CNRS interview

The Fine-Grained Complexity of Evaluating Database Queries

Nofar CARMELI

The Fine-Grained Complexity of Evaluating Queries

- To use data: store & **query**

The Fine-Grained Complexity of Evaluating Queries

- To use data: store & **query**

- Big data requires extremely efficient algorithms
  - **Fine-grained** complexity: ‘polynomial’ is not enough

The Fine-Grained Complexity of Evaluating Queries

- To use data: store & **query**
- Big data requires extremely efficient algorithms
  - **Fine-grained** complexity: ‘polynomial’ is not enough
- What is the **most efficient way** of answering a database query?

In the Slides

- Updates to CV
- One result: Enumeration Complexity of Unions of Conjunctive Queries
- Research project

In the Slides

- Updates to CV
- **One result: Enumeration Complexity of Unions of Conjunctive Queries**
- Research project

Enumeration Complexity of UCQs

- Goal
- Overview
- Explanations
  - Easy ∪ Hard
    - Why isn’t it always hard?
    - When is it easy?
  - Hard ∪ Hard
    - Sometimes it is easy
    - Dichotomy

Enumeration Complexity of UCQs

- **Goal**
- Overview
- Explanations
  - Easy ∪ Hard
    - Why isn’t it always hard?
    - When is it easy?
  - Hard ∪ Hard
    - Sometimes it is easy
    - Dichotomy

Goal

Which Unions of Conjunctive Queries can be answered with optimal time guarantees?

Studied Queries

tutorials:

Person	Title
Alan Fekete	Making Consistency…
Suresh Venkatasu…	Algorithmic Fairness…

schedule:

Title	Day
Making Consistency…	Tue
Algorithmic Fairness…	Wed
Regularizing Conjunct…	Mon

Studied Queries

tutorials:

Person	Title
Alan Fekete	Making Consistency…
Suresh Venkatasu…	Algorithmic Fairness…

schedule:

Title	Day
Making Consistency…	Tue
Algorithmic Fairness…	Wed
Regularizing Conjunct…	Mon

$$Q_1(\text{Person}, \text{Day}) \leftarrow \text{tutorials}(\text{Person}, \text{Title}), \text{schedule}(\text{Title}, \text{Day})$$

Studied Queries

tutorials:

Person	Title
Alan Fekete	Making Consistency…
Suresh Venkatasu…	Algorithmic Fairness…

schedule:

Title	Day
Making Consistency…	Tue
Algorithmic Fairness…	Wed
Regularizing Conjunct…	Mon

Person	Day
Alan Fekete	Tue
Suresh Venkatasu…	Wed

$$Q_1(\text{Person}, \text{Day}) \leftarrow \text{tutorials}(\text{Person}, \text{Title}), \text{schedule}(\text{Title}, \text{Day})$$

Studied Queries

tutorials:

Person	Title
Alan Fekete	Making Consistency…
Suresh Venkatasu…	Algorithmic Fairness…

schedule:

Title	Day
Making Consistency…	Tue
Algorithmic Fairness…	Wed
Regularizing Conjunct…	Mon

research talks:

Person	Title
Pablo Barceló	Regularizing Conjunct…
Peter Lindner	Probabilistic Database…
Muhammad Tibi	Query Evaluation in…

Person	Day
Alan Fekete	Tue
Suresh Venkatasu…	Wed

$$Q_1(\text{Person}, \text{Day}) \leftarrow \text{tutorials}(\text{Person}, \text{Title}), \text{schedule}(\text{Title}, \text{Day})$$

Studied Queries

tutorials:

Person	Title
Alan Fekete	Making Consistency…
Suresh Venkatasu…	Algorithmic Fairness…

schedule:

Title	Day
Making Consistency…	Tue
Algorithmic Fairness…	Wed
Regularizing Conjunct…	Mon

research talks:

Person	Title
Pablo Barceló	Regularizing Conjunct…
Peter Lindner	Probabilistic Database…
Muhammad Tibi	Query Evaluation in…

Person	Day
Alan Fekete	Tue
Suresh Venkatasu…	Wed

$$Q_1(\text{Person}, \text{Day}) \leftarrow \text{tutorials}(\text{Person}, \text{Title}), \text{schedule}(\text{Title}, \text{Day})$$
$$Q_2(\text{Person}, \text{Day}) \leftarrow \text{research}(\text{Person}, \text{Title}), \text{schedule}(\text{Title}, \text{Day})$$

Studied Queries

tutorials:

Person	Title
Alan Fekete	Making Consistency…
Suresh Venkatasu…	Algorithmic Fairness…

schedule:

Title	Day
Making Consistency…	Tue
Algorithmic Fairness…	Wed
Regularizing Conjunct…	Mon

research talks:

Person	Title
Pablo Barceló	Regularizing Conjunct…
Peter Lindner	Probabilistic Database…
Muhammad Tibi	Query Evaluation in…

Person	Day
Alan Fekete	Tue
Suresh Venkatasu…	Wed

Person	Day
Pablo Barceló	Mon
Martin Grohe	Mon
Muhammad Tibi	Mon

$$Q_1(\text{Person}, \text{Day}) \leftarrow \text{tutorials}(\text{Person}, \text{Title}), \text{schedule}(\text{Title}, \text{Day})$$
$$Q_2(\text{Person}, \text{Day}) \leftarrow \text{research}(\text{Person}, \text{Title}), \text{schedule}(\text{Title}, \text{Day})$$

Studied Queries

tutorials:

Person	Title
Alan Fekete	Making Consistency…
Suresh Venkatasu…	Algorithmic Fairness…

schedule:

Title	Day
Making Consistency…	Tue
Algorithmic Fairness…	Wed
Regularizing Conjunct…	Mon

research talks:

Person	Title
Pablo Barceló	Regularizing Conjunct…
Peter Lindner	Probabilistic Database…
Muhammad Tibi	Query Evaluation in…

Person	Day
Alan Fekete	Tue
Suresh Venkatasu…	Wed

Person	Day
Pablo Barceló	Mon
Martin Grohe	Mon
Muhammad Tibi	Mon

$$Q_1(\text{Person}, \text{Day}) \leftarrow \text{tutorials}(\text{Person}, \text{Title}), \text{schedule}(\text{Title}, \text{Day})$$
$$Q_2(\text{Person}, \text{Day}) \leftarrow \text{research}(\text{Person}, \text{Title}), \text{schedule}(\text{Title}, \text{Day})$$
$$Q_3 = Q_1 \cup Q_2$$

Studied Queries

tutorials:

Person	Title
Alan Fekete	Making Consistency…
Suresh Venkatasu…	Algorithmic Fairness…

schedule:

Title	Day
Making Consistency…	Tue
Algorithmic Fairness…	Wed
Regularizing Conjunct…	Mon

research talks:

Person	Title
Pablo Barceló	Regularizing Conjunct…
Peter Lindner	Probabilistic Database…
Muhammad Tibi	Query Evaluation in…

Person	Day
Alan Fekete	Tue
Suresh Venkatasu…	Wed
Pablo Barceló	Mon
Martin Grohe	Mon
Muhammad Tibi	Mon

$$Q_1(\text{Person}, \text{Day}) \leftarrow \text{tutorials}(\text{Person}, \text{Title}), \text{schedule}(\text{Title}, \text{Day})$$
$$Q_2(\text{Person}, \text{Day}) \leftarrow \text{research}(\text{Person}, \text{Title}), \text{schedule}(\text{Title}, \text{Day})$$
$$Q_3 = Q_1 \cup Q_2$$

Complexity of Queries

- Query = problem

Complexity of Queries

- Query = problem
- Time complexity, data complexity, RAM model

Complexity of Queries

- Query = problem
- Time complexity, data complexity, RAM model
- Achievable time bounds:

