Fine-Grained Dichotomies for Evaluating Database Queries

Nofar Carmeli

Plan

- Introduction
- Join queries
  - Basic dichotomy
  - Self-joins
- Higher expressivity
  - Conjunctive queries
  - Unions of conjunctive queries
- Other Evaluation Tasks

Plan

- **Introduction**
- Join queries
  - Basic dichotomy
  - Self-joins
- Higher expressivity
  - Conjunctive queries
  - Unions of conjunctive queries
- Other Evaluation Tasks

Example: Join Query

Problem

Description	Room
Moisture	5/129
Broken ceiling	Cafeteria
Missing board	5/127

Office

Room	Person
5/127	Nofar
5/128	Florent
5/128	Guillaume
5/129	David
5/129	Akira

Contact

Person	Email
Nofar	nc@lirmm.fr
Florent	ft@lirmm.fr
Guillaume	gpk@lirmm.fr
David	dc@lirmm.fr

$$Q(E,P,R,D) \leftarrow \text{Problem}(D,R), \text{Office}(R,P), \text{Contact}(P,E)$$

$$\{(E,P,R,D) | (D,R) \in \text{Problem}, (R,P) \in \text{Office}, (P,E) \in \text{Contact}\}$$

Email	Person	Room	Description
nc@lirmm.fr	Nofar	5/127	Missing board
dc@lirmm.fr	David	5/129	Moisture

Challenges

- Many answers
- Many intermediate answers

R

x	y
a1	b1
a2	b1
a3	b1

S

y	z
b1	c1
b1	c2

T

x	z
a2	c1
a4	c2

dangling tuples

$$Q_1(x,y,z) \leftarrow \mathrm{R}(x,y),\mathrm{S}(y,z)$$

x	y	z
a1	b1	c1
a1	b1	c2
a2	b1	c1
a2	b1	c2
a3	b1	c1
a3	b1	c2

$$Q_2(x,y,z) \leftarrow \mathrm{R}(x,y),\mathrm{S}(y,z),\mathrm{T}(x,z)$$

x	y	z
a2	b1	c1

Example: Algorithm

Problem

Description	Room
Moisture	5/129
~~Broken ceiling~~	~~Cafeteria~~
Missing board	5/127

Office

Room	Person
5/127	Nofar
~~5/128~~	~~Florent~~
~~5/128~~	~~Guillaume~~
5/129	David
~~5/129~~	~~Akira~~

Contact

Person	Email
Nofar	nc@lirmm.fr
~~Florent~~	~~ft@lirmm.fr~~
~~Guillaume~~	~~gpk@lirmm.fr~~
David	dc@lirmm.fr

$$Q(E,P,R,D) \leftarrow \text{Problem}(D,R), \text{Office}(R,P), \text{Contact}(P,E)$$

Email	Person	Room	Description
nc@lirmm.fr	Nofar	5/127	Missing board
dc@lirmm.fr	David	5/129	Moisture

Example: Algorithm Fails

Registration

Student	Exam
Anna	algorithms
Thomas	databases

Staff

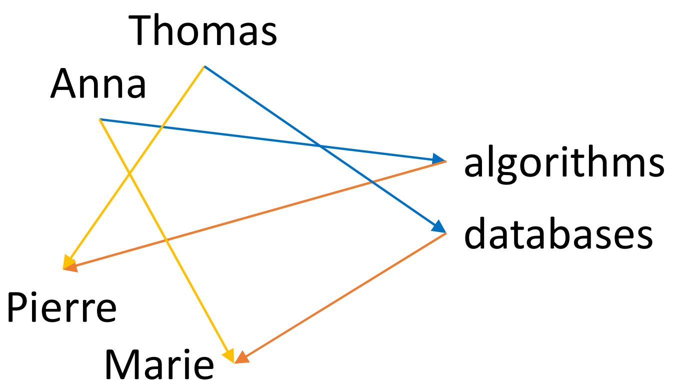
Exam	Professor
algorithms	Pierre
databases	Marie

COI

Student	Professor
Thomas	Pierre
Anne	Marie

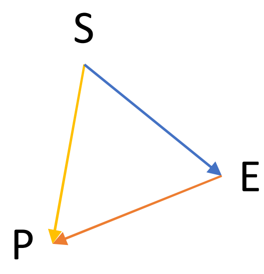
$$Q(S,E,P) \leftarrow \text{Registration}(S,E), \text{Staff}(E,P), \text{COI}(S,P)$$

Database



No query answers

Query



Complexity Guarantees

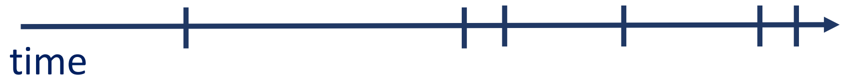
- Data complexity
  - input = database
  - query size = constant
- Possibly: output $\gg$ input

$$|answers| = |database|^{|query|}$$

- Minimal requirements:
  - Linear time (to read input)
  - Constant time per answer (to print output)
- RAM model
- We allow log factors

Complexity Guarantees

- Worst-case-optimal total time [Atserias, Grohe, Marx; FOCS 08]
  - Linear in input + worst-case output

