Technion

TU WIEN

Humboldt-Universität Informatik

# Complexity of Answering Unions of Conjunctive Queries

**Nofar Carmeli**

Joint work with Christoph Berkholz, Benny Kimelfeld,
Markus Kröll, Nicole Schweikardt, and Shai Zeevi

# Recent Work on DB Enumeration Fine-Grained Complexity

- Query Evaluation incorporating updates
  - [Berkholz,Keppeler,Schweikardt ICDT18]
- Query Evaluation using integrity constraints
  - [C,Kröll ICDT18]
- Query Evaluation using sparsity
  - [Schweikardt,Segoufin,Vigny PODS18]
- Query Evaluation over graphs and strings
  - [Amarilli,Bourhis,Mengel,Niewerth PODS19]
- Query Evaluation over extractions from text
  - [Florenzano,Riveros,Ugarte,Vansummeren,Vrgoc PODS18]

# Goal

CQs
UCQs

# Relational DBs and UCQs

researchers :

| Name | Affiliation |
|---|---|
| Karl Bringmann | Max Planck Institute |
| Seth Pettie | University of Michigan |
| Barna Saha | UC Berkeley |
| … | |

attendance:

| Person | Workshop |
|---|---|
| Daniel Soudry | Learning Theory |
| Karl Bringmann | Fine-grained Complexity |
| Vinod Vaikuntanathan | Cryptography |
| … | |

| Institution | Workshop |
|---|---|
| Technion | Learning Theory |
| Max Planck Institute | Fine-grained Complexity |
| … | |

$$Q(\mathrm{y},\mathrm{z}) \leftarrow \mathrm{researchers}(\mathrm{x},\mathrm{y}), \mathrm{attendance}(\mathrm{x},\mathrm{z})$$

$$\{(\mathrm{y},\mathrm{z}) | \exists \mathrm{z}\colon (\mathrm{x},\mathrm{y}) \in \mathrm{researchers}, (\mathrm{x},\mathrm{z}) \in \mathrm{attendance}\}$$

- CQs: Conjunctive Queries
- UCQs: Unions of CQs
  - Equivalent to positive relational algebra
- The lower bounds assume no self-joins

# Complexity of Queries

researchers :

| Name | Affiliation |
|---|---|
| Karl Bringmann | Max Planck Institute |
| Seth Pettie | University of Michigan |
| Barna Saha | UC Berkeley |
| … | |

attendance:

| Person | Workshop |
|---|---|
| Daniel Soudry | Learning Theory |
| Karl Bringmann | Fine-grained Complexity |
| Vinod Vaikuntanathan | Cryptography |
| … | |

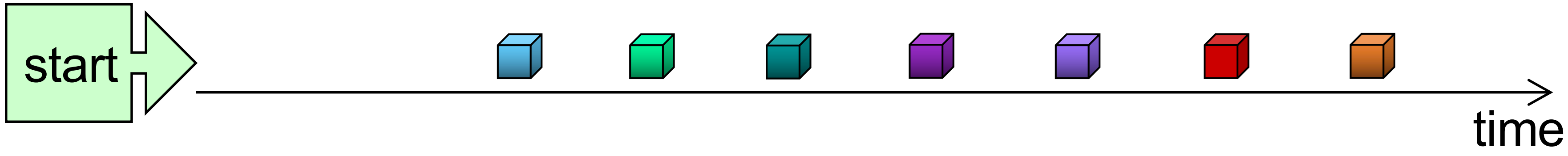
| Institution | Workshop |
|---|---|
| Technion | Learning Theory |
| Max Planck Institute | Fine-grained Complexity |
| … | |

$$Q(\mathrm{x},\mathrm{z}) \leftarrow \mathrm{researchers}(\mathrm{x},\mathrm{y}),\mathrm{attendance}(\mathrm{x},\mathrm{z})$$

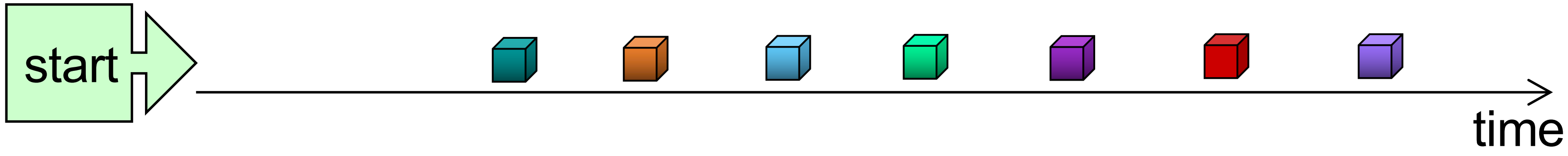
- Treat every query as a problem
  - Input: DB instance
  - Query size: constant
- Using the RAM model

