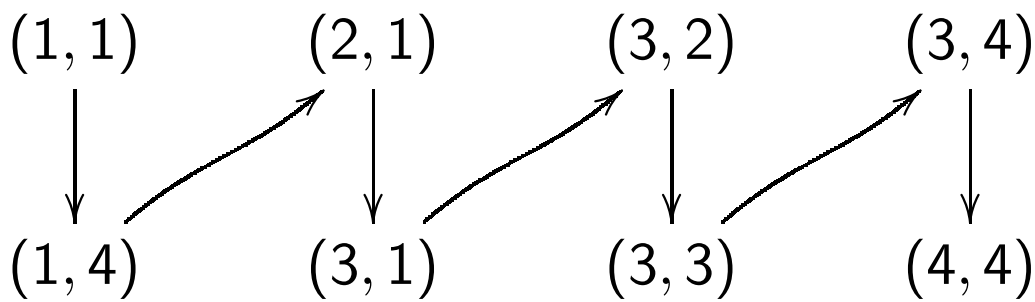
Query:
$q(x, y) = \neg E(x, y)$

- Constant-Delay$_{lin}$ enumeration already not trivial.
- $O(\|\mathbf{G}\|)$ counting is not too difficult.
- Testing still requires logarithmic time.
  - $O(1)$ testing if $\mathbf{G}$ is a tree.