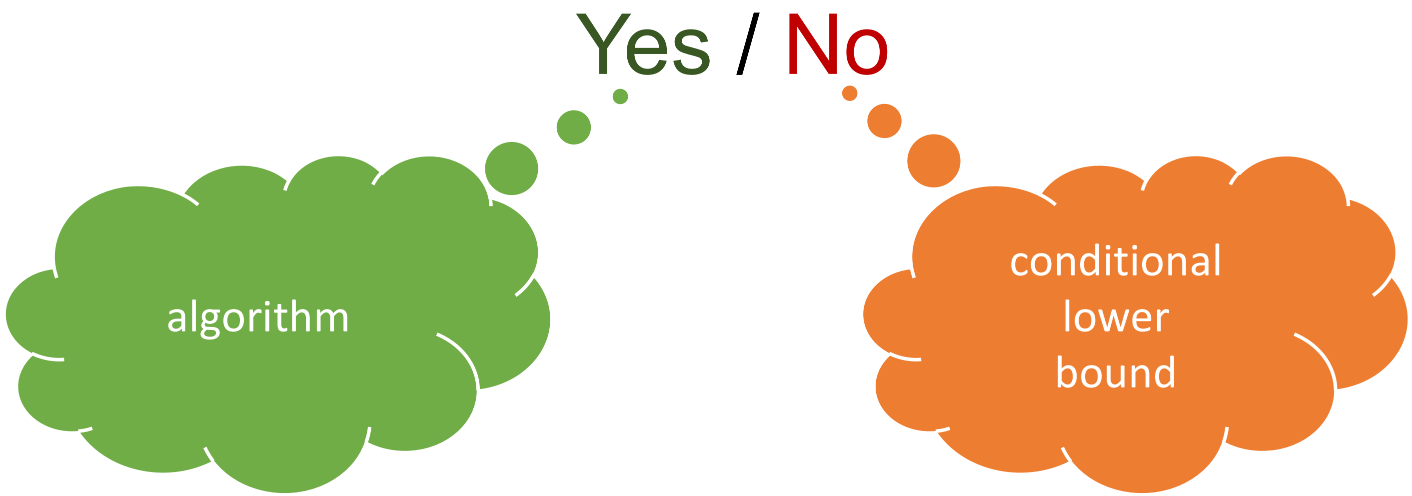


- Instance-optimal total time (also relevant)
  - Linear in input + output

- Enumeration (“ideal”; our focus)
  - Preprocessing: linear in input
  - Delay: constant

Research Question

- Goal: Given a query, what is the most efficient algorithm?
- Type of results:
  Can we solve a task for a given query in a given time complexity?



Plan

- Introduction
- **Join queries**
  - **Basic dichotomy**
  - Self-joins
- Higher expressivity
  - Conjunctive queries
  - Unions of conjunctive queries
- Other Evaluation Tasks

Acyclicity

- A query that has a join tree is called <u>acyclic</u>

2. tree

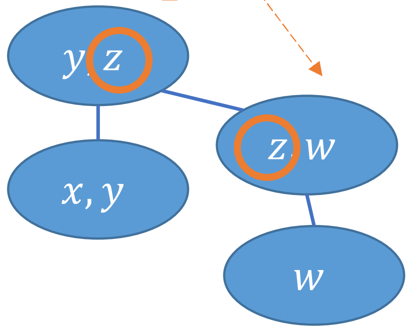
3. For every variable:
the nodes containing it form a subtree

1. a node for every atom

Query: $Q_1(x, y, z, w) \leftarrow R(x, y), S(y, z), T(z, w), U(w)$

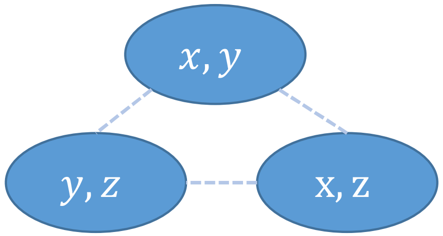
acyclic

Join Tree:



cyclic

Query: $Q_2(x, y, z) \leftarrow R(x, y), S(y, z), T(x, z)$



Dichotomy

[BaganDurandGrandjean; CSL 07]
[Brault-Baron 2013]
[Bringmann, **C**, Mengel; PODS 22]

- Given a join query Q,

If Q is acyclic, $Q \in \mathrm{Enum}<\mathrm{lin},\mathrm{const}>$

If Q is cyclic, $Q \notin \mathrm{Enum}<\mathrm{lin},\mathrm{const}>$*

* no self-joins, assuming sHyperclique or Zero-Clique

Acyclic Joins

[Yannakakis 81]

- An efficient algorithm for acyclic joins
  1. Find a join tree and set a root
  2. Remove dangling tuples
  3. Join

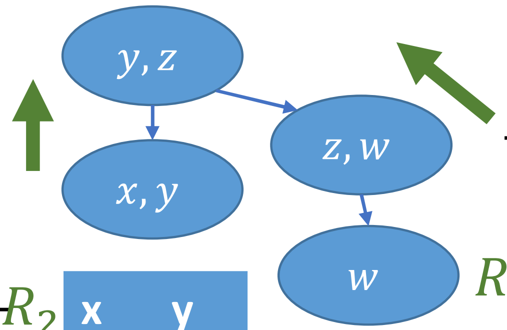
1. Leaf-to-root:
   $r_{parent} \leftarrow r_{parent} \ltimes r_{chil}$
2. Root-to-leaf:
   $r_{child} \leftarrow r_{child} \ltimes r_{parent}$

R_1

y	z
y1	z1
y1	z2
y2	z2
~~y3~~	~~z1~~
~~y4~~	~~z3~~

R_2

x	y
x1	y1
x1	y2
x2	y2
~~x2~~	~~y4~~



R_3

z	w
z1	w1
z2	w2
~~z3~~	~~w3~~

R_4

w
w1
w2

No dangling tuples!