Complexity of Queries

- Query = problem
- Time complexity, data complexity, RAM model
- Achievable time bounds:
  - Need to print every answer (|OUT| >> |IN|)

Complexity of Queries

- Query = problem
- Time complexity, data complexity, RAM model
- Achievable time bounds:
  - Need to print every answer (|OUT| >> |IN|)
  - Need to read the input before the first answer

Complexity of Queries

- Query = problem
- Time complexity, data complexity, RAM model
- Achievable time bounds:
  - Need to print every answer (|OUT| >> |IN|)
  - Need to read the input before the first answer
- $\mathbf{DelayC_{lin}}$: solvable in linear preprocessing and constant delay

Complexity of Queries

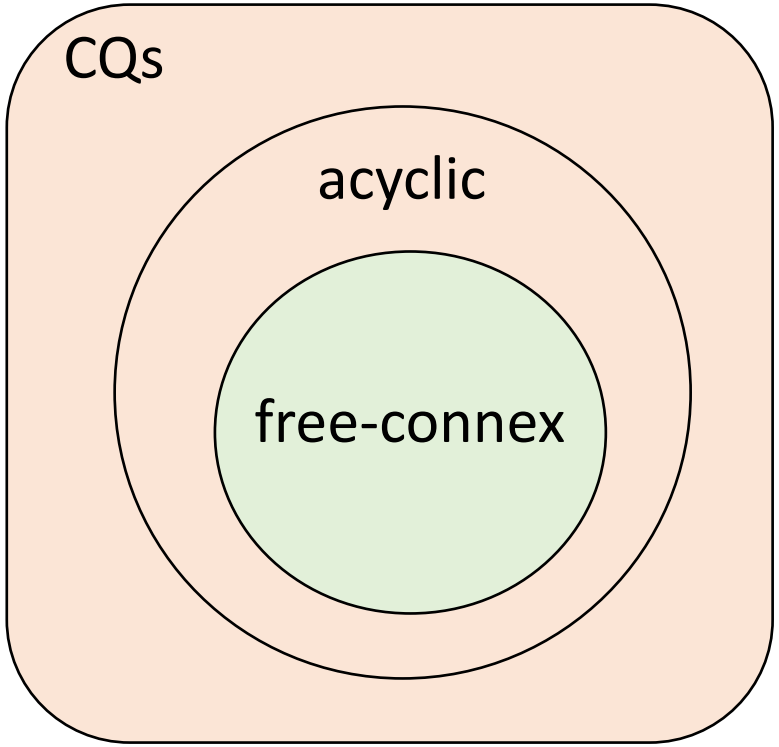
- Query = problem
- Time complexity, data complexity, RAM model
- Achievable time bounds:
  - Need to print every answer (|OUT| >> |IN|)
  - Need to read the input before the first answer
- $\mathbf{DelayC_{lin}}$: solvable in linear preprocessing and constant delay

Which queries are in $\mathrm{DelayC_{lin}}$?

Starting Point

[BaganDurandGrandjean CSL'2007]
[Brault-Baron 2013]

CQs: $\in DelayC_{lin} \Leftrightarrow^{*}$ free-connex



* Hardness results assume:
(1) no self-joins
(2) hardness of Boolean matrix multiplication and hyperclique