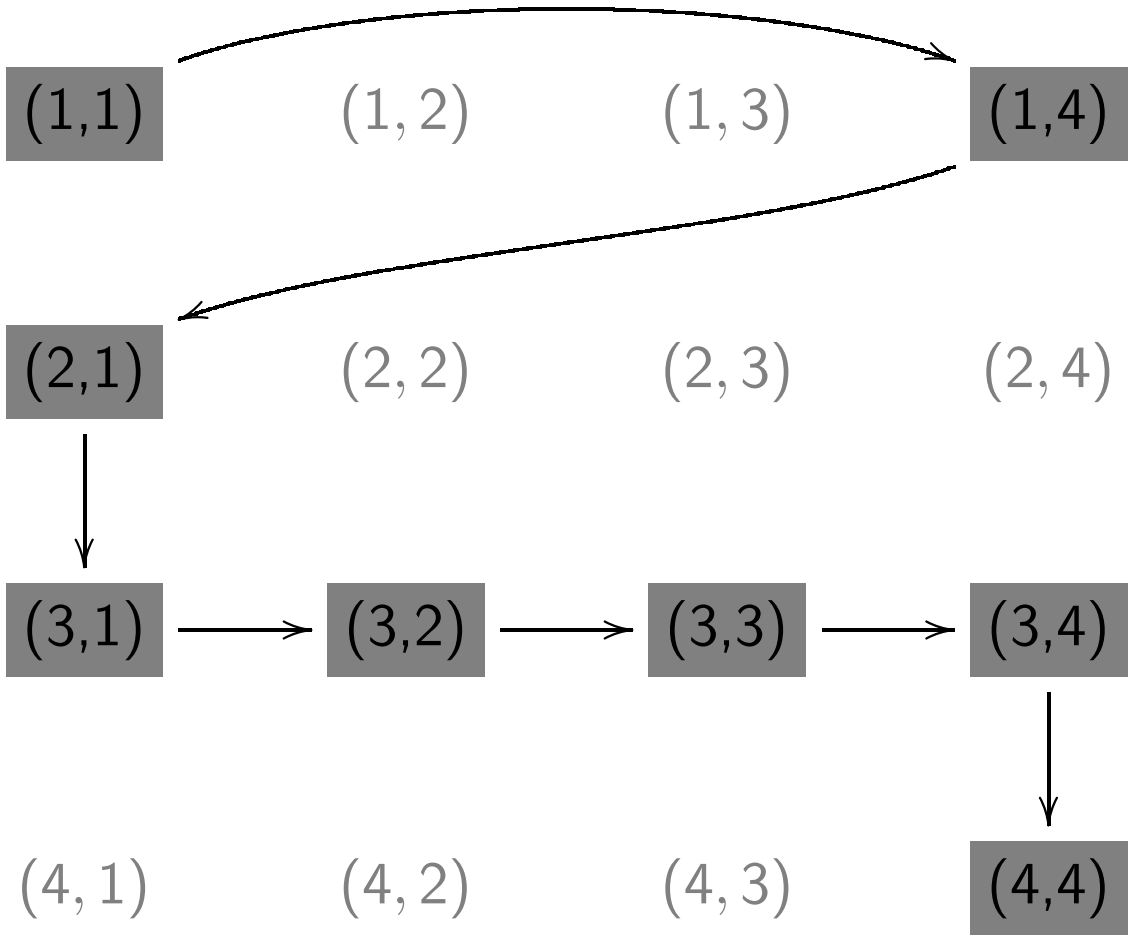
Example 2: complement of a graph



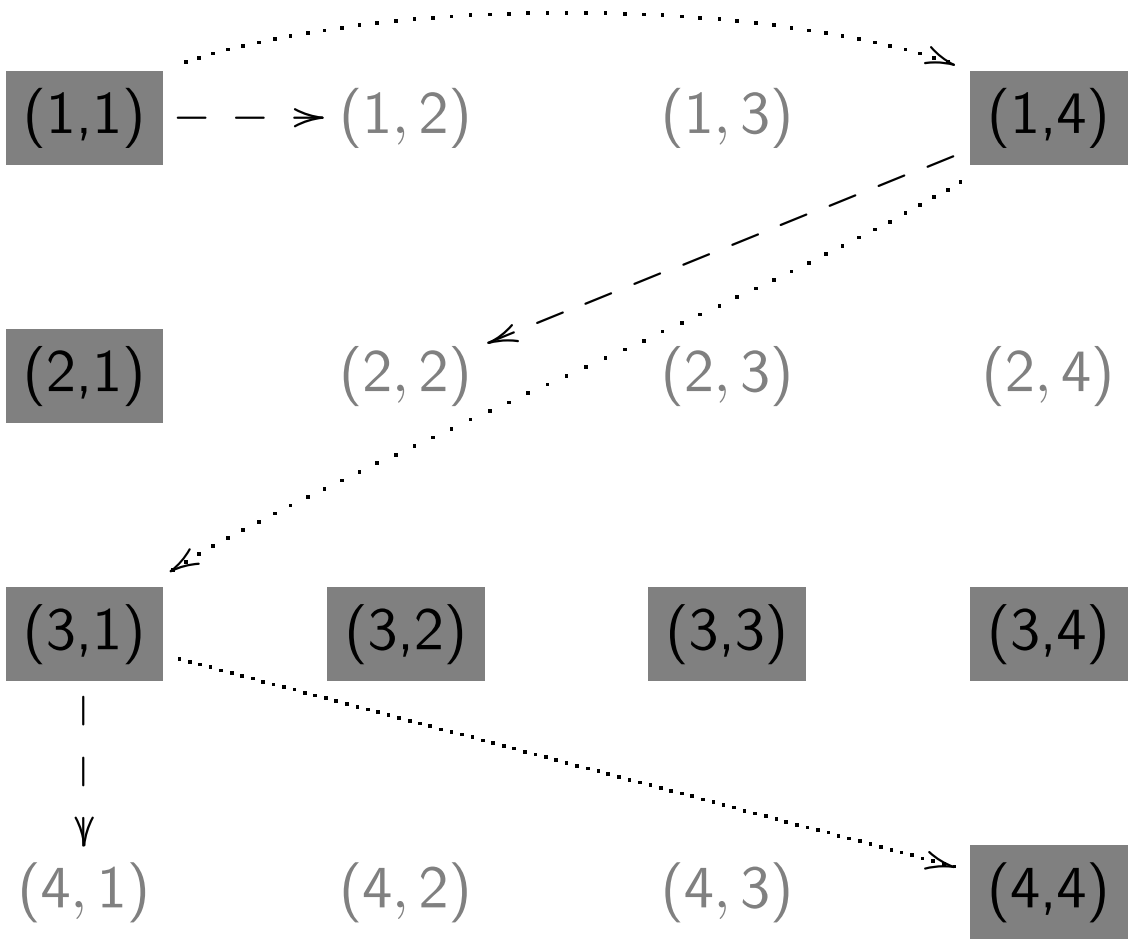
Is given by the following list:



Example 2: complement of a graph



Example 2: complement of a graph



Example 3: 2-handshake distance

Database:
graph $\mathbf{G} = (V, E)$
$\|\mathbf{G}\| = |V| + |E|$

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

- $\textsc{Constant-Delay}_{lin}$ enumeration not possible? (Bagan'09)
- $\textsc{Constant-Delay}_{lin}$ enumeration if $\mathbf{G}$ has bounded degree.

