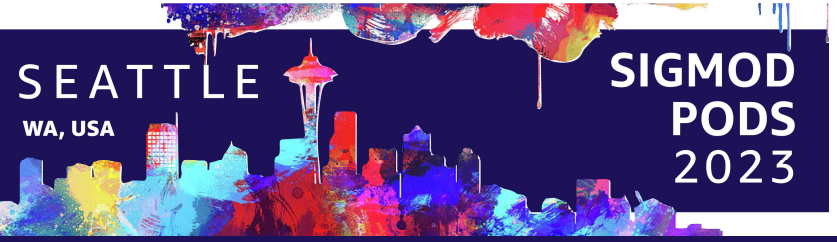


CQs With Self-Joins

Towards a Fine-Grained Enumeration Complexity Analysis

Nofar Carmeli, Luc Segoufin

Example: Conjunctive Query

SeatTogether

Student 1	Student 2
Emma	Andrew
Madison	Ethan

Grade

Student	Grade
Emma	85
Andrew	85
Madison	61
Ethan	97

Q_1 answers

Student 1	Student 2
Emma	Andrew

not a full CQ
CQ of arity 2
free variables: S_1, S_2

self-join
(same relation in different atoms)

$$Q_1(S_1, S_2) \leftarrow \text{SeatTogether}(S_1, S_2), \text{Grade}(S_1, G), \text{Grade}(S_2, G)$$

$$Q_2(S, E, P) \leftarrow \text{Registration}(S, E), \text{Staff}(E, P), \text{COI}(S, P)$$

full CQ
(no projections)

self-join free CQ
(no self-joins)

Tractability Measure

- Possibly: output ≫ input
  - No linear-time algorithm
- Minimal requirements:
  - Linear time (to read input)
  - Constant time per answer (to print output)
- Enum<lin,const>:
  queries that can be answered
  with linear preprocessing and constant delay
  in the RAM model in data complexity

  (input = database; query size = constant)

Enumeration Dichotomy

[Bagan,Durand,Grandjean; CSL 2007]
[Brault-Baron; 2013]