Free-connex Definition

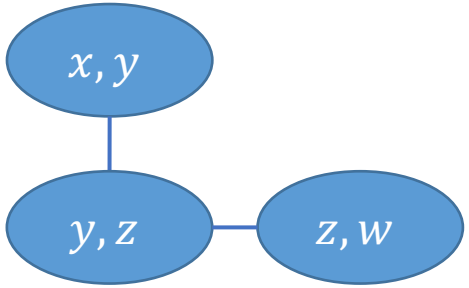
1. a node for every atom

2. tree

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

acyclic

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



Free-connex Definition

1. a node for every atom

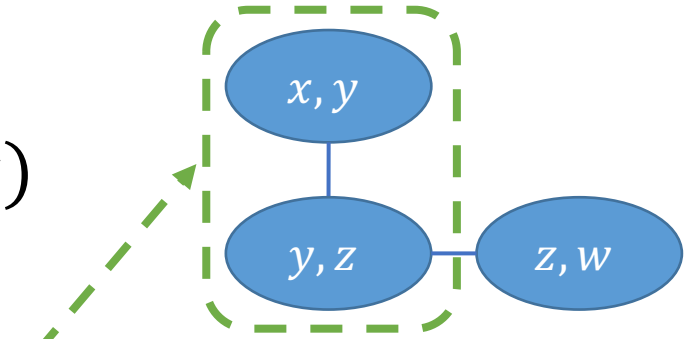
2. tree

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

acyclic

free − connex

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



4. a subtree with exactly the free variables

Free-connex Definition

1. a node for every atom

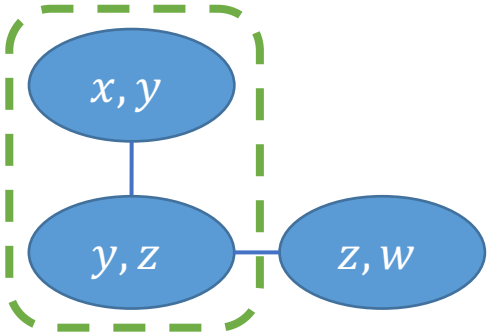
2. tree

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

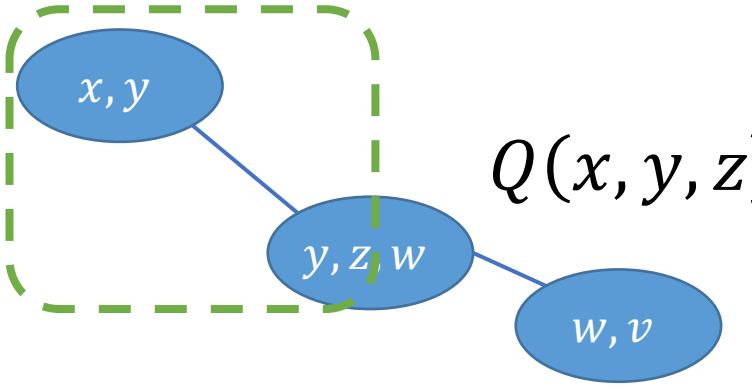
acyclic

free − connex

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



4. a subtree with exactly the free variables



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

free − connex

Free-connex Definition

1. a node for every atom
   possibly also subsets

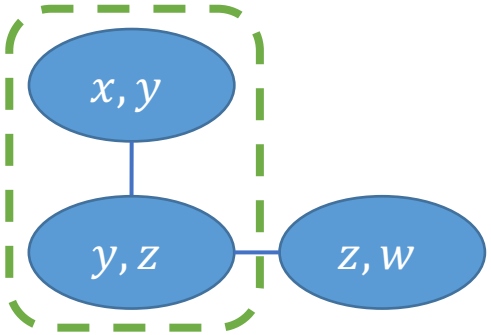
2. tree

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

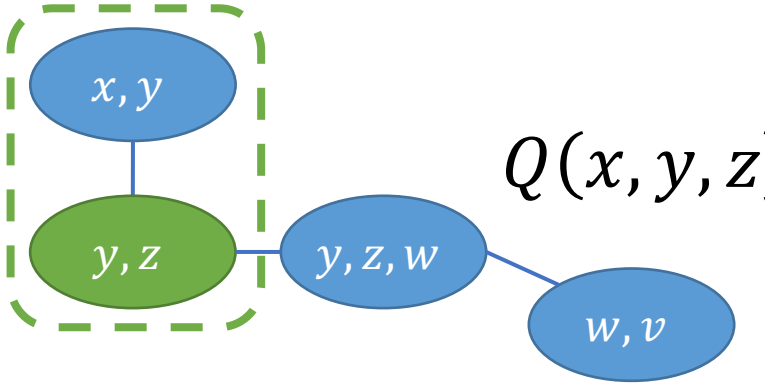
acyclic

free − connex

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



4. a subtree with exactly the free variables



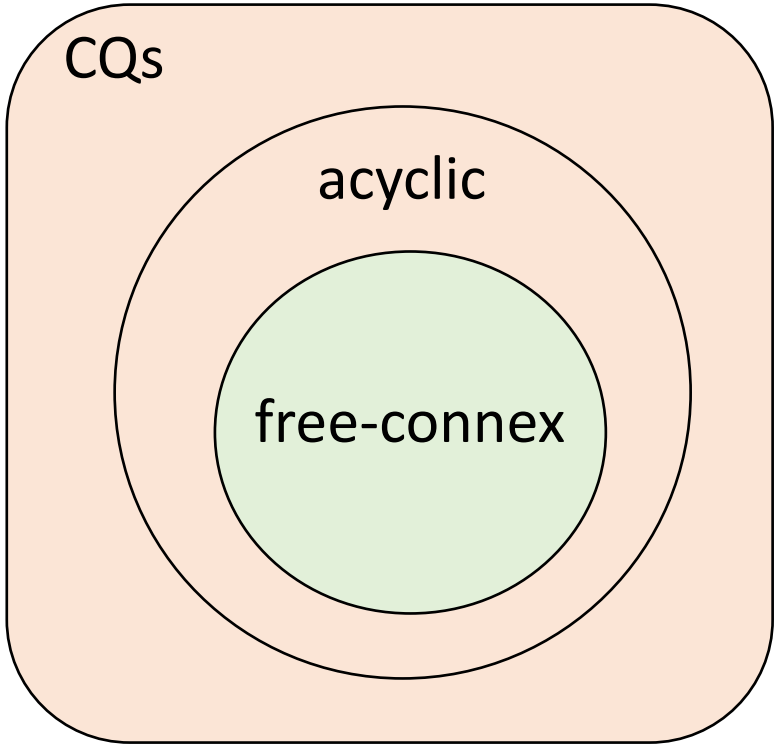
$$Q(x, y, z) \leftarrow R_1(x, y), R_2(y, z, w), R_3(w, v)$$

free − connex

Starting Point

[BaganDurandGrandjean CSL'2007]
[Brault-Baron 2013]

CQs: $\in DelayC_{lin} \Leftrightarrow^{*}$ free-connex

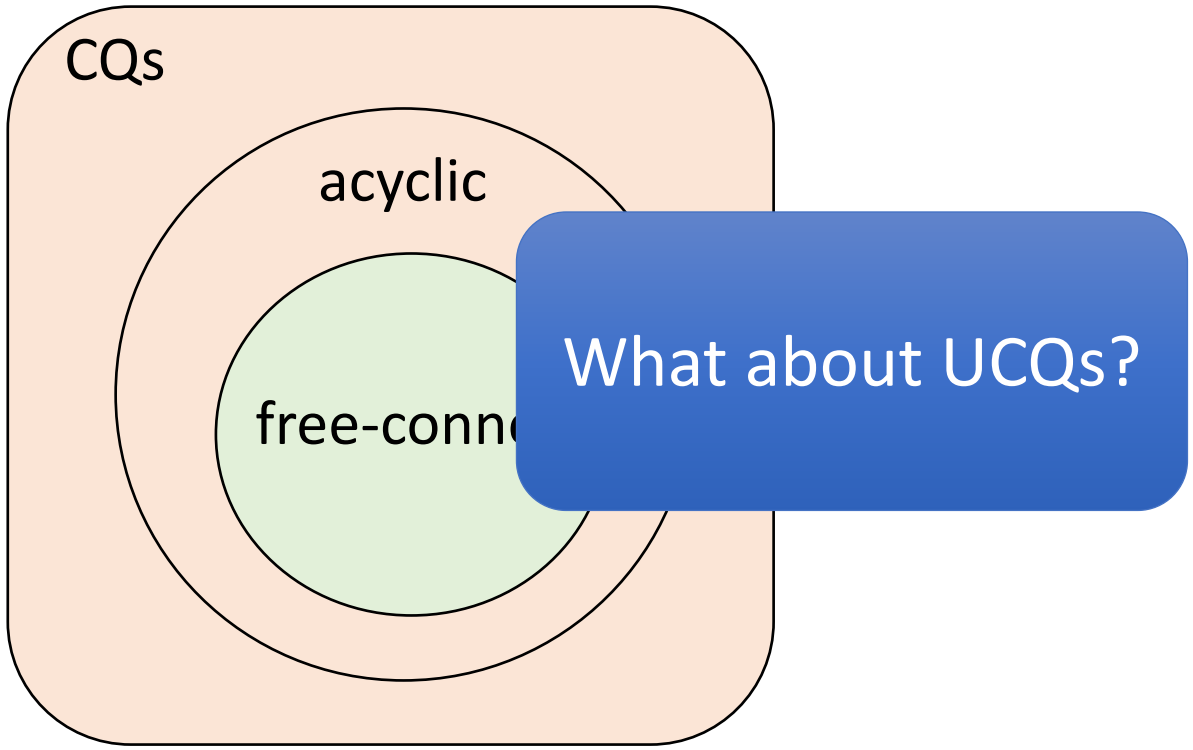


* Hardness results assume:
(1) no self-joins
(2) hardness of Boolean matrix multiplication and hyperclique

Starting Point

[BaganDurandGrandjean CSL'2007]
[Brault-Baron 2013]

CQs: $\in DelayC_{lin} \Leftrightarrow^{*}$ free-connex



* Hardness results assume:
(1) no self-joins
(2) hardness of Boolean matrix multiplication and hyperclique

Enumeration Complexity of UCQs

- Goal
- **Overview**
- Explanations
  - Easy ∪ Hard
    - Why isn’t it always hard?
    - When is it easy?
  - Hard ∪ Hard
    - Sometimes it is easy
    - Dichotomy

Cases for UCQs

All CQs are Easy

Some Easy, Some Hard

All CQs are Hard

Cases for UCQs

All CQs are Easy

Some Easy, Some Hard

All CQs are Hard

Cases for UCQs

All CQs are Easy

Some Easy, Some Hard

All CQs are Hard

Cases for UCQs

All CQs are Easy

Some Easy, Some Hard

All CQs are Hard

unions **with** a hard CQ
can be equivalent to
unions **without** a hard CQs

Cases for UCQs

All CQs are Easy

Some Easy, Some Hard

All CQs are Hard

unions **with** a hard CQ
can be equivalent to
unions **without** a hard CQs

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

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

Cases for UCQs

All CQs are Easy

always easy

Some Easy, Some Hard

sometimes hard

All CQs are Hard

unions **with** a hard CQ
can be equivalent to
unions **without** a hard CQs

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

non free − connex

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

free − connex

Cases for UCQs

All CQs are Easy

always easy

Some Easy, Some Hard

sometimes hard

All CQs are Hard

unions **with** a hard CQ
can be equivalent to
unions **without** a hard CQs

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

non free − connex

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

free − connex

$$Q_1 \subseteq Q_2$$

Cases for UCQs

All CQs are Easy

always easy

Some Easy, Some Hard

sometimes hard

All CQs are Hard

unions **with** a hard CQ
can be equivalent to
unions **without** a hard CQs

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

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

non free − connex

free − connex

$$Q_1 \subseteq Q_2 \quad \Longrightarrow \quad Q_1 \cup Q_2 = Q_2$$

Cases for UCQs

All CQs are Easy

always easy

Some Easy, Some Hard

sometimes hard

sometimes easy

All CQs are Hard

unions **with** a hard CQ
can be equivalent to
unions **without** a hard CQs

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

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

non free − connex

free − connex

$$Q_1 \subseteq Q_2 \quad \Longrightarrow \quad Q_1 \cup Q_2 = Q_2$$

Cases for UCQs

non-redundant unions?