Acyclic Joins

- An efficient algorithm for acyclic joins
  1. Find a join tree and set a root
  2. Remove dangling tuples
  3. Join

```
for t1 in R1:
  for t2 in R2 matching t1:
    for t3 in R3 matching t1,t2:
      for t4 in R4 matching t1,t2,t3:
        output t1,t2,t3,t4
```

R_1

y	z
y1	z1
y1	z2
y2	z2
~~y3~~	~~z1~~
~~y4~~	~~z3~~

R_2

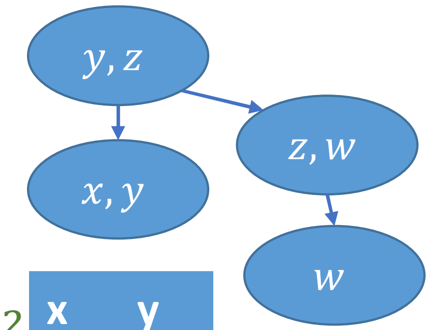
x	y
x1	y1
x1	y2
x2	y2
~~x2~~	~~y4~~

R_3

z	w
z1	w1
z2	w2
~~z3~~	~~w3~~

R_4

w
w1
w2



Dichotomy

[BaganDurandGrandjean; CSL 07]
[Brault-Baron 2013]
[Bringmann, **C**, Mengel; PODS 22]

- Given a join query Q,

If Q is acyclic, $Q \in \mathrm{Enum}<\mathrm{lin,const}>$

If Q is cyclic, $Q \notin \mathrm{Enum}<\mathrm{lin,const}>$*

* no self-joins, assuming sHyperclique or Zero-Clique

Example: Algorithm Fails

Registration

Student	Exam
Anna	algorithms
Thomas	databases

Staff

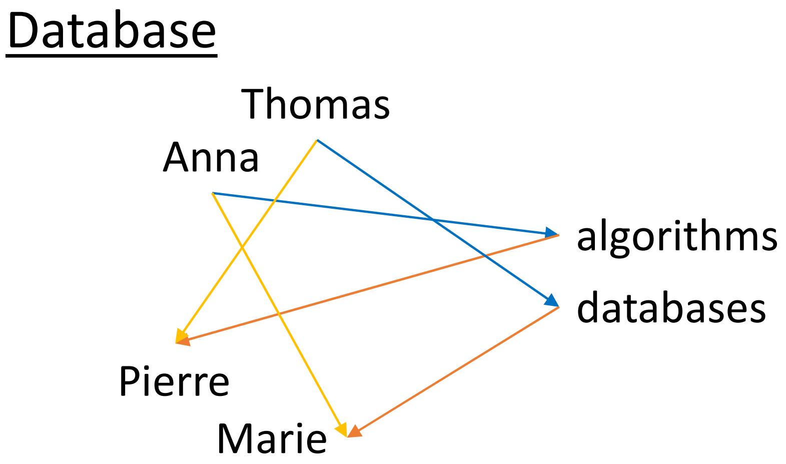
Exam	Professor
algorithms	Pierre
databases	Marie

COI

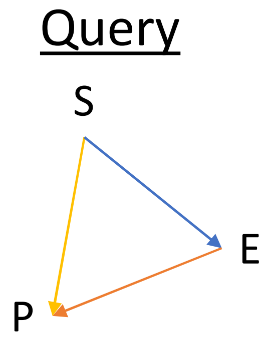
Student	Professor
Thomas	Pierre
Anne	Marie

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

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



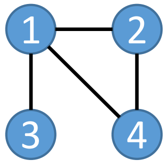
No query answers



Example: Conditional Lower Bound

[Brault-Baron 13]

Assumption: cannot detect triangles in a graph in linear time



edges (a,b) with a<b

$\boldsymbol{Q}$

x	y	z
1	2	4

$\boldsymbol{R_1 = R_2 = R_3}$

1	2
1	3
1	4
2	4

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

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

sHyperclique Hypothesis

- $(k, k-1)$-hyperclique: k vertices, each $k-1$ of them form an edge.

- sHyperclique Hypothesis:
  $\forall k \geq 3$, deciding the existence of a $(k, k-1)$-hyperclique in a hypergraph with m edges cannot be done in time $O(m)$.

- Lemma:
  A cyclic hypergraph contains an induced k-cycle or an induced $(k, k-1)$-hyperclique for some $k \geq 3$.

Dichotomy

[BaganDurandGrandjean; CSL 07]
[Brault-Baron 2013]
[Bringmann, **C**, Mengel; PODS 22]

- Given a join query Q,

If Q is acyclic, Q ∈ Enum<lin,const>

If Q is cyclic, Q ∉ Enum<lin,const>*

* no self-joins, assuming sHyperclique or Zero-Clique

RAM Model Subtleties

- Constant time in the RAM model, what does it mean?
- Assumptions:
  - Length of registers: $\theta(\log n)$
  - Basic operations in $O(1)$
  - Available memory: $O(n^c)$ / $O(n)$
  - Modified memory: everything / $O(n)$
  - Modified memory during enumeration: everything / … / $O(1)$
- Implications:
  - Domain values $\leq n^c$
  - Sorting the input in $O(n)$
    - Radix Sort handles k integers, each bounded by n^c, in time $O(ck + cn)$
  - If $O(n^c)$ available memory,
    - Lookup table with k elements: construction in $O(k)$, search in $O(1)$

"saves" log factors

n = size of input database

RAM Model Subtleties

- Constant time in the RAM model, what does it mean?
- Assumptions:
  - Length of registers: $\theta(\log n)$
  - Basic operations in $O(1)$
  - Available memory: $\boldsymbol{O(n^c)}$ / $O(n)$
  - Modified memory: **everything** / $O(n)$
  - Modified memory during enumeration: **everything** / … / $O(1)$
- Implications:
  - Domain values $\leq n^c$
  - Sorting the input in $O(n)$
    - Radix Sort handles k integers, each bounded by n^c, in time $O(ck + cn)$
  - If $O(n^c)$ available memory,
    - Lookup table with k elements: construction in $O(k)$, search in $O(1)$