# Goals

## Enumeration



## Random Permutation



# Idea: Separate the Task

- Find the number N of answers

8

- Find a random permutation of 1,…,N

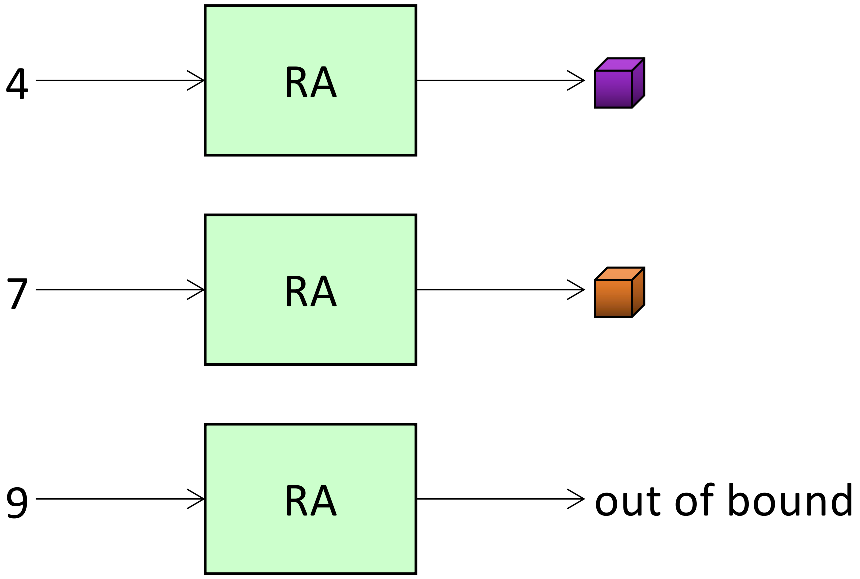
3 7 1 2 4 6 5 8

- Random access to answers

# Definitions

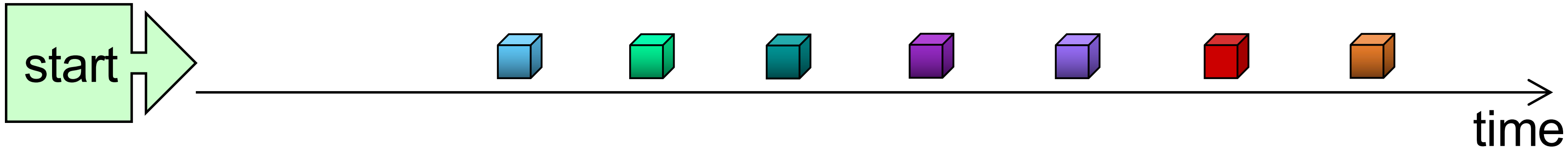
## Random Access

- Given i, returns the ith answer or “out of bound”.
- No constraints on the ordering used

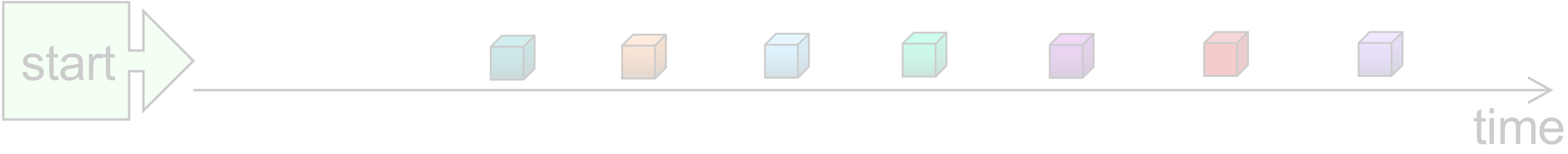


# Goals

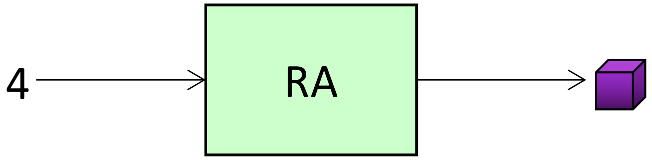
## Enumeration



## Random Permutation



## Random Access



Goal

# CQs

UCQs

# CQs Dichotomy

After linear preprocessing

| Acyclic<br>Free Connex | Acyclic<br>Not Free Connex | Cyclic |
|---|---|---|
| random access<br>$O(\log n)$ | random access<br>~~$O(\log n)$~~ | random access<br>~~$O(\log n)$~~ |
| enumeration<br>$O(1)$ delay | enumeration<br>~~$O(1)$ delay~~ | enumeration<br>~~$O(1)$ delay~~ |
| random permutation<br>$O(\log n)$ delay | random permutation<br>~~$O(\log n)$ delay~~ | random permutation<br>~~$O(\log n)$ delay~~ |
| Also efficient counting, membership testing, etc. | Assuming the hardness of Boolean matrix multiplication | Cannot find any answer in $O(n)$ time<br>Assuming the hardness of finding hypercliques |

# Definitions

An acyclic CQ has a graph with:

A free-connex CQ also requires:

1. a node for every atom
   possibly also subsets

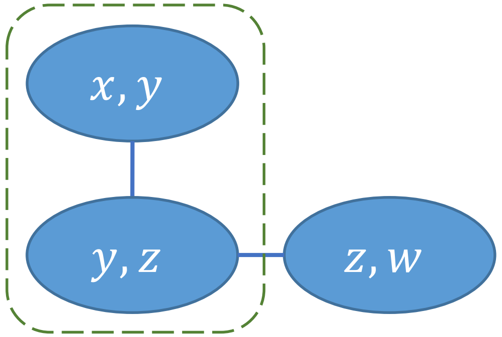
2. tree

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

free – connex

acyclic

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