Given a self-join-free conjunctive query Q,

$Q \in$ Enum<lin,const>

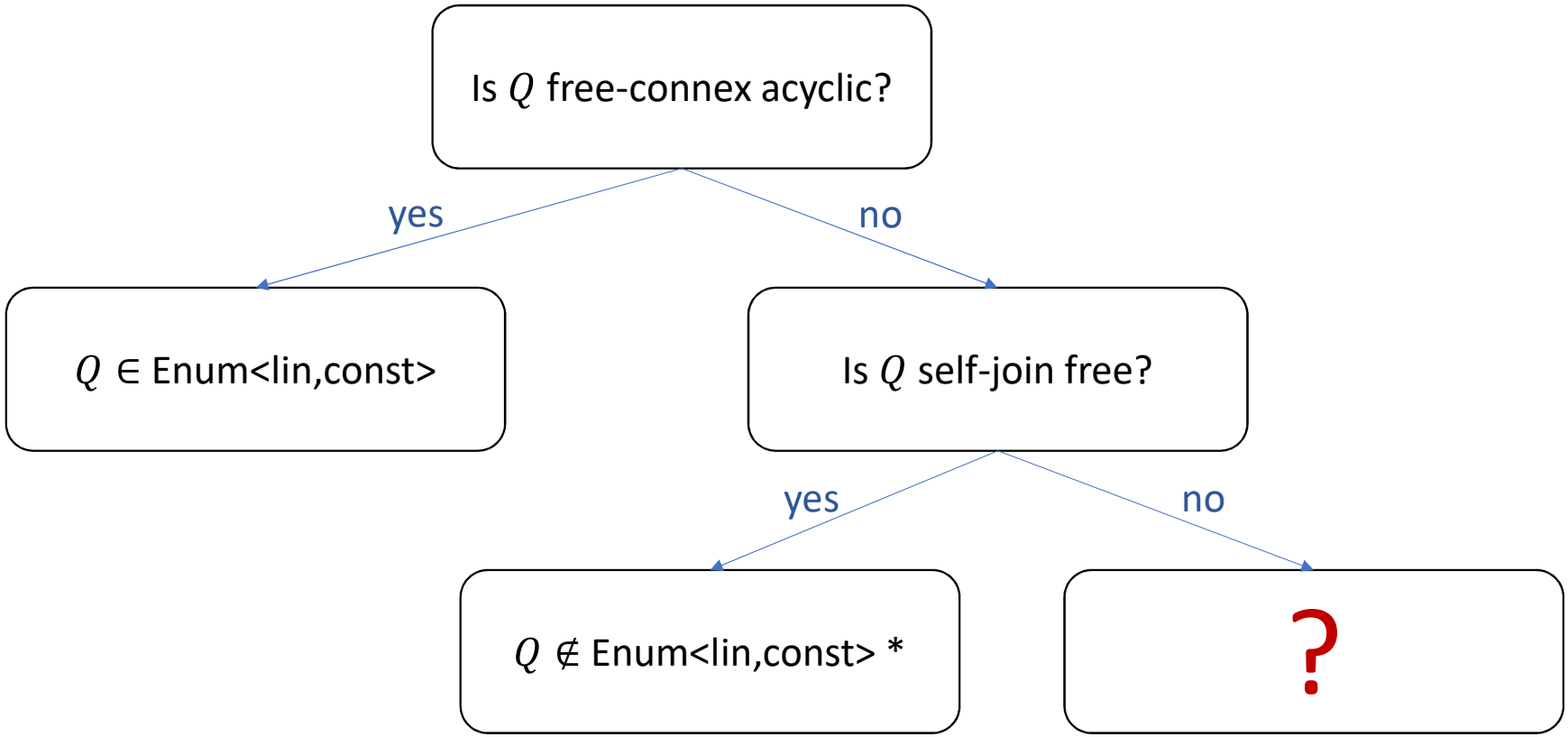
$\Updownarrow$*

Q is free-connex acyclic

* assuming the sHyperclique and sBMM hypotheses:
Boolean matrices cannot be multiplied in linear time in the number of their zero entries
$\forall k \geq 3$ it is not possible to determine the existence of a k-hyperclique
in a $(k-1)$-uniform hypergraph with m edges in time $O(m)$

Classifying a CQ

[Bagan,Durand,Grandjean; CSL 2007]
[Brault-Baron; 2013]



* assuming the sHyperclique and sBMM hypotheses:
Boolean matrices cannot be multiplied in linear time in the number of their zero entries
$\forall k \geq 3$ it is not possible to determine the existence of a k-hyperclique
in a $(k-1)$-uniform hypergraph with m edges in time $O(m)$

Definition: Free-Connex Acyclic

An acyclic CQ has a graph with:

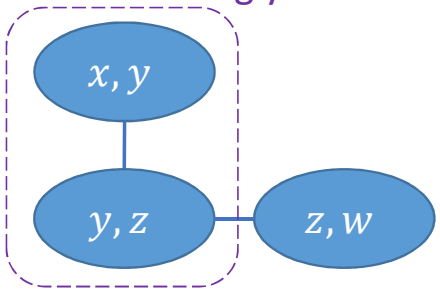
A free-connex CQ also requires:

1. a node for every atom

2. tree

3. for every variable:
the nodes containing it form a subtree

nodes containing y

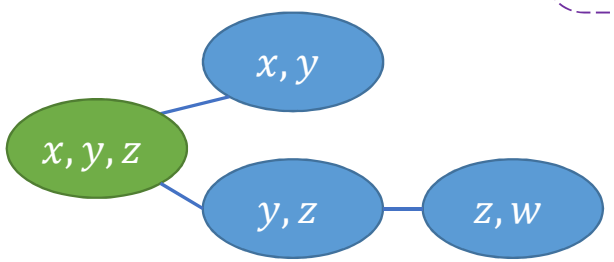


free − connex

acyclic

$$Q(x,y,z) \leftarrow R_1(x,y), R_2(y,z), R_3(z,w)$$

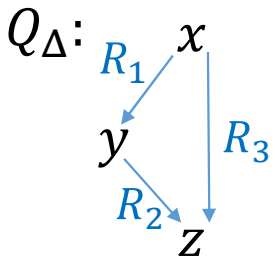
4. remains acyclic when introducing an atom with the free variables

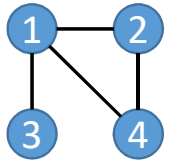


Lower Bound: Cyclic Joins

[Brault-Baron 13]

Assumption: cannot detect triangles in a graph in linear time





edges (a,b) with a<b

Q

x	y	z
1	2	4

$R_1 = R_2 = R_3$

1	2
1	3
1	4
2	4

Cyclic: $Q_\Delta(x, y, z) \leftarrow R_1(x, y), R_2(y, z), R_3(x, z)$

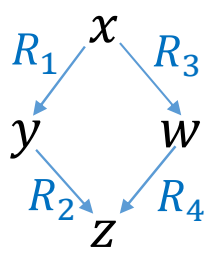
first answer in linear time $\Longrightarrow$ triangle in linear time $\Longrightarrow$ not possible

Lower Bound: Cyclic Joins

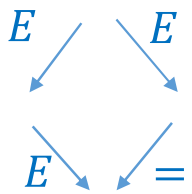
[Brault-Baron 13]

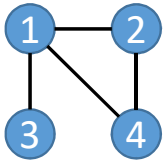
Assumption: cannot detect triangles in a graph in linear time

Q_1:



Construction:





edges (a,b) with a<b

with self-joins,
cannot assign a different relation
to different atoms

Q

x	y	z	w
1	2	4	4

$R_1 = R_2 = R_3$

1	2
1	3
1	4
2	4

R_4

1	1
2	2
3	3
4	4

Cyclic: $Q_1(x, y, z, w) \leftarrow R_1(x, y), R_2(y, z), R_3(x, w), \cancel{R_4}^{R_3}(w, z)$

first answer in linear time $\Longrightarrow$ triangle in linear time $\Longrightarrow$ not possible

Self-Joins

- Lower bounds do not apply with self-joins
- Can they be easier?
  - Yes! [Berkholz, Gerhardt, Schweikardt; SIGLOG News 20]

- A simple example:

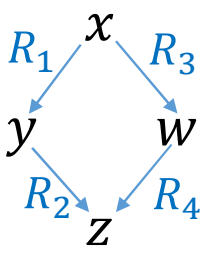
$$Q_1(x,y,z,w) \leftarrow R_1(x,y), R_2(y,z), R_3(x,w)\ R_4(w,z)$$

$\notin$ Enum<lin,const>

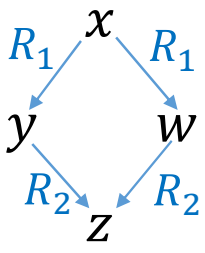
$$Q_2(x,y,z,w) \leftarrow R_1(x,y), R_2(y,z), R_1(x,w)\ R_2(z,w)$$

$\in$ Enum<lin,const>

Q_1:



Q_2:



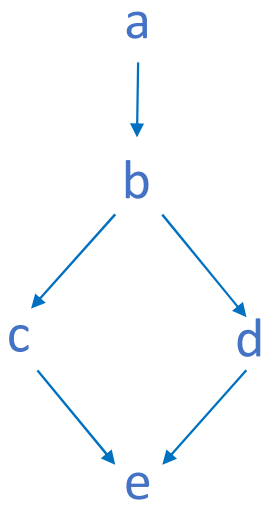
Algorithm

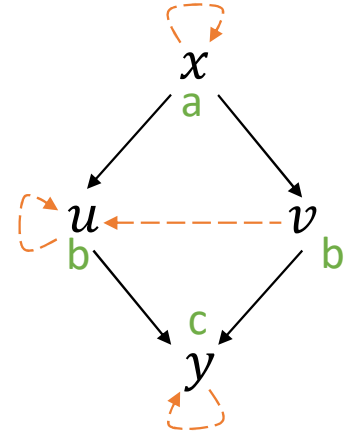
Query

$$Q(x,u,v,y) \leftarrow R(x,u), R(u,y), R(x,v), R(v,y)$$

Database

R	
a	b
b	c
b	d
c	e
d	e





endomorphism

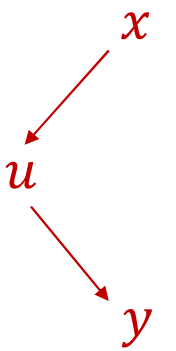


Image I

Algorithm

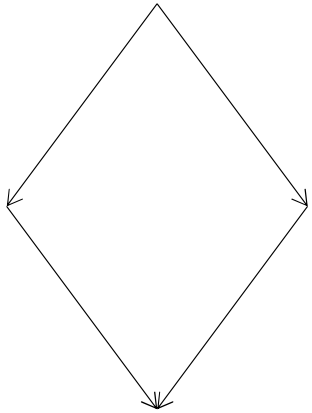
α = empty dictionary
for answer (x,u,y) to I :
  output (x,u,u,y)
  for v in $\alpha(x,y)$:
    output (x,u,v,y)
    output (x,v,u,y)
  $\alpha(x,y)$.insert(u)