"saves" log factors

- **In this talk, assume the relaxed model**

n = size of input database

Plan

- Introduction
- **Join queries**
  - Basic dichotomy
  - **Self-joins**
- Higher expressivity
  - Conjunctive queries
  - Unions of conjunctive queries
- Other Evaluation Tasks

Dichotomy

[BaganDurandGrandjean; CSL 07]
[Brault-Baron 2013]
[Bringmann, **C**, Mengel; PODS 22]

- Given a join query Q,

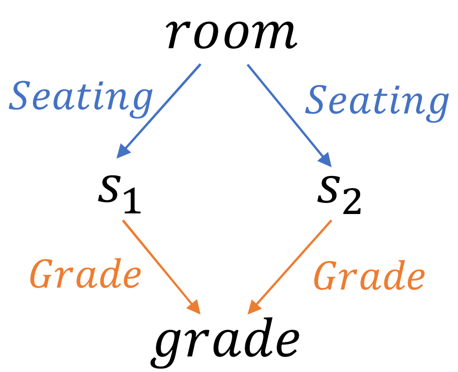
If Q is acyclic, $Q \in$ Enum<lin,const>

If Q is cyclic, $Q \notin$ Enum<lin,const>*

* no self-joins, assuming sHyperclique or Zero-Clique

Example 1

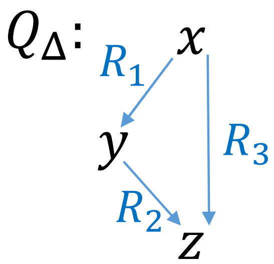
$Q(s_1, s_2, room, grade) \leftarrow$
$Seating(room, s_1), Seating(room, s_2), Grade(s_1, grade), Grade(s_2, grade)$

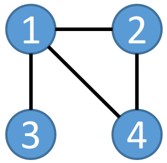


Lower Bound: Cyclic Joins

[Brault-Baron 13]

Assumption: cannot detect triangles in a graph in linear time





edges (a,b) with a<b

$\boldsymbol{Q}$

x	y	z
1	2	4

$\boldsymbol{R_1 = R_2 = R_3}$

1	2
1	3
1	4
2	4

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

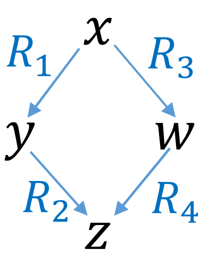
first answer in linear time $\Longrightarrow$ triangle in linear time $\Longrightarrow$ not possible

Lower Bound: Cyclic Joins

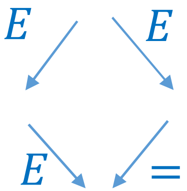
[Brault-Baron 13]

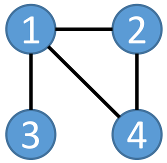
Assumption: cannot detect triangles in a graph in linear time

Q_1:



Construction:





edges (a,b) with a<b

with self-joins, cannot assign different relations to different atoms

$\boldsymbol{Q}$

x	y	z	w
1	2	4	4

$\boldsymbol{R_1 = R_2 = R_3}$

1	2
1	3
1	4
2	4

$\boldsymbol{R_4}$

1	1
2	2
3	3
4	4

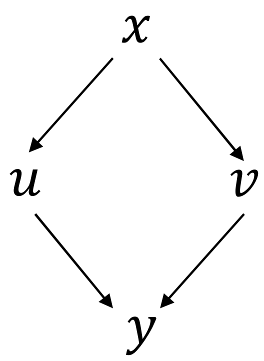
Cyclic: $Q_1(x,y,z,w) \leftarrow R_1(x,y), R_2(y,z), R_3(x,w), \overset{R_3}{\cancel{R_4}}(w,z)$

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

Algorithm [**C**, Segoufin; PODS 23]