- $O(\|\mathbf{G}\|)$ counting not possible?
- $O(\|\mathbf{G}\|)$ counting if $\mathbf{G}$ has bounded degree.

- $O(1)$ testing not possible? (Bagan'09)
- $O(1)$ testing if $\mathbf{G}$ has bounded degree.

Introduction

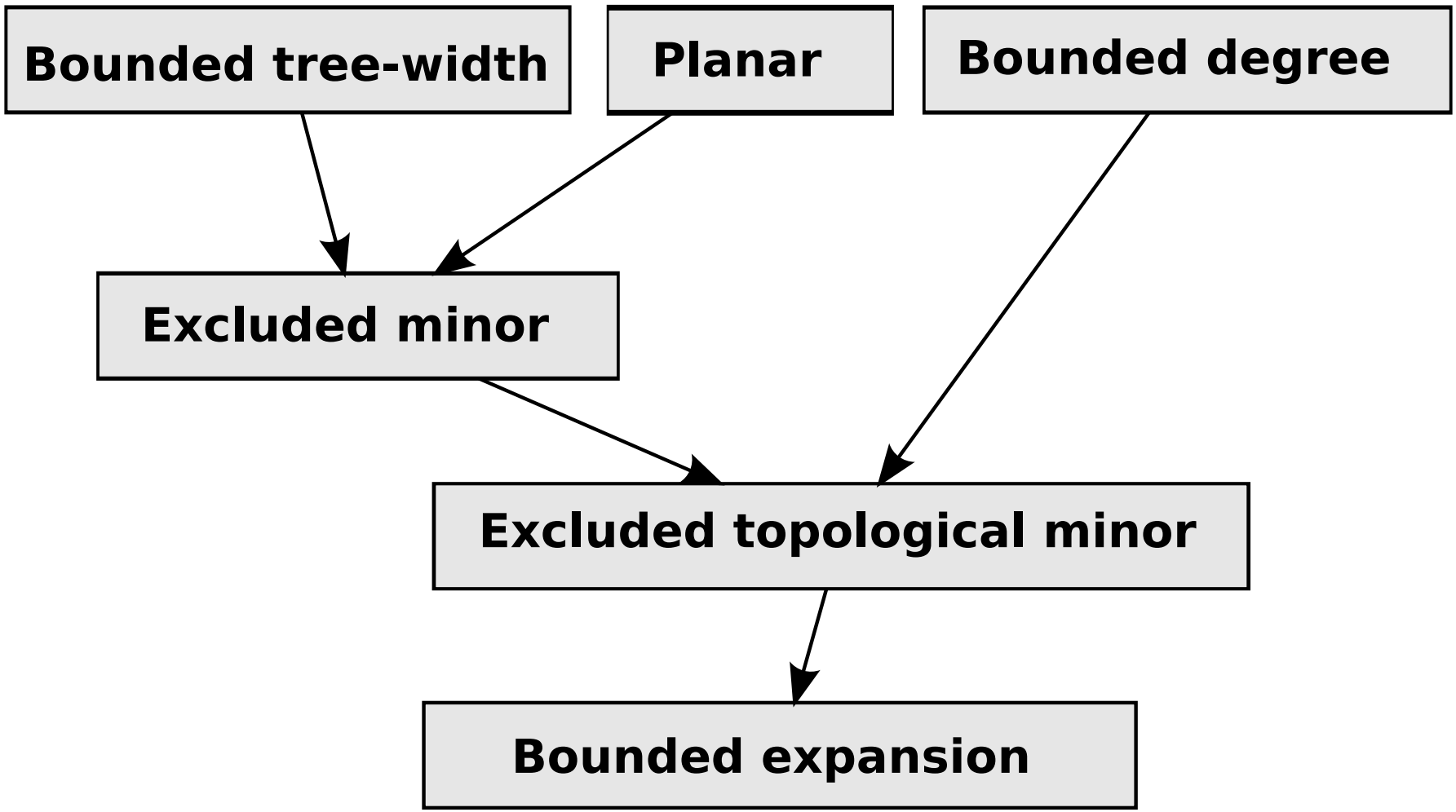
Enumeration

Examples

Results

Conclusions

Classes of graphs



Results - FO over structures with bounded degree

(K, Segoufin'11) gives a new proof of:

Theorem 1 (Durand, Grandjean'07)

$\mathcal{C}$ with bounded degree, $q(\bar{x}) \in$ FO, given $\mathbf{G} \in \mathcal{C}$,
the enumeration of q over $\mathbf{G}$ is in $\textsc{Constant-Delay}_{lin}$.

Moreover, the output is returned in the lexicographical order.

The hidden constants are triply exponential in $|q|$ (K, Segoufin'11).

Results - FO over structures with bounded degree

(K, Segoufin'11) gives new proofs of:

Theorem 2 (Bagan, Durand, Grandjean, Olive'08)

$\mathcal{C}$ with bounded degree, $q(\bar{x}) \in \mathrm{FO}$, given $\mathbf{G} \in \mathcal{C}$,
the counting $|q(\mathbf{D})|$ is in $O(\|\mathbf{G}\|)$.

Theorem 3 (Lindell'08)

$\mathcal{C}$ with bounded degree, $q(\bar{x}) \in \mathrm{FO}$, given $\mathbf{G} \in \mathcal{C}$,
the testing of q over $\mathbf{G}$ is in $O(1)$ after $O(\|\mathbf{G}\|)$ preprocessing.

The hidden constants are triply exponential in $|q|$ (K, Segoufin'11).

Comments

- The original proof:
  - Quantifier elimination procedure based on bijective representation.
  - Non-elementary dependency on $|q|$ ($\underbrace{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}_{|q|}$).
- Our proof:
  - Based on Gaifman locality of FO.
  - Gives the $2^{2^{2^{O(|q|)}}}$ dependency on $|q|$.

Results - FO over structures with bounded expansion

Theorem 4 (K, Segoufin'13)

$\mathcal{C}$ with bounded expansion, $q(\bar{x}) \in \mathsf{FO}$, given $\mathbf{G} \in \mathcal{C}$, the enumeration of q over $\mathbf{G}$ is in $\textsc{Constant-Delay}_{lin}$.

Moreover, the output is returned in the lexicographical order.

Results - FO over structures with bounded expansion

Theorem 5 (K, Segoufin'13)

$\mathcal{C}$ with bounded expansion, $q(\bar{x}) \in \mathsf{FO}$, given $\mathbf{G} \in \mathcal{C}$,
the counting $|q(\mathbf{D})|$ is in $O(\|\mathbf{G}\|)$.

Theorem 6 (K, Segoufin'13)

$\mathcal{C}$ with bounded expansion, $q(\bar{x}) \in \mathsf{FO}$, given $\mathbf{G} \in \mathcal{C}$,
the testing of q over $\mathbf{G}$ is in $O(1)$ after $O(\|\mathbf{G}\|)$ preprocessing.

Comments

- The hidden constants are non-elementary ($\underbrace{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}_{|q|}$).

- This is unavoidable already for model checking over unranked trees, unless FPT $=$ AW[$*$] (Frick, Grohe'04).

Proof Strategy

- Quantifier elimination procedure:

  - For all $q(\bar{x}y)$ quantifier free,
  - Exists $q'(\bar{x})$ quantifier free s.t.
  - given $\mathbf{G} \in \mathcal{C}$, in $O(\|\mathbf{G}\|)$ we construct $\mathbf{G}'$ s.t.
  - $(\exists y q)(\mathbf{G}) = q'(\mathbf{G}')$.

  Graph $\mathbf{G}'$ is an "augmentation" of $\mathbf{G}$, which allows us to continue the inductive process.

- We then solve the quantifier free case.