Answers

I answers		
a	b	c
a	b	d
b	c	e
b	d	e

Q answers			
b	c	d	e
b	d	c	e
a	b	b	c
a	b	b	d
b	c	c	e
b	d	d	e

Examples: Full CQs

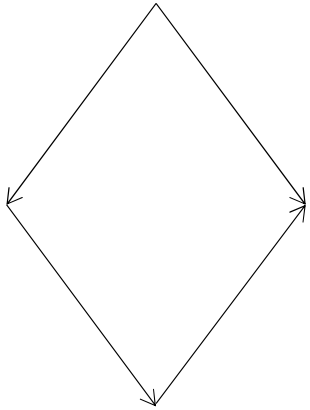


∈ Enum<lin,const>

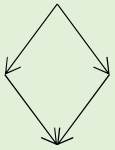
∉ Enum<lin,const> *

* assuming no triangle detection in linear time

Examples: Full CQs



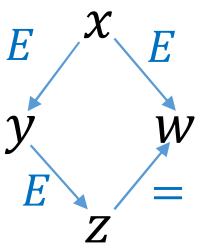
∈ Enum<lin,const>

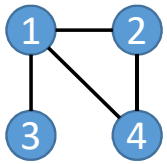


∉ Enum<lin,const> *

* assuming no triangle detection in linear time

Hardness Proof





$\boldsymbol{R(x,y) \leftarrow E}$

1	2
1	3
1	4
2	4

$\boldsymbol{R(y,z) \leftarrow E}$

1	2
1	3
1	4
2	4

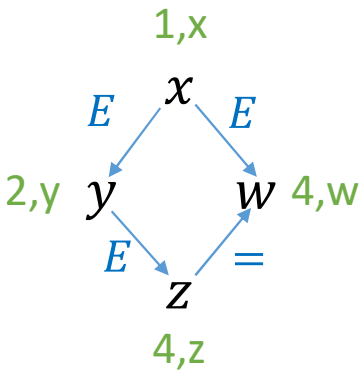
$\boldsymbol{R(x,w) \leftarrow E}$

1	2
1	3
1	4
2	4

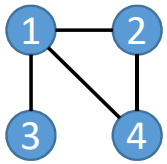
$\boldsymbol{R(w,z) \leftarrow =}$

1	1
2	2
3	3
4	4

Hardness Proof



Works because Q is a core!



$R(x, y) \leftarrow E$

1,x	2,y
1,x	3,y
1,x	4,y
2,x	4,y

$R(y, z) \leftarrow E$

1,y	2,z
1,y	3,z
1,y	4,z
2,y	4,z

$R(x, w) \leftarrow E$

1,x	2,w
1,x	3,w
1,x	4,w
2,x	4,w

$R(w, z) \leftarrow =$

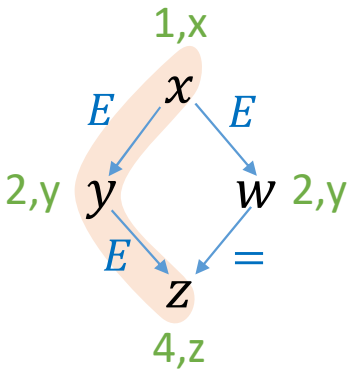
1,w	1,z
2,w	2,z
3,w	3,z
4,w	4,z

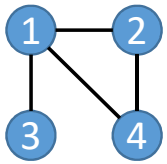
union $\Longrightarrow$

R

1,x	2,y
1,x	3,y
1,x	4,y
2,x	4,y
1,y	2,z
1,y	3,z
1,y	4,z
2,y	4,z
1,x	2,w
1,x	3,w
1,x	4,w
2,x	4,w
1,w	1,z
…	…

Hardness Proof Fails





$R(x, y) \leftarrow E$

1,x	2,y
1,x	3,y
1,x	4,y
2,x	4,y

$R(y, z) \leftarrow E$

1,y	2,z
1,y	3,z
1,y	4,z
2,y	4,z

$R(x, w) \leftarrow E$

1,x	2,w
1,x	3,w
1,x	4,w
2,x	4,w

$R(w, z) \leftarrow =$

1,w	1,z
2,w	2,z
3,w	3,z
4,w	4,z

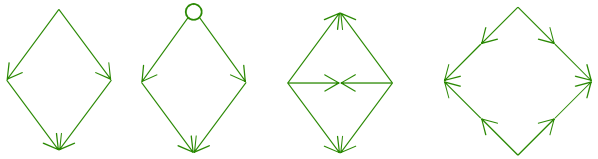
union $\Longrightarrow$

R

1,x	2,y
1,x	3,y
1,x	4,y
2,x	4,y
1,y	2,z
1,y	3,z
1,y	4,z
2,y	4,z
1,x	2,w
1,x	3,w
1,x	4,w
2,x	4,w
1,w	1,z
…	…

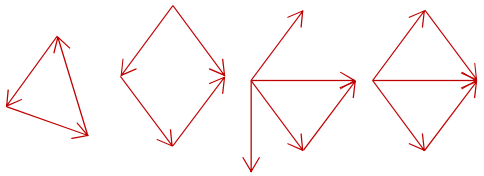
Sufficient and Necessary Conditions

Let Q be a full CQ.