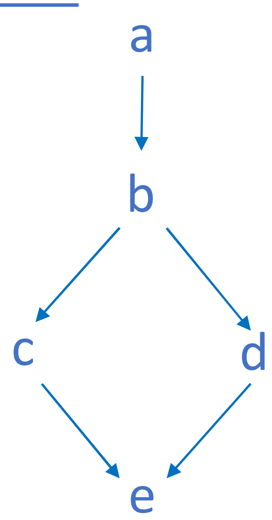
Query

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



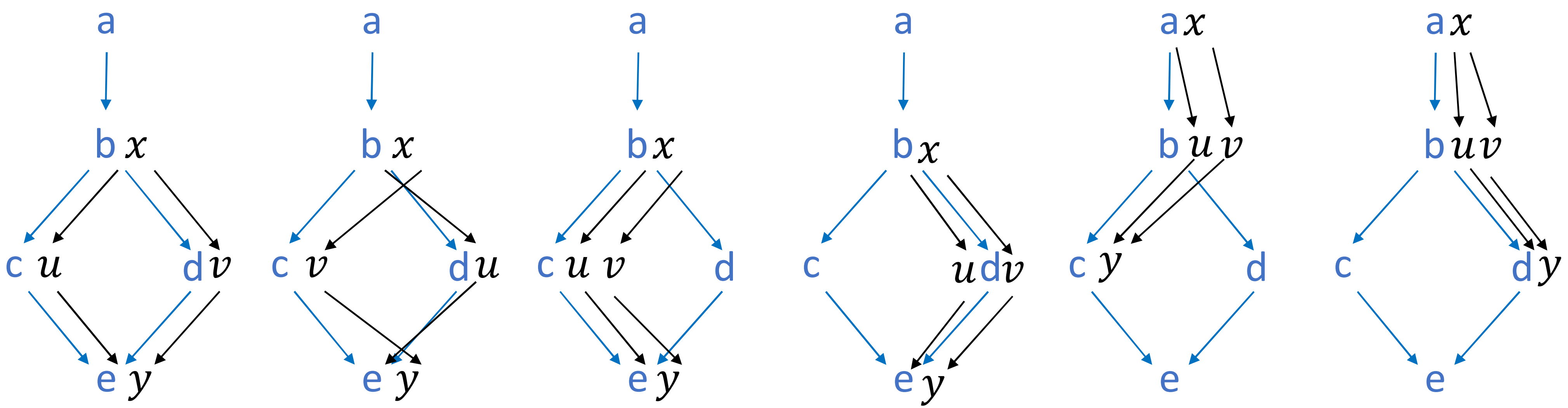
Database

R	
a	b
b	c
b	d
c	e
d	e



Answers

Algorithm [**C**, Segoufin; PODS 23]



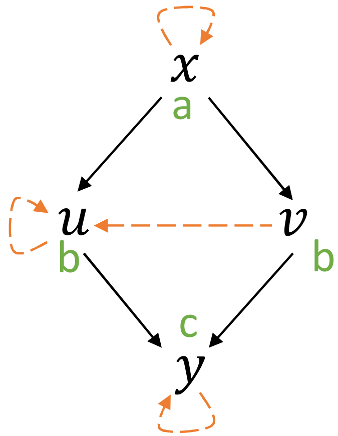
Algorithm [C, Segoufin; PODS 23]

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)

Query

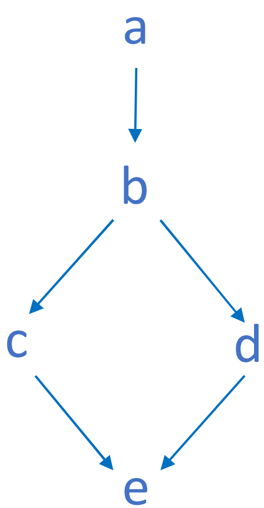
$$Q(x, u, v, y) \leftarrow R(x, u), R(u, y), R(x, v), R(v, y)$$



endomorphism

Database

R	
a	b
b	c
b	d
c	e
d	e



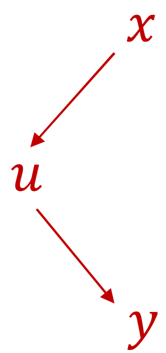


Image I

Answers

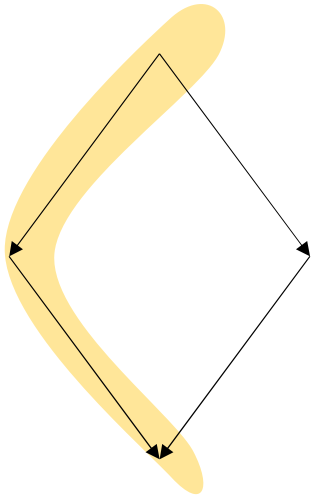
I answers

x	u	y
a	b	c
a	b	d
b	c	e
b	d	e

Q answers

x	u	v	y
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

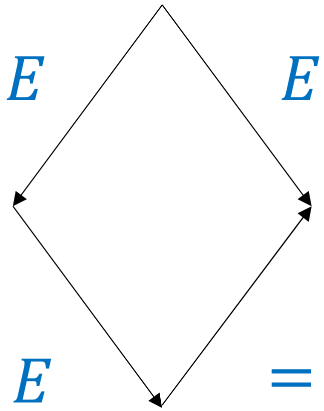


∈ Enum<lin,const>

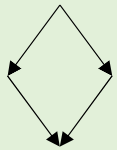
∉ Enum<lin,const> *

* assuming sTriangle

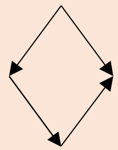
Examples: Full CQs



∈ Enum<lin,const>

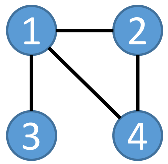


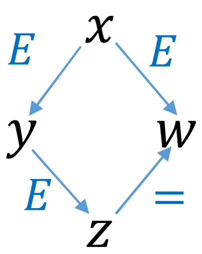
∉ Enum<lin,const> *



* assuming sTriangle

Hardness Proof





$R(x, y) \leftarrow E$

1	2
1	3
1	4
2	4

$R(y, z) \leftarrow E$

1	2
1	3
1	4
2	4

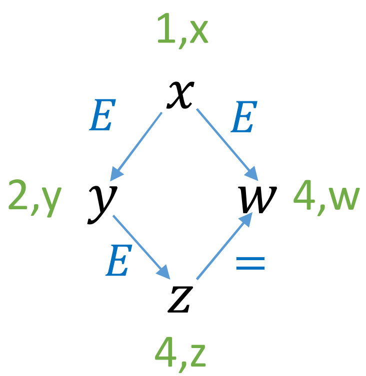
$R(x, w) \leftarrow E$

1	2
1	3
1	4
2	4

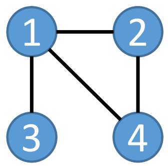
$R(w, z) \leftarrow =$

1	1
2	2
3	3
4	4

Hardness Proof (tagging technique)



Works because Q is a core!