Both steps are not trivial.

Results - MSO over structures with bounded treewidth

(K, Segoufin'12) gives a new proof of:

Theorem 7 (Bagan'06)

$\mathcal{C}$ with bounded treewidth, $q(\bar{x}) \in$ MSO, given $\mathbf{G} \in \mathcal{C}$, the enumeration of q over $\mathbf{G}$ is in $\textsc{Constant-Delay}_{lin}$.

Results - MSO over structures with bounded treewidth

(K, Segoufin'12) gives new proofs of:

Theorem 8 (Arnborg, Lagergren, Seese'91)

$\mathcal{C}$ with bounded treewidth, $q(\bar{x}) \in$ MSO, given $\mathbf{G} \in \mathcal{C}$, the counting $|q(\mathbf{D})|$ is in $O(\|\mathbf{G}\|)$.

Theorem 9

$\mathcal{C}$ with bounded treewidth, $q(\bar{x}) \in$ MSO, given $\mathbf{G} \in \mathcal{C}$, the testing of q over $\mathbf{G}$ is in $O(1)$ after $O(\|\mathbf{G}\|)$ preprocessing.

Comments

- The original proof:
  - Allows for monadic second-order free variables.
  - No bound on total memory usage during the enumeration phase.
  - Rather complicated reasoning concerning tree automatons.

- Our proof:
  - Only first-order free variables.
  - Constant total memory usage during the enumeration phase.
  - Sequence of reduction steps.

- In both cases the hidden constants are non-elementary ($\underbrace{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}_{|q|}$).

MSO enumeration – outline

Theorem 6 (Bagan'06)

$\mathcal{C}$ with bounded treewidth, $q(\bar{x}) \in$ MSO, given $\mathbf{G} \in \mathcal{C}$, the enumeration of q over $\mathbf{G}$ is in $\textsc{Constant-Delay}_{lin}$.

The proof of (K, Segoufin'12) is a sequence of consecutive reduction steps:

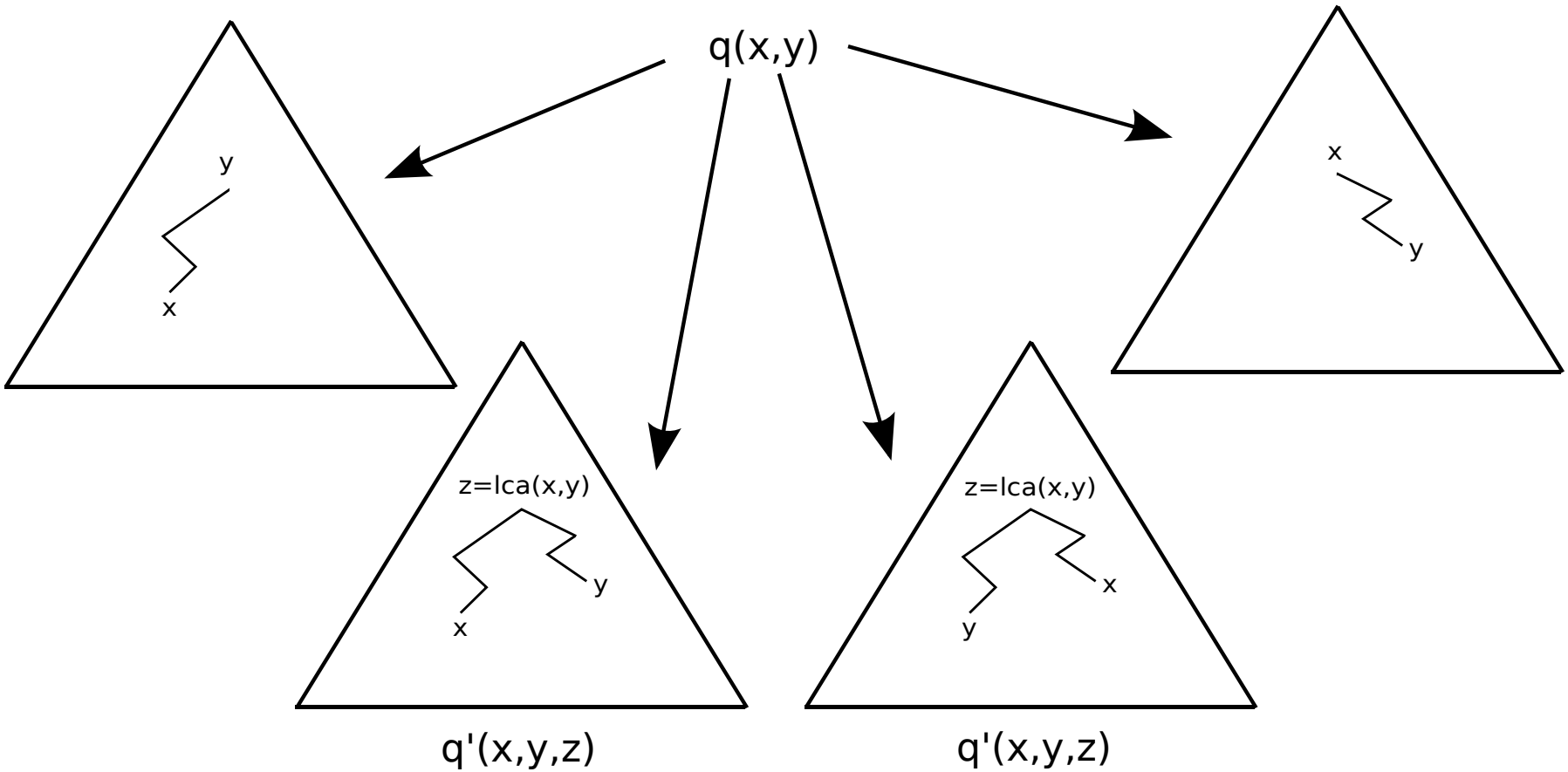
MSO enumeration – outline

Reduction steps:

1. Trees instead of structures of bounded tree-width.
   Compute the tree decomposition in linear time. (Bodlaender)
   Interpret the tree decomposition in MSO. (Courcelle)
2. Binary trees instead of unranked trees.
   First child – next sibling encoding. (Rabin)
3. Ancestor-typed outputs including all the least common ancestors.
4. Binary queries.
   Composition Lemma.
5. Binary queries from $\Sigma_2(<)$.
   Colcombet.
6. Final induction.

MSO enumeration – outline (3/6)

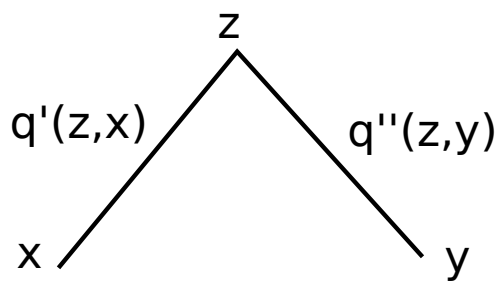
Step 3: Ancestor-typed outputs including all the least common ancestors.



MSO enumeration – outline (4/6)

Step 4: Binary queries.

$q(x, y, z) = \bigvee_{q', q''} q'(x, y) \wedge q''(x, z)$.



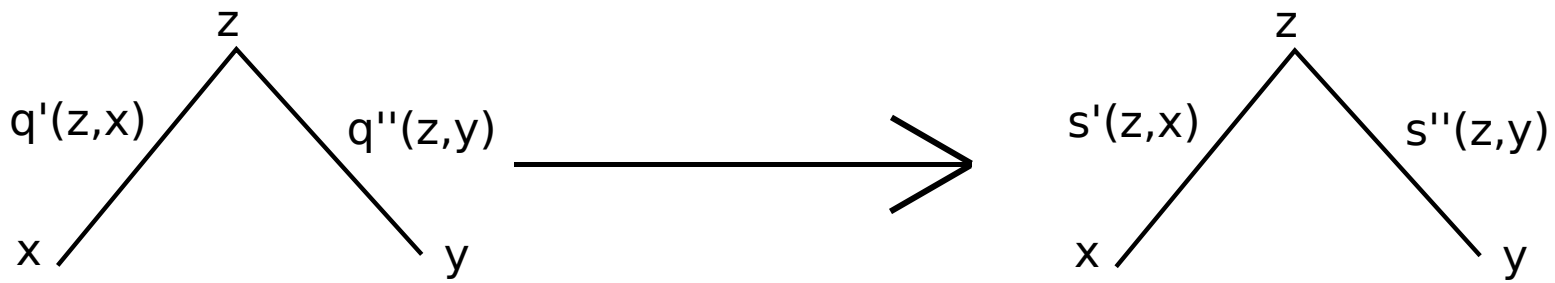
Disjunction is exclusive.

Composition Lemma for MSO over trees (can be proved using a simple Ehrenfeucht-Fraïssé game argument).

MSO enumeration – outline (5/6)

Step 5: Binary queries from $\Sigma_2(<)$.

$s'(z,x)$ and $s''(z,y)$ are of the form $\exists \bar{v} \forall \bar{u}\ \theta(x,y,z,\bar{v},\bar{u})$, where θ is quantifier free.



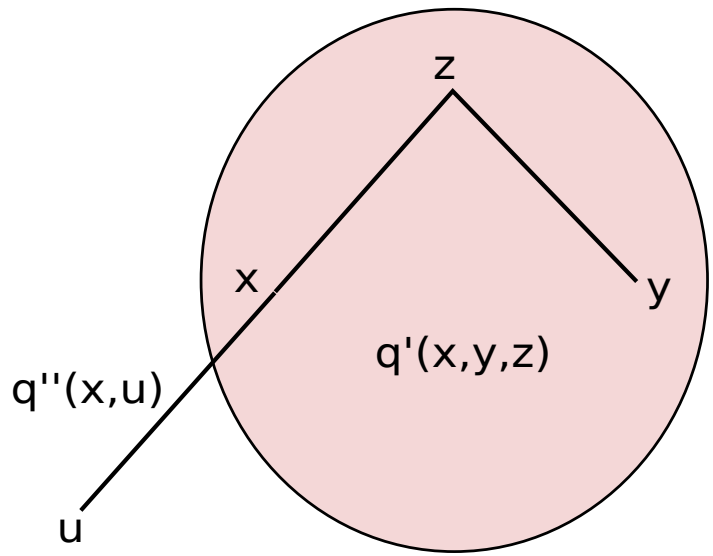
Theorem 10 (Colcombet)

Over binary trees, every MSO formula $q(x,y)$ is equivalent to a $\Sigma_2^+(<)$ formula $q'(x,y)$. $q' = \exists \bar{v} \forall \bar{u}\ \theta(x,y,z,\bar{v},\bar{u})$, where θ is a disjunction of conjunctions of atomic predicates or MSO *queries with one free variable or atoms using $<$.*

MSO enumeration – outline (6/6)