If Q is a mirror, then $Q \in$ Enum<lin,const>

If Q has a cyclic core, then $Q \notin$ Enum<lin,const> *

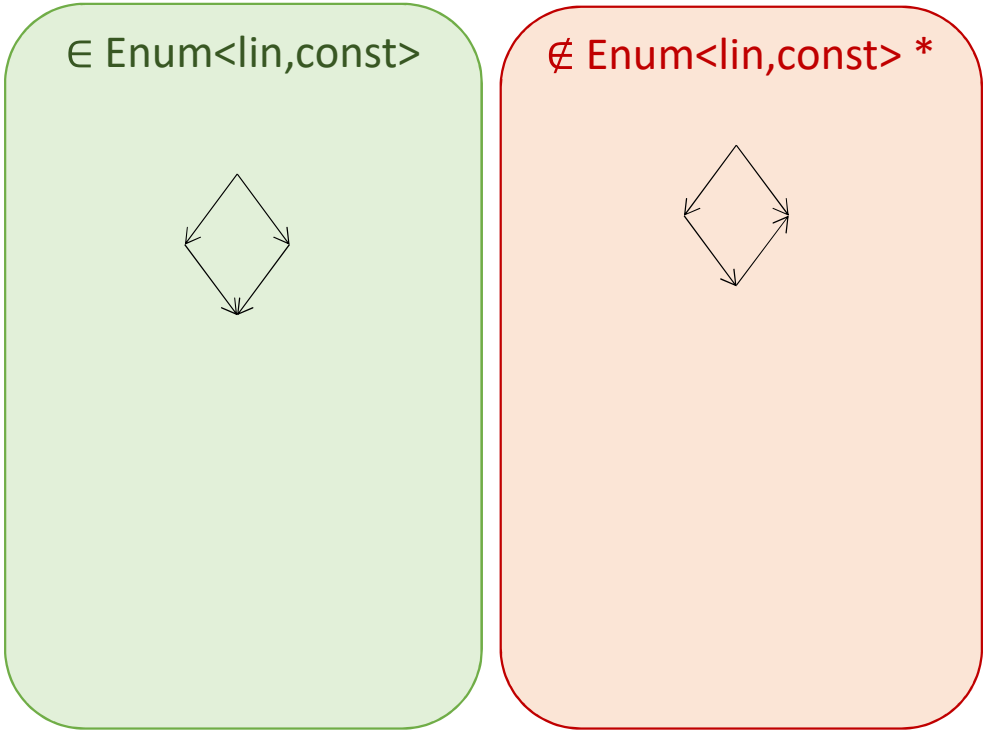


* assuming the sHyperclique hypothesis:
$\forall k \geq 3$ it is not possible to determine the existence of a k-hyperclique in a $(k-1)$-uniform hypergraph with m edges in time $O(m)$

Examples: Full CQs

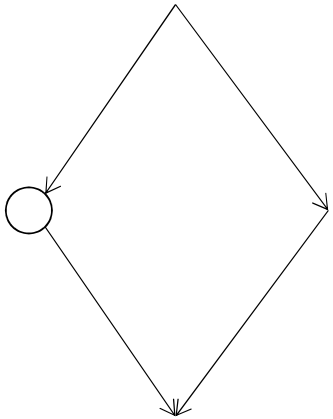
Unlike the self-join-free case,
may affect the complexity:

- reordering variables inside an atom



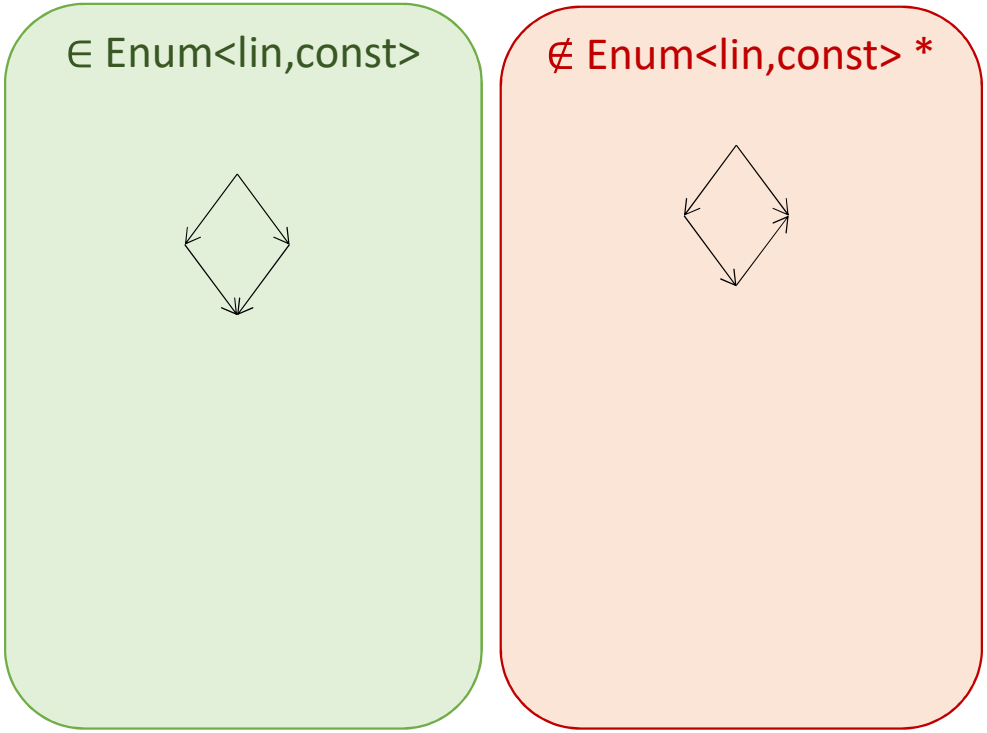
* assuming no triangle detection in linear time