Cases for UCQs

non-redundant unions?

Claimed [ICDT 2018]:
hard if contains a hard CQ

Cases for UCQs

All CQs are Easy

Some Easy, Some Hard

All CQs are Hard

some **non-redundant** unions
with a hard CQ
are **easy**

Cases for UCQs

* Even for non-redundant unions

some **non-redundant** unions
with a hard CQ
are **easy**

Cases for UCQs

* Even for non-redundant unions

If each CQ in Q is hard
and
there is no body-isomorphism

$\Rightarrow \quad Q \notin DelayC_{lin}$

Cases for UCQs

* Even for non-redundant unions

UCQs containing **only hard** CQs can be **easy**!

Cases for UCQs

* Even for non-redundant unions

UCQs containing **only hard** CQs can be **easy**!

Enumeration Complexity of UCQs

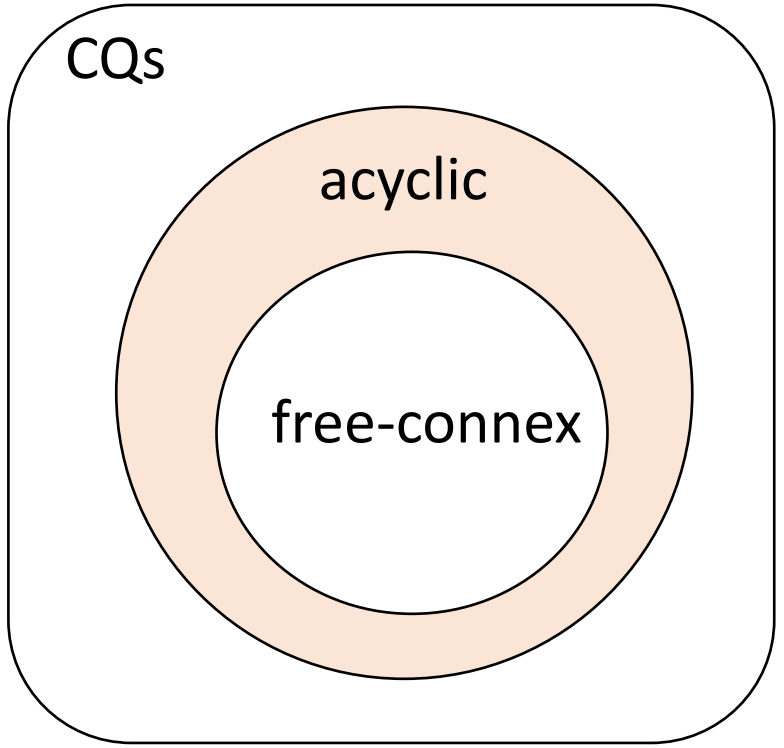
- Goal
- Overview
- Explanations
  - Easy ∪ Hard
    - **Why isn't it always hard?**
    - When is it easy?
  - Hard ∪ Hard
    - Sometimes it is easy
    - Dichotomy

Lower Bound

[BaganDurandGrandjean CSL'2007]

Acyclic non-free-connex:

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



Lower Bound

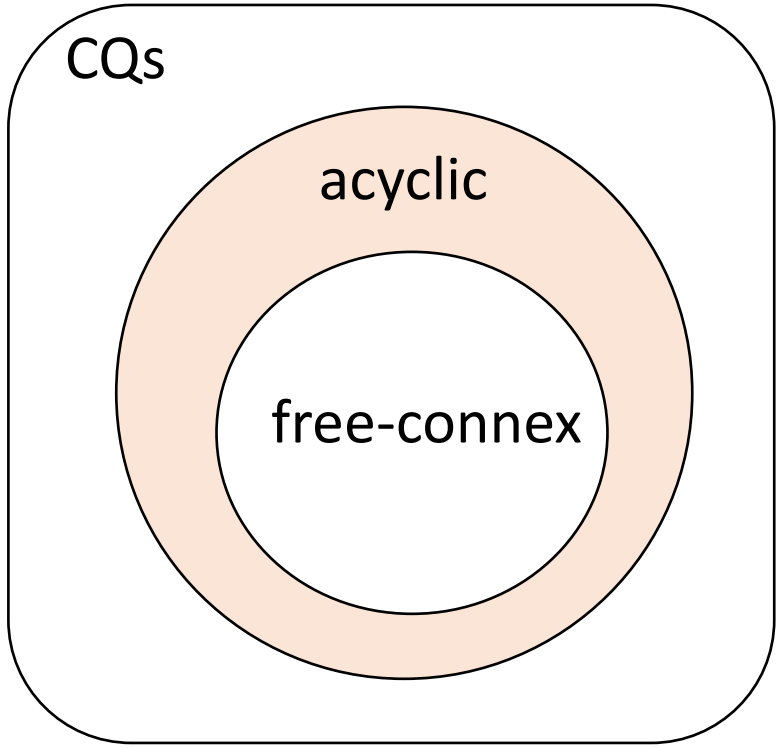
[BaganDurandGrandjean CSL'2007]

Assumption: Boolean $n \times n$ matrices cannot be multiplied in time $O(n^2)$

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} ? & ? \\ ? & ? \end{pmatrix}$$

Acyclic non-free-connex:

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



Lower Bound

[BaganDurandGrandjean CSL'2007]

Assumption: Boolean $n \times n$ matrices cannot be multiplied in time $O(n^2)$

$$\underset{A}{\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}} \underset{B}{\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}} = \begin{pmatrix} ? & ? \\ ? & ? \end{pmatrix}$$

Acyclic non-free-connex:

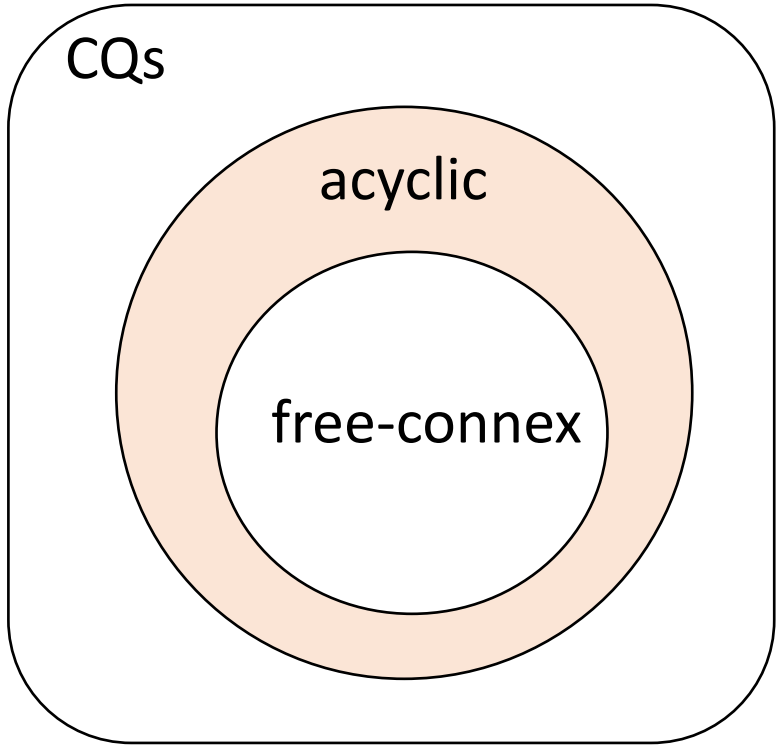
$$Q(x, z) \leftarrow R_1(x, y), R_2(y, z)$$