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

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

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

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

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

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

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

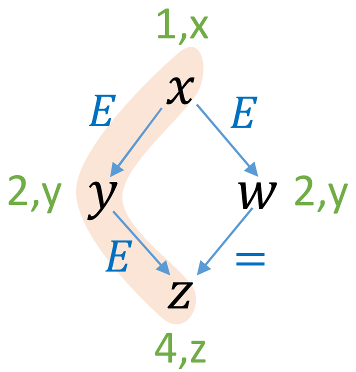
1,w	1,z
2,w	2,z
3,w	3,z
4,w	4,z

union
$\Longrightarrow$

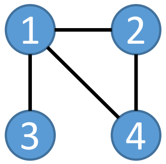
$\boldsymbol{R}$

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



Finds triangles and 2-paths



$R(x, y) \leftarrow E$

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

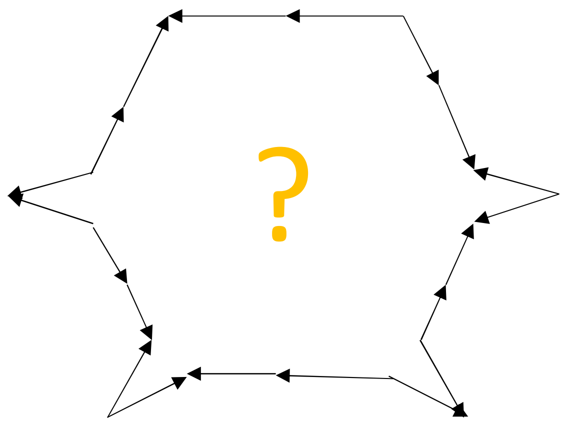
union $\Longrightarrow$

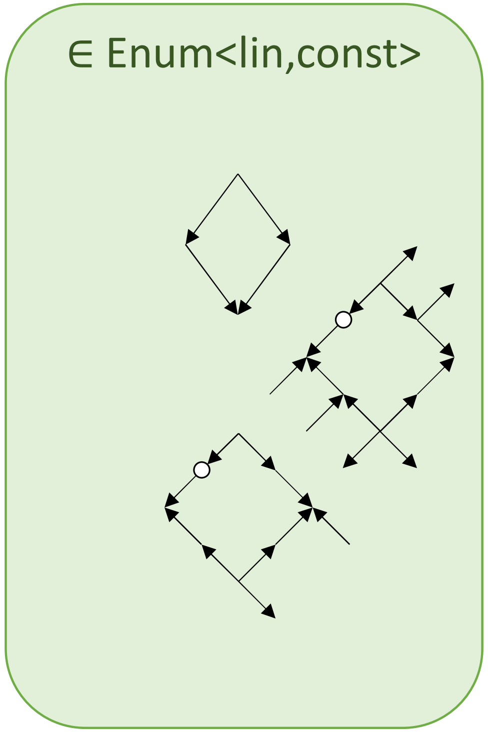
R

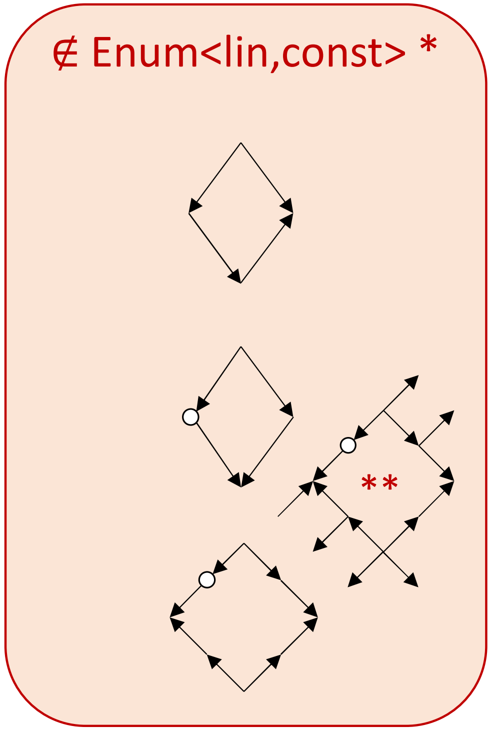
1,x	2,y
1,x	3,y
1,x	4,y
2,x	4,y
1,y	2,z
1,y	3,z
1,y	4,z
2,y	4,z
1,x	2,w
1,x	3,w
1,x	4,w
2,x	4,w
1,w	1,z
…	…

Examples: Full CQs

[**C**, Segoufin; PODS 2023]







* assuming sTriangle

** assuming VUTD

Plan

- Introduction
- Join queries
  - Basic dichotomy
  - Self-joins
- **Higher expressivity**
  - **Conjunctive queries**
  - Unions of conjunctive queries
- Other Evaluation Tasks

Example: Query

Problem

Description	Room
Moisture	5/129
Broken ceiling	Cafeteria
Missing board	5/127

Office

Room	Person	Phone
5/127	Nofar	9590
5/127	Nofar	9591
5/128	Florent	6548
5/128	Guillaume	6548
5/129	David	7544
5/129	Akira	7544

Contact

Person	Email
Nofar	nc@lirmm.fr
Florent	ft@lirmm.fr
Guillaume	gpk@lirmm.fr
David	dc@lirmm.fr

Conjunctive query $\{(E, P, R, D, \cancel{N}) \mid \underbrace{\exists N}\ (D, R) \in \text{Problem}, (R, P, N) \in \text{Office}, (P, E) \in \text{Contact}\}$

~~Join query~~: $Q(E, P, R, D, \cancel{N}) \leftarrow \text{Problem}(D, R), \text{Office}(R, P, N), \text{Contact}(P, E)$

Email	Person	Room	Description	~~Phone~~
nc@lirmm.fr	Nofar	5/127	Missing board	~~9590~~
~~nc@lirmm.fr~~	~~Nofar~~	~~5/127~~	~~Missing board~~	~~9591~~
dc@lirmm.fr	David	5/129	Moisture	~~7544~~

Dichotomy for CQs

[Bagan, Durand, Grandjean; CSL 07]
[Brault-Baron 2013]