Examples: Full CQs



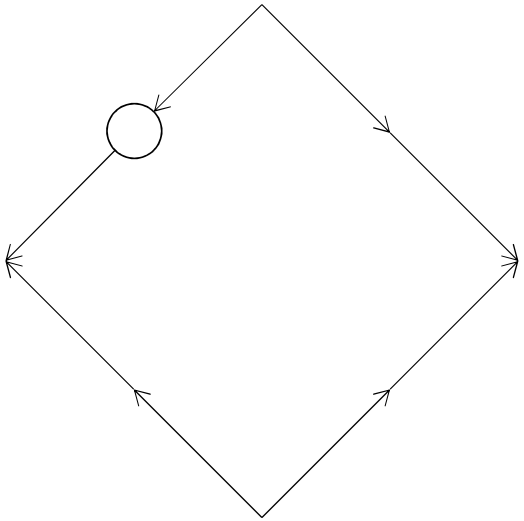
Unlike the self-join-free case,
may affect the complexity:

- reordering variables inside an atom



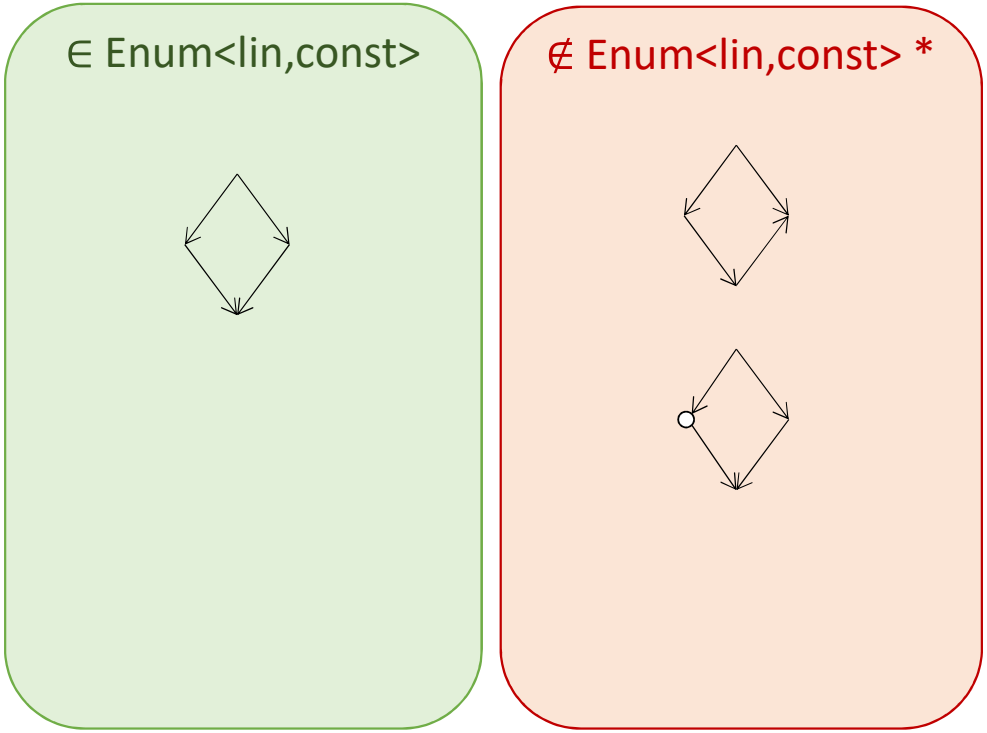
* assuming no triangle detection in linear time

Examples: Full CQs



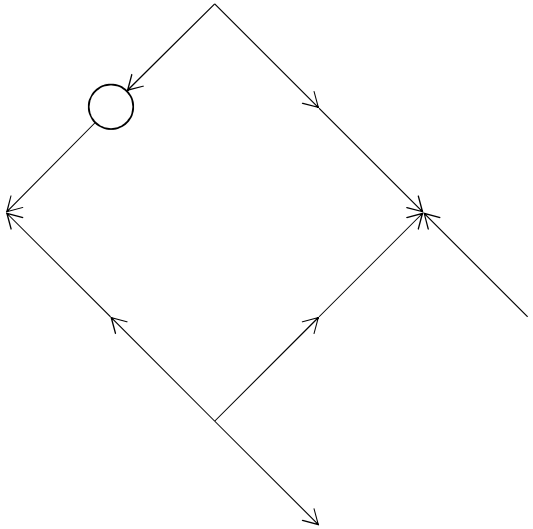
Unlike the self-join-free case,
may affect the complexity:

- reordering variables inside an atom
- introducing unary atoms



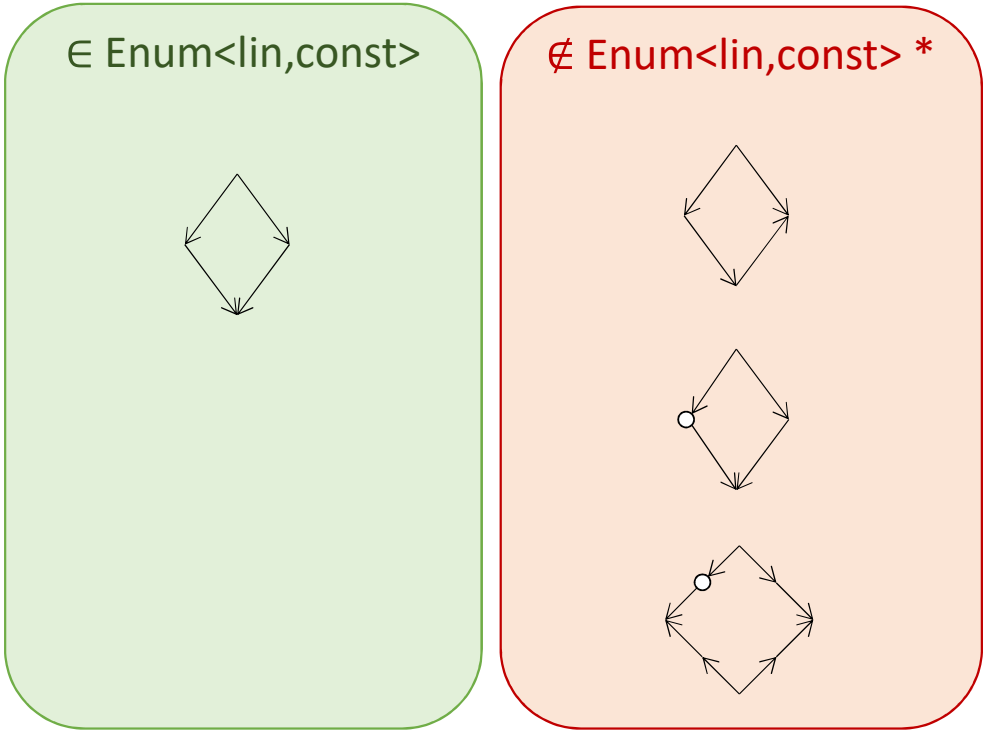
* assuming no triangle detection in linear time

Examples: Full CQs



Unlike the self-join-free case,
may affect the complexity:

- reordering variables inside an atom
- introducing unary atoms

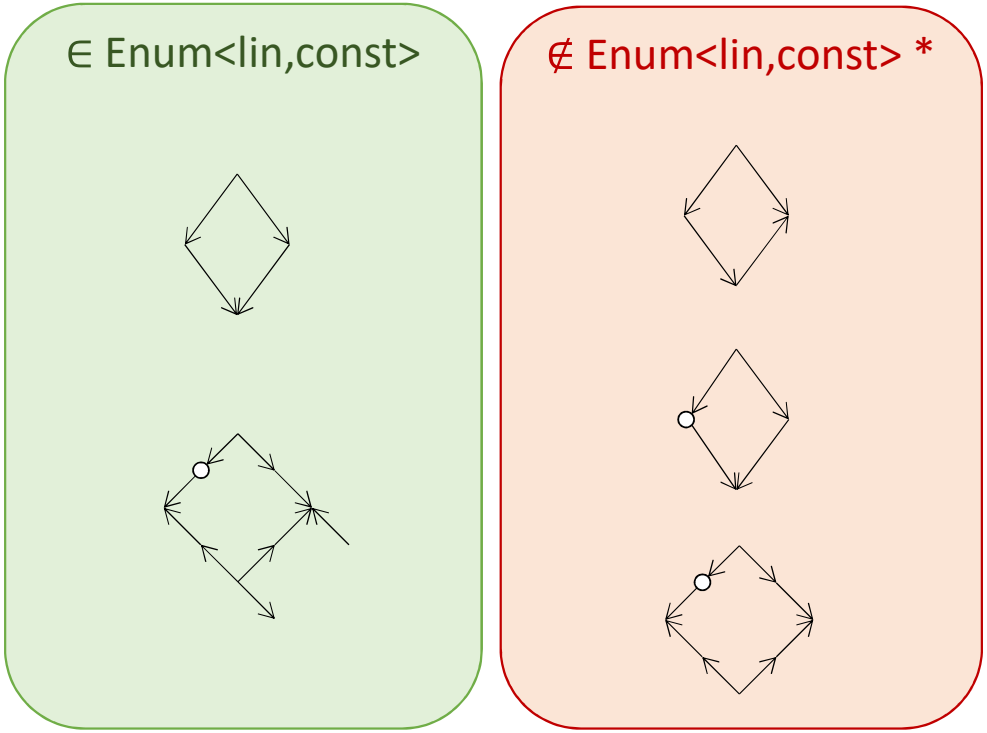


* assuming no triangle detection in linear time

Examples: Full CQs

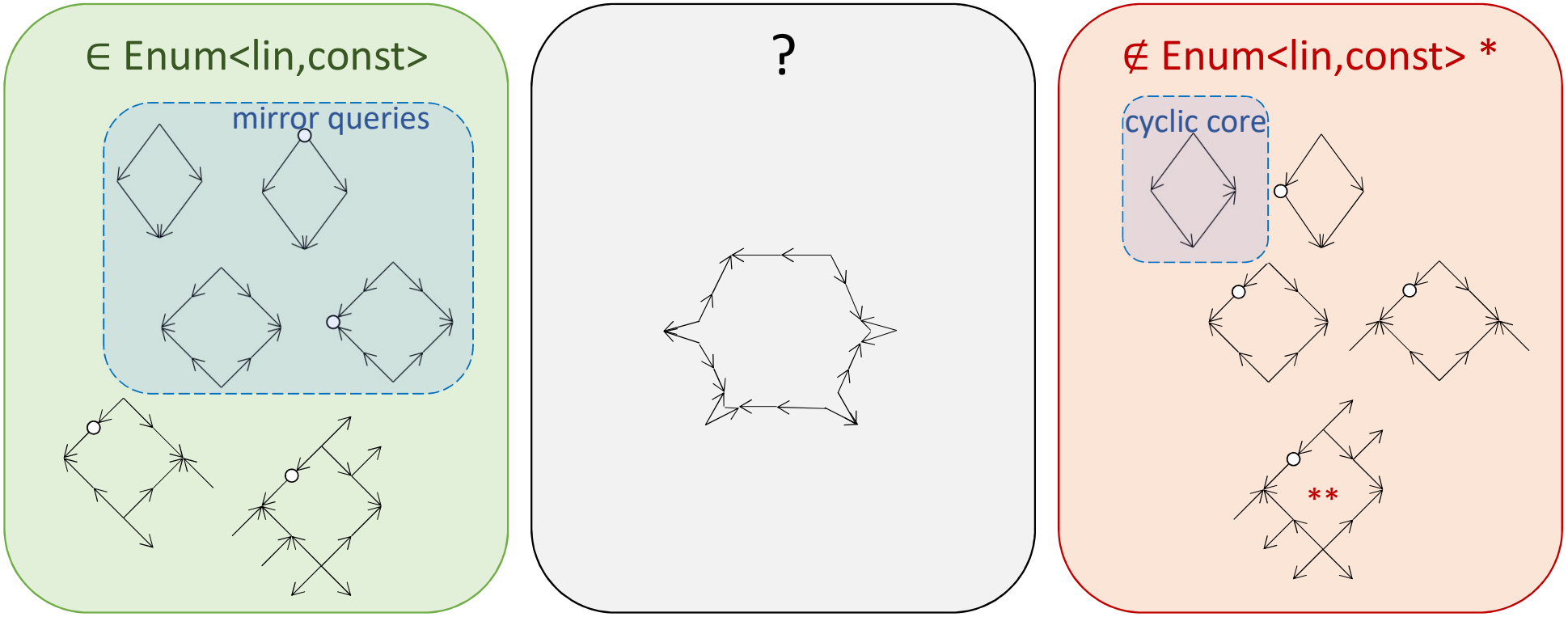
Unlike the self-join-free case,
may affect the complexity:

- reordering variables inside an atom
- introducing unary atoms
- introducing ‘spikes’



* assuming no triangle detection in linear time

Tractability of Self-Joins



* assuming no triangle detection in linear time

** assuming the Vertex-Unbalanced Triangle Detection Hypothesis [Bringmann, **C**; 22] :
$\forall \alpha \in (0,1]$ it is not possible to determine the existence of a triangle in a tripartite graph with $|V_1| = n$ and $|V_2| = |V_3| = \Theta(n^{\alpha})$ in time $O(n^{1+\alpha})$

Theorem: Low Arity

Let Q be a minimal CQ of arity ≤ 2.
The following are equivalent: *

- $Q \in$ Enum<lin,const>
- Self-join-free(Q) $\in$ Enum<lin,const>
- Q is free-connex acyclic

* assuming the Hyperclique hypothesis:
$\forall k \geq 3$ it is not possible to determine the existence of a k-hyperclique in a $(k-1)$-uniform hypergraph with n nodes in time $O(n^{k-1})$

Conclusion

- Summary
  - May affect the complexity: reordering variables, unary atoms, ‘spikes’.
  - Full CQs: cyclic-core $\subset$ hard; mirror $\subset$ easy.
  - Arity ≤ 2: self-joins don’t affect classification
  - In paper: linear delay
- Related Problems
  - Self-Joins don’t affect the complexity for: Counting [Dalmau, Jonsson; 2004], Lexicographic direct access [Bringmann, **C**, Mengel; 2023]
- Open
  - Specific examples in paper
  - Reducing space in algorithms
  - Acyclic non free-connex CQs and constant delay
  - Unary CQs and linear delay
  - CQs with disequalities