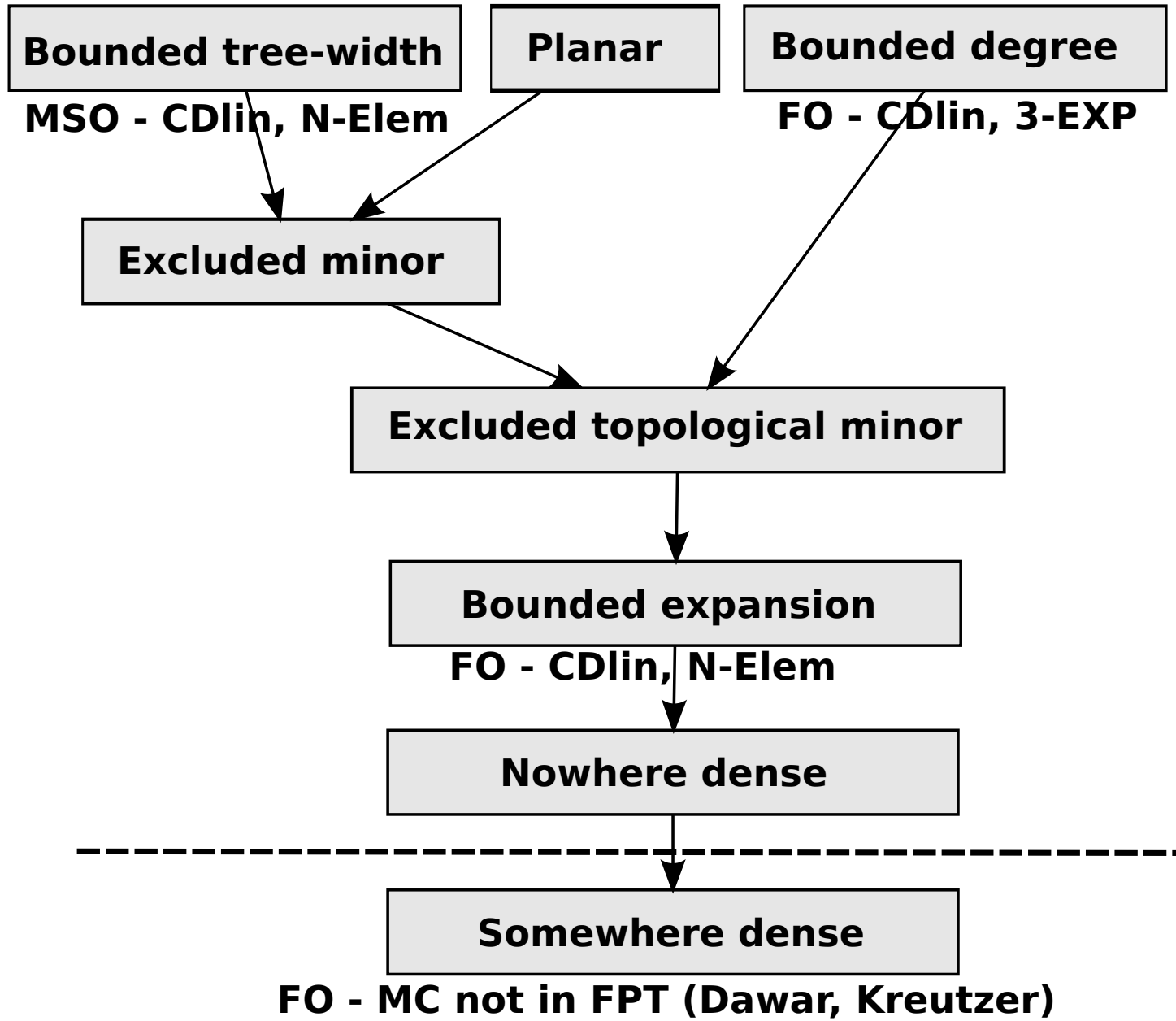
The rest is an induction on the number of free variables:

$$q(x, y, z, u) = q'(x, y, z) \wedge q''(x, u)$$



- We inductively enumerate $q'(x, y, z)$.
- For every solution (a, b, c) to q' we (efficiently) extend it with all solutions to q'' of the form $(a, \_)$.

Summary


Bounded tree-width
Planar
Bounded degree
MSO - CDlin, N-Elem
FO - CDlin, 3-EXP
Excluded minor
Excluded topological minor
Bounded expansion
FO - CDlin, N-Elem
Nowhere dense
Somewhere dense
FO - MC not in FPT (Dawar, Kreutzer)

Perspectives

Push further along the lines of the current approach:

- Consider FO queries over classes of nowhere dense structures.
- Consider other query languages over structures with particular properties.

Perspectives

$\textsc{Constant-Delay}_{lin}$ implies evaluation algorithm working in time $O(\|\mathbf{G}\| + |q(\mathbf{G})|)$.

- When is the converse true?

When is $\textsc{Constant-Delay}_{lin}$ enumeration impossible?

- Most of the lower bounds require complexity assumptions (matrix multiplication, FPT $\neq$ AW[$*$], etc.).
- In the mentioned cases also $O(\|\mathbf{G}\| + |q(\mathbf{G})|)$ evaluation is not possible.
- How much can we prove directly?

Perspectives

Let $A, B \in \text{Constant-Delay}_{lin}$ (black boxes).

Under what assumptions

- $A \cup B \in \text{Constant-Delay}_{lin}$, (The best understood.)
- $A \cap B \in \text{Constant-Delay}_{lin}$,
- $\bar{A} \in \text{Constant-Delay}_{lin}$?

Thank You!