A

R	C
1	1
1	2
2	2

B

R	C
1	2
2	2

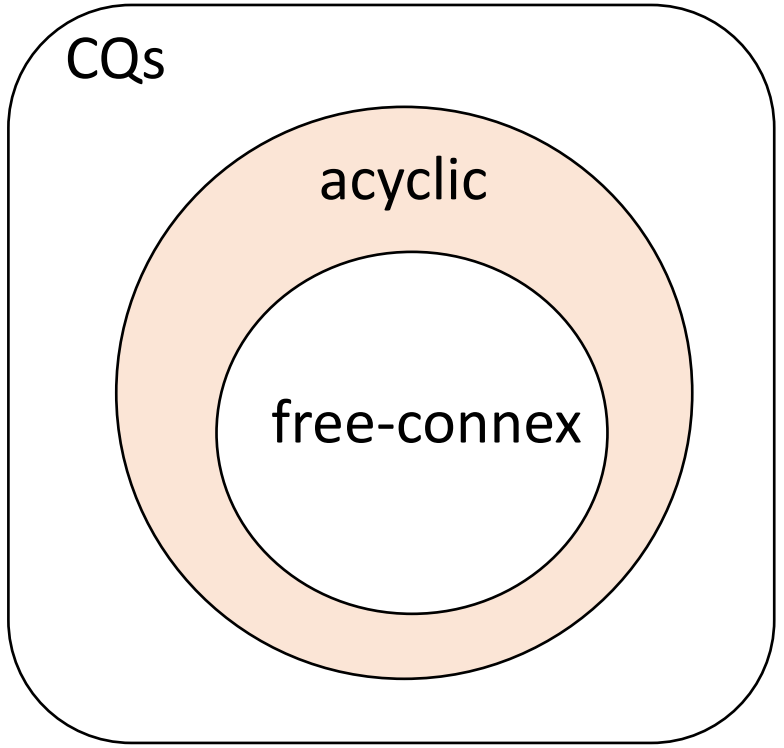


Lower Bound

[BaganDurandGrandjean CSL'2007]

Assumption: Boolean $n \times n$ matrices cannot be multiplied in time $O(n^2)$

$$\underset{A}{\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}} \underset{B}{\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}} = \begin{pmatrix} ? & ? \\ ? & ? \end{pmatrix}$$



Acyclic non-free-connex:

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

Q

R	C
1	2
2	2

A

R	C
1	1
1	2
2	2

B

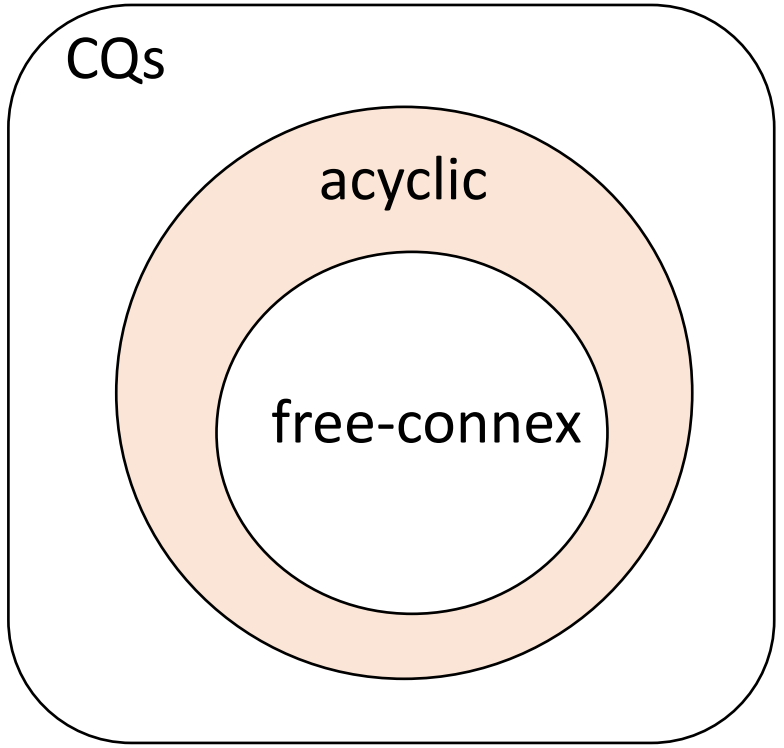
R	C
1	2
2	2

Lower Bound

[BaganDurandGrandjean CSL'2007]

Assumption: Boolean $n \times n$ matrices cannot be multiplied in time $O(n^2)$

$$\underset{A}{\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}} \underset{B}{\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}} = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$$



Acyclic non-free-connex:

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

Q

R	C
1	2
2	2

A

R	C
1	1
1	2
2	2

B

R	C
1	2
2	2

Lower Bound

[BaganDurandGrandjean CSL'2007]

Assumption: Boolean $n \times n$ matrices cannot be multiplied in time $O(n^2)$

$$\underset{A}{\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}} \underset{B}{\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}} = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$$

Acyclic non-free-connex:

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

Q

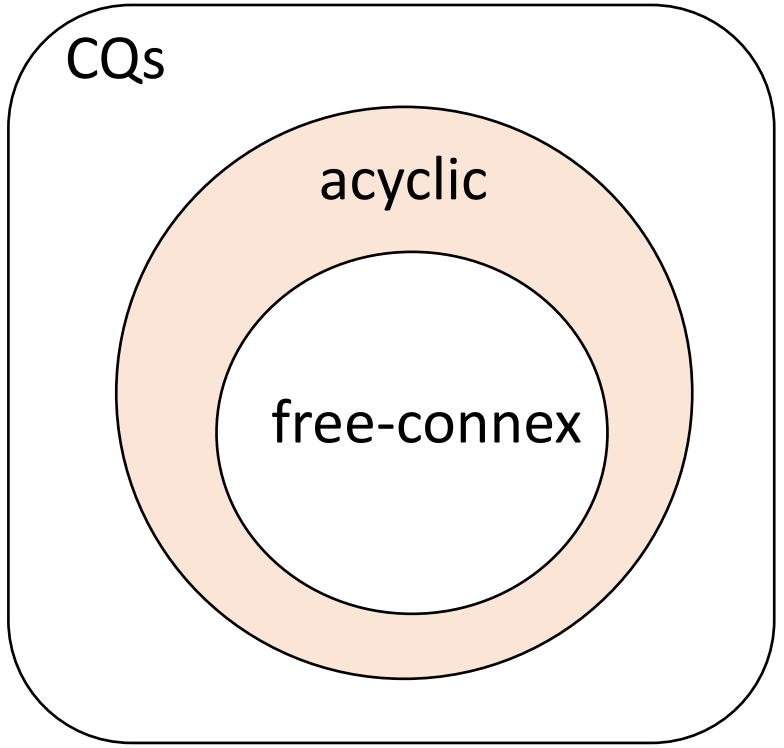
R	C
1	2
2	2

A

R	C
1	1
1	2
2	2

B

R	C
1	2
2	2



Lower Bound

[BaganDurandGrandjean CSL'2007]

Assumption: Boolean $n \times n$ matrices cannot be multiplied in time $O(n^2)$

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$$

A B

Acyclic non-free-connex:

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

Q

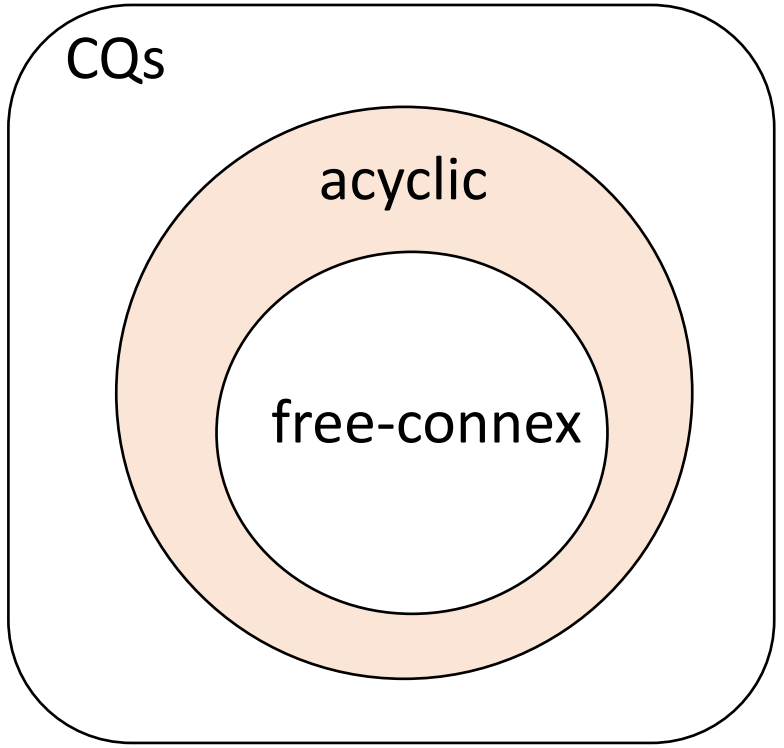
R	C
1	2
2	2

A

R	C
1	1
1	2
2	2

B

R	C
1	2
2	2



Lower Bound

[BaganDurandGrandjean CSL'2007]

Assumption: Boolean $n \times n$ matrices cannot be multiplied in time $O(n^2)$

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$$

A B

Acyclic non-free-connex:

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

$\notin \mathbf{DelayC_{lin}}$

Q

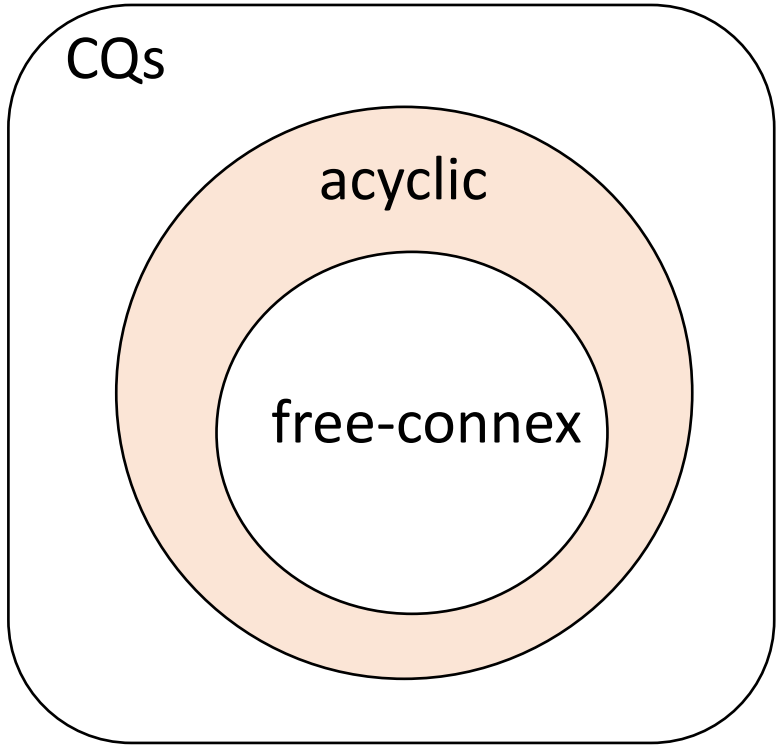
R	C
1	2
2	2

A

R	C
1	1
1	2
2	2

B

R	C
1	2
2	2



Within Unions

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

not free connex

Within Unions

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

not free connex

R_1	
1	1
1	2
2	2

R_2	
1	2
2	2

Within Unions

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

not free connex

Q_1		
1	2	⊥
2	2	⊥

R_1	
1	1
1	2
2	2

R_2	
1	2
2	2

R_3	
2	⊥

Within Unions

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

$$\cup$$

$$Q_2(a, b, c) \leftarrow R_1(a, b), R_2(b, c)$$

not free connex

Q_1		
1	2	⊥
2	2	⊥

R_1	
1	1
1	2
2	2

R_2	
1	2
2	2

R_3	
2	⊥

Within Unions

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

$$\cup$$

$$Q_2(a, b, c) \leftarrow R_1(a, b), R_2(b, c)$$

not free connex

Q_1		
1	2	⊥
2	2	⊥

Q_2		
1	1	2
1	2	2
2	2	2

R_1	
1	1
1	2
2	2

R_2	
1	2
2	2

R_3	
2	⊥

Within Unions

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

U

$$Q_2(a, b, c) \leftarrow R_1(a, b), R_2(b, c)$$

not free connex

Q_1		
1	2	⊥
2	2	⊥

Q_2		
1	1	2
1	2	2
2	2	2

$O(n^3)$ solutions:
The computation does not contradict the assumption

R_1	
1	1
1	2
2	2

R_2	
1	2
2	2

R_3	
2	⊥

Within Unions

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

$$\cup$$

$$Q_2(a, b, c) \leftarrow R_1(a, b), R_2(b, c)$$

not free connex

Q_1		
1	2	⊥
2	2	⊥

Q_2		
1	1	2
1	2	2
2	2	2

$O(n^3)$ solutions:
The computation does not contradict the assumption

R_1	
1	1
1	2
2	2

R_2	
1	2
2	2

R_3	
2	⊥

The hardness results do not hold within a union

Enumeration Complexity of UCQs

- Goal
- Overview
- Explanations
  - Easy ∪ Hard
    - Why isn't it always hard?
    - **When is it easy?**
  - Hard ∪ Hard
    - Sometimes it is easy
    - Dichotomy

Providing Variables

non free − connex

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

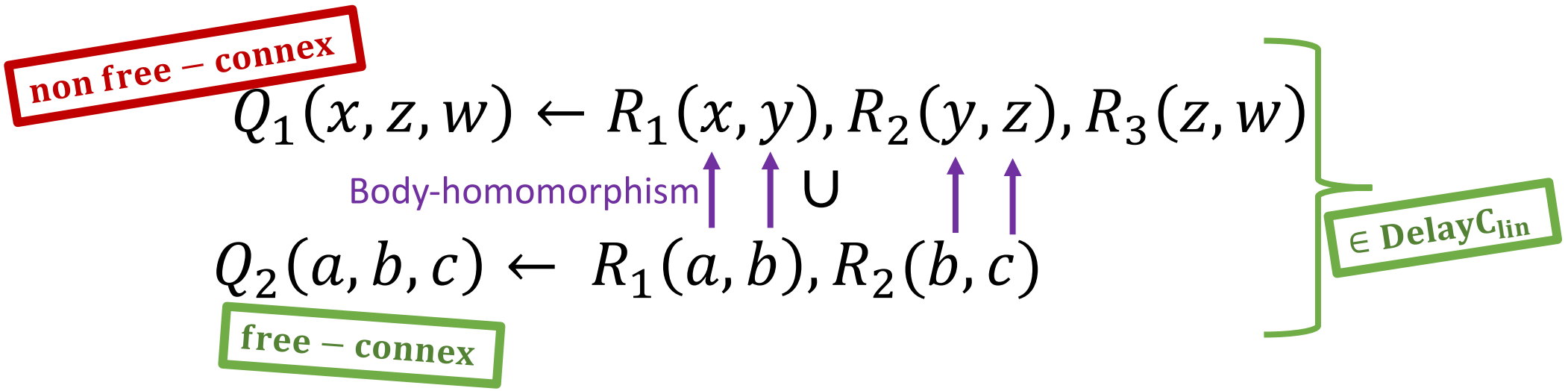
$$\cup$$

$$Q_2(a, b, c) \leftarrow R_1(a, b), R_2(b, c)$$

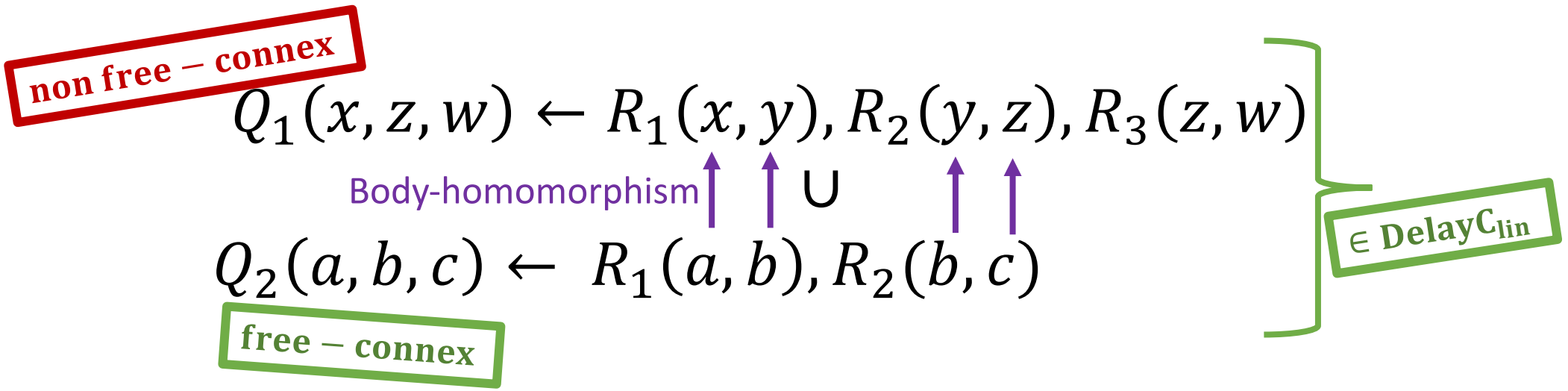
free − connex

$\in \mathbf{DelayC_{lin}}$

Providing Variables

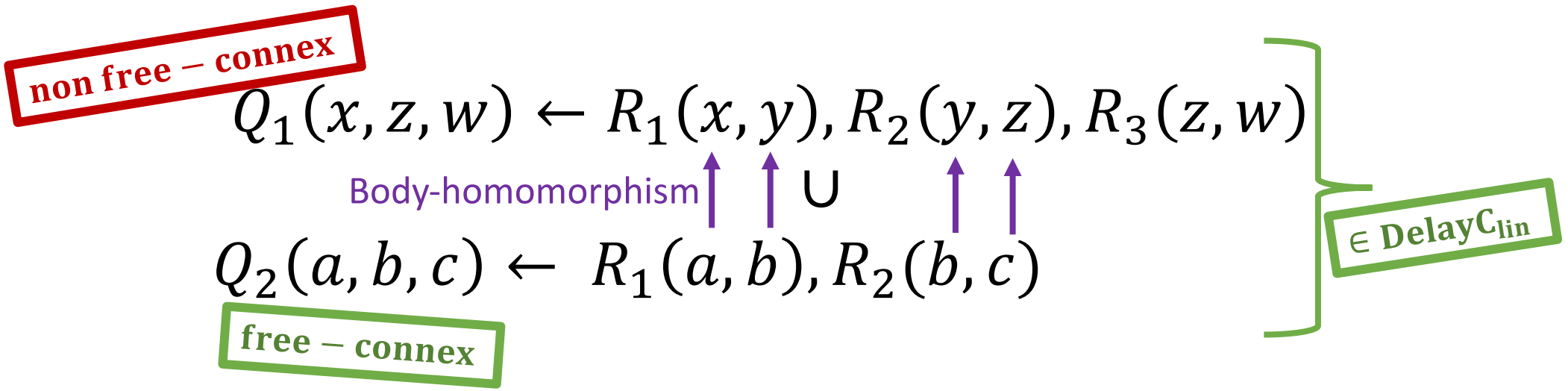


Providing Variables



Q_2 computes $R_1 \bowtie R_2$

Providing Variables



Q_2 computes $R_1 \bowtie R_2 \implies Q_2$ provides $\{x, y, z\}$ to Q_1

Providing Variables

non free − connex

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

Body-homomorphism ↑ ↑ ∪ ↑ ↑

$$Q_2(a, b, c) \leftarrow R_1(a, b), R_2(b, c)$$

free − connex

$\in \mathbf{DelayC_{lin}}$

Q_2 computes $R_1 \bowtie R_2 \implies Q_2$ provides $\{x, y, z\}$ to Q_1

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

Providing Variables

non free − connex

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

Body-homomorphism ↑ ↑ ∪ ↑ ↑

$$Q_2(a, b, c) \leftarrow R_1(a, b), R_2(b, c)$$

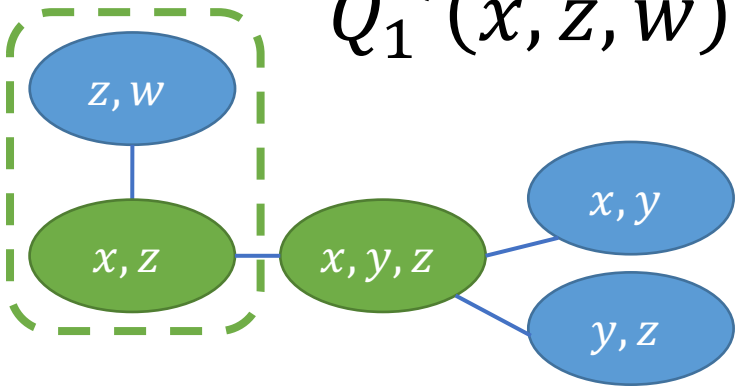
free − connex

$\in \mathbf{DelayC_{lin}}$

Q_2 computes $R_1 \bowtie R_2 \implies Q_2$ provides $\{x, y, z\}$ to Q_1

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

free − connex



Enumeration Complexity of UCQs

- Goal
- Overview
- Explanations
  - Easy ∪ Hard
    - Why isn’t it always hard?
    - When is it easy?
  - **Hard** ∪ **Hard**
    - Sometimes it is easy
    - Dichotomy

Hard ∪ Hard = Easy

- Example: CQs with **isomorphic bodies**.

$$Q_1(x, z, w, u) \leftarrow R_1(x, y), R_2(y, z), R_3(z, w), R_4(w, u)$$
$$Q_2(x, y, z, u) \leftarrow R_1(x, y), R_2(y, z), R_3(z, w), R_4(w, u)$$

Hard ∪ Hard = Easy

- Example: CQs with **isomorphic bodies**.

hard part

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

$$Q_2(x, y, z, u) \leftarrow R_1(x, y), R_2(y, z), R_3(z, w), R_4(w, u)$$

hard part

Hard ∪ Hard = Easy

- Example: CQs with **isomorphic bodies**.

hard part

$$Q_1(x, z, w, u) \leftarrow R_1(x, y), R_2(y, z), R_3(z, w), R_4(w, u)$$
$$Q_2(x, y, z, u) \leftarrow R_1(x, y), R_2(y, z), R_3(z, w), R_4(w, u)$$

hard part

	Step	Output	Side Effect
1	Solve Q_2'	$\subseteq Q_2$	Find $R_1 \bowtie R_2$
2	Solve Q_1^+	Q_1	Find $R_3 \bowtie R_4$
3	Solve Q_2^+	Q_2	

Theorem

We can classify every UCQ of the form Hard ∪ Hard:

Both queries are body-isomorphic, free-path guarded and bypass guarded $\iff$ Both queries become free-connex by adding provided atoms $\iff$ $\in \mathrm{DelayC_{lin}}$

* Hardness results assume:
(1) no self-joins
(2) hardness of Boolean matrix multiplication, hyperclique detection and 4-clique detection.

In the Slides

- Updates to CV
- One result: Enumeration Complexity of Unions of Conjunctive Queries
- **Research project**

The Fine-Grained Complexity of Evaluating Queries

- To use data: store & **query**
- Big data requires extremely efficient algorithms
  - **Fine-grained** complexity: ‘polynomial’ is not enough
- What is the **most efficient way** of answering a database query?

The Fine-Grained Complexity of Evaluating Queries

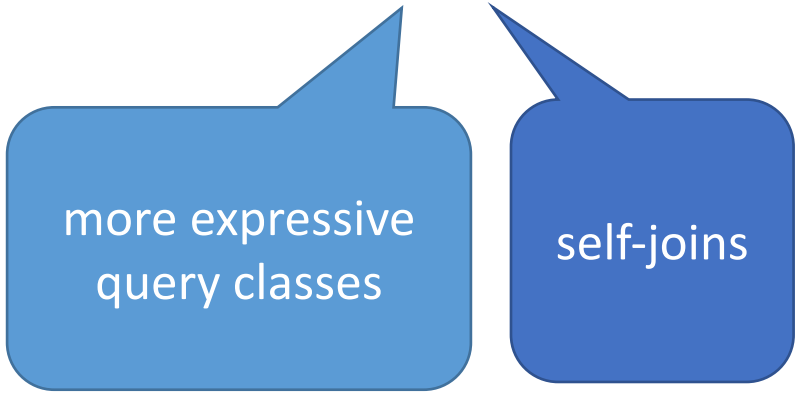
What is the most efficient way
of answering a database query?

The Fine-Grained Complexity of Evaluating Queries

What is the most efficient way
of answering a database **query**?

The Fine-Grained Complexity of Evaluating Queries

What is the most efficient way
of answering a database **query**?



[C+, PODS'19][C+, TODS'21]

The Fine-Grained Complexity of Evaluating Queries

What is the most efficient **way of answering** a database query?

The Fine-Grained Complexity of Evaluating Queries

What is the most efficient **way of answering** a database query?

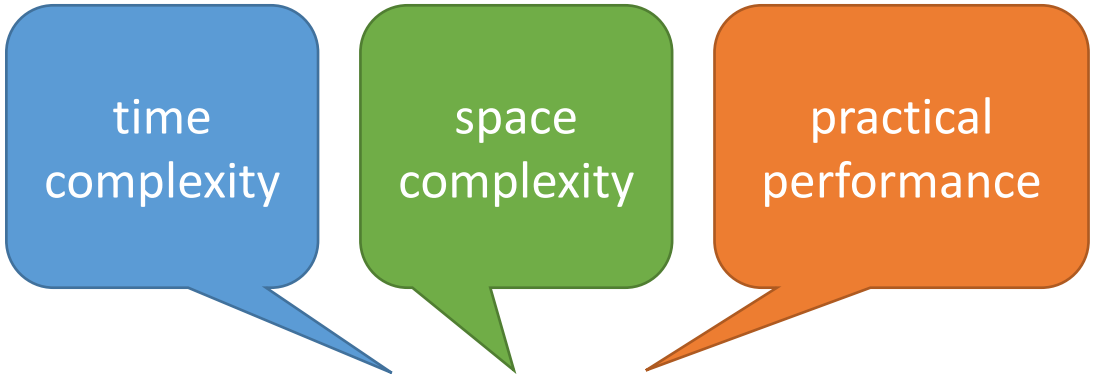
enumeration
vs.
direct access

[C+, PODS'20]
[C+, PODS'21]

The Fine-Grained Complexity of Evaluating Queries

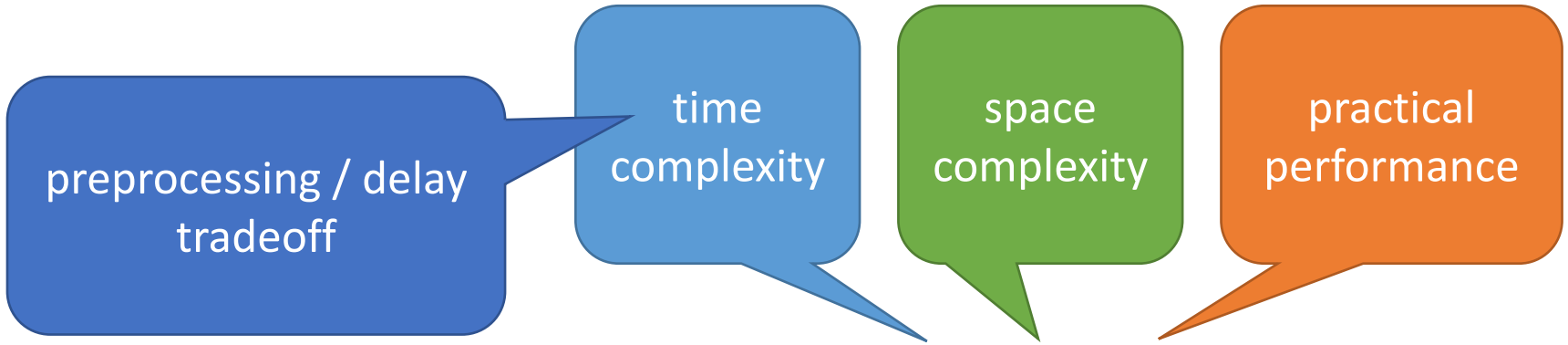
What is the most **efficient** way
of answering a database query?

The Fine-Grained Complexity of Evaluating Queries



What is the most **efficient** way of answering a database query?

The Fine-Grained Complexity of Evaluating Queries



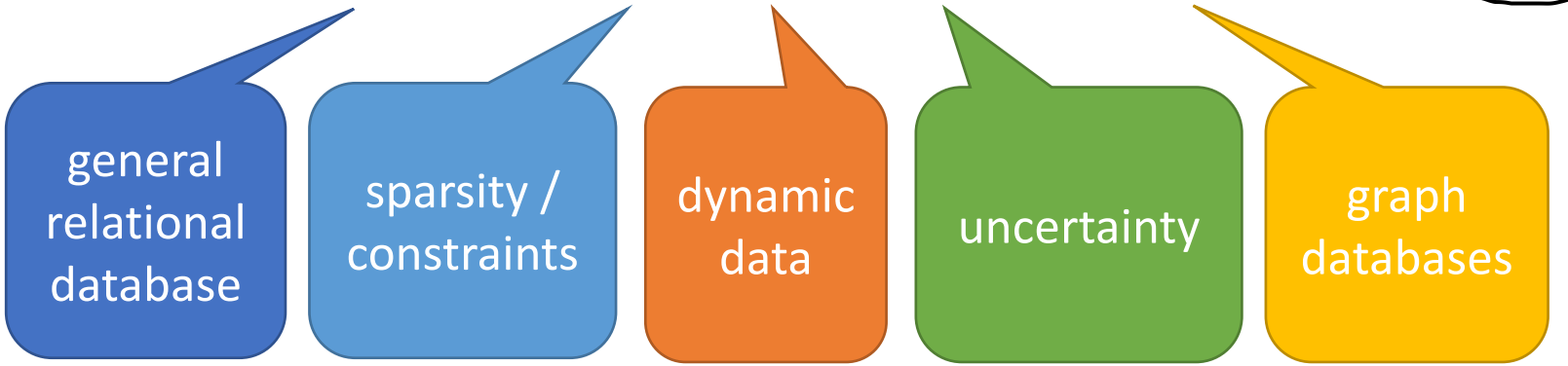
What is the most **efficient** way of answering a database query?

The Fine-Grained Complexity of Evaluating Queries

What is the most efficient way
of answering a **database** query?

The Fine-Grained Complexity of Evaluating Queries

What is the most efficient way of answering a **database** query?



Ontologies

[C+, ICDT'18][C+, TOCS'19]

The Fine-Grained Complexity of Evaluating Queries

What is the most efficient way of answering a database query?

In the Slides

- Updates to CV
- One result: Enumeration Complexity of Unions of Conjunctive Queries
- Research project