- Given a conjunctive query Q,

If Q is acyclic free-connex, $Q \in \mathrm{Enum}\langle\mathrm{lin},\mathrm{const}\rangle$

If Q is acyclic not free-connex, $Q \notin \mathrm{Enum}\langle\mathrm{lin},\mathrm{const}\rangle$*

If Q is cyclic, $Q \notin \mathrm{Enum}\langle\mathrm{lin},\mathrm{const}\rangle$**

* no self-joins, assuming sBMM

** no self-joins, assuming sHyperclique

Lower Bound: acyclic non-free-connex

[Bagan, Durand, Grandjean; CSL 07]

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

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

Indices of ones

$\boldsymbol{R_1}$

R	C
1	1
1	2
2	2

$\boldsymbol{R_2}$

R	C
1	2
2	2

$\boldsymbol{Q}$

R	C
1	2
2	2

Acyclic non-free-connex: $Q(x,z) \leftarrow R_1(x,y), R_2(y,z)$

$O(n^2)$ preprocessing + $O(1)$ delay = $O(n^2)$ total $\implies$ not possible

Plan

- Introduction
- Join queries
  - Basic dichotomy
  - Self-joins
- **Higher expressivity**
  - Conjunctive queries
  - **Unions of conjunctive queries**
- Other Evaluation Tasks

Example: Union of CQs

Posts

Amazing vacation	Alice
Amazing vacation	Bob
Angry post	Bob

Followers

Alice	Bob
Bob	Carol

Friends

Bob	Carol
Carol	Dafni

$$Q_1(post, p2, p3) \leftarrow Posts(post, p1), Followers\ (p1, p2), Friends\ (p2, p3)$$

$$\cup$$

$$Q_2(post, p1, p2) \leftarrow Posts(post, p1), Followers\ (p1, p2)$$

Post	Person 1	Person 2	
Amazing vacation	Bob	Carol	due to Q_1 or Q_2
Amazing vacation	Alice	Bob	due to Q_1
Angry post	Carol	Dafni	due to Q_2
Angry post	Bob	Carol	due to Q_1

Cases for UCQs

All CQs are Easy

always easy

Some Easy, Some Hard

All CQs are Hard

Cases for UCQs

[**C**, Kröll; PODS 19]

Lower Bound: acyclic non-free-connex

[Bagan, Durand, Grandjean; CSL 07]

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

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

Indices of ones

$\boldsymbol{R_1}$

R	C
1	1
1	2
2	2

$\boldsymbol{R_2}$

R	C
1	2
2	2

$\boldsymbol{Q}$

R	C
1	2
2	2

Acyclic non-free-connex: $Q(x, z) \leftarrow R_1(x, y), R_2(y, z)$

$O(n^2)$ preprocessing + $O(1)$ delay = $O(n^2)$ total $\implies$ not possible

Why this isn't hard

not free connex

hard part

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

$$\cup$$

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

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

Example: Tractable Union

acyclic non free-connex

hard part

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

Body-homomorphism

$\cup$

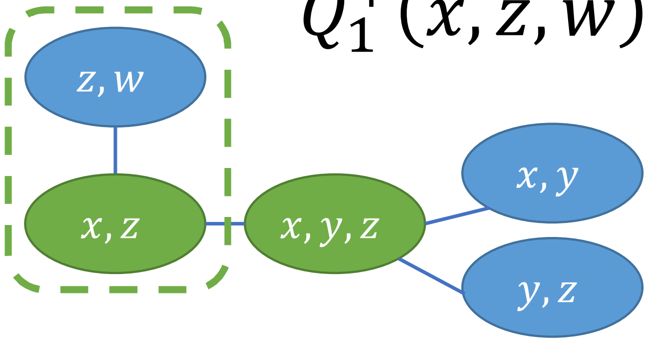
free-connex

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

$\in$ Enum<lin,const>

free-connex

$$Q_1^+(x,z,w) \leftarrow R_1(x,y), R_2(y,z), R_3(z,w), Q_2(x,y,z)$$



	Step	Output	Side Effect
1	Solve Q_2	Q_2	Find $R_1 \bowtie R_2$
2	Solve Q_1^+	Q_1	

Cheater's Lemma

[C, Kröll; PODS 19]

If an enumeration problem can be solved with:

- Usually constant delay
- Almost no duplicates

constant number of linear delay steps

constant number of duplicates per answer

Then*, it is ∈ Enum<lin,const>

Can be solved in: linear preprocessing, constant delay, no duplicates

* using polynomial space

Complexity Measures

[**C**, Kröll; TODS 21]

- (Instance-optimal) linear total time
  - Total time $O(n+N)$

- Linear partial time
  - Time before the ith answer is $O(n+i)$

- Linear preprocessing and constant delay
  - Time before the first answer $O(n)$
  - Time between successive answers $O(1)$

equivalent
assuming we can use polynomial space

n = input size, N = output size

Cases for UCQs

[**C**, Kröll; PODS 19]

Hard ∪ Hard = Easy

[**C**, Kröll; PODS 19]

- 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

Body-homomorphisms

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

Dichotomy for Unions of 2 CQs

[**C**, Bringmann; LMCS 25]

- Given a union of two conjunctive queries Q,

If Q has an acyclic free-connex union extension,
Q ∈ Enum<lin,const>

Otherwise, Q ∉ Enum<lin,const>*

* no self-joins, assuming VUTD

There exists a family of UCQs with no free-connex union extensions s.t.
VUTD hypothesis holds ⇔ no query of the family is in Enum<lin,const>

Example: Intractable Union (Assuming VUTD)

[**C**, Bringmann; LMCS 25]

acyclic non free-connex

hard part

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

$$\cup$$

free-connex

Body-homomorphism

$$Q_2(x, y, w) \leftarrow R_1(x, t_1), R_2(t_2, y), R_3(w, t_3)$$

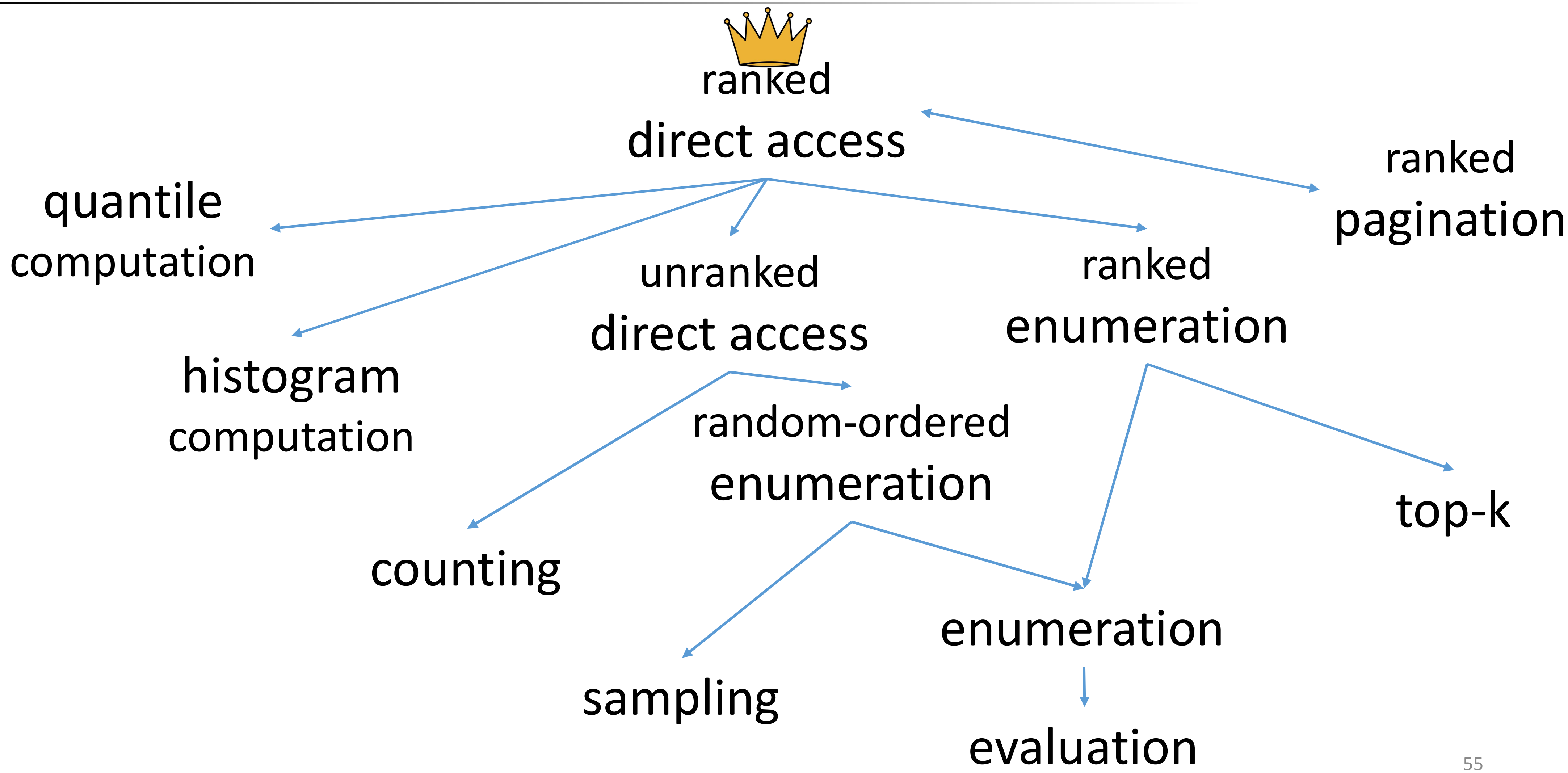
VUTD (Vertex-Unbalanced Triangle Detection) :
$\forall \alpha \in (0,1]$ the existence of a triangle in a tripartite graph
with $|V_1| = n$ and $|V_2| = |V_3| = \Theta(n^{\alpha})$ cannot be decided in time $O(n^{1+\alpha})$

- Q_2 can't help Q_1 : it doesn't provide z
- Construction: assigns large vertex set to z, small vertex sets to x and y, constant $\perp$ to w
- Answers:
  - Ignore answers to Q_2 (there are $O(n^{2\alpha})$ such answers)
  - Check whether answers to Q_1 form an edge (if so, triangle detected)

Plan

- Introduction
- Join queries
  - Basic dichotomy
  - Self-joins
- Higher expressivity
  - Conjunctive queries
  - Unions of conjunctive queries
- **Other Evaluation Tasks**

Overview of Tasks



Conclusion

- **Summary**
  - Introduction (RAM model)
  - Basic dichotomy (reduction from sHyperclique)
  - Self-joins (tagging technique)
  - Conjunctive queries (reduction from matrix multiplication)
  - Unions of conjunctive queries (cheater's lemma)
- **Future**
  - Dichotomy with self-joins
  - Dichotomy for unions of conjunctive queries
  - Connections to VUTD
  - Beyond linear preprocessing time
- **More details**
  - Survey: Enumeration Theory through the Lens of Database Challenges (section 5). PODS 26. Capelli, Carmeli, Conte, Kimelfeld, Pichler, and Tziavelis.
  - Website: https://nofar.carme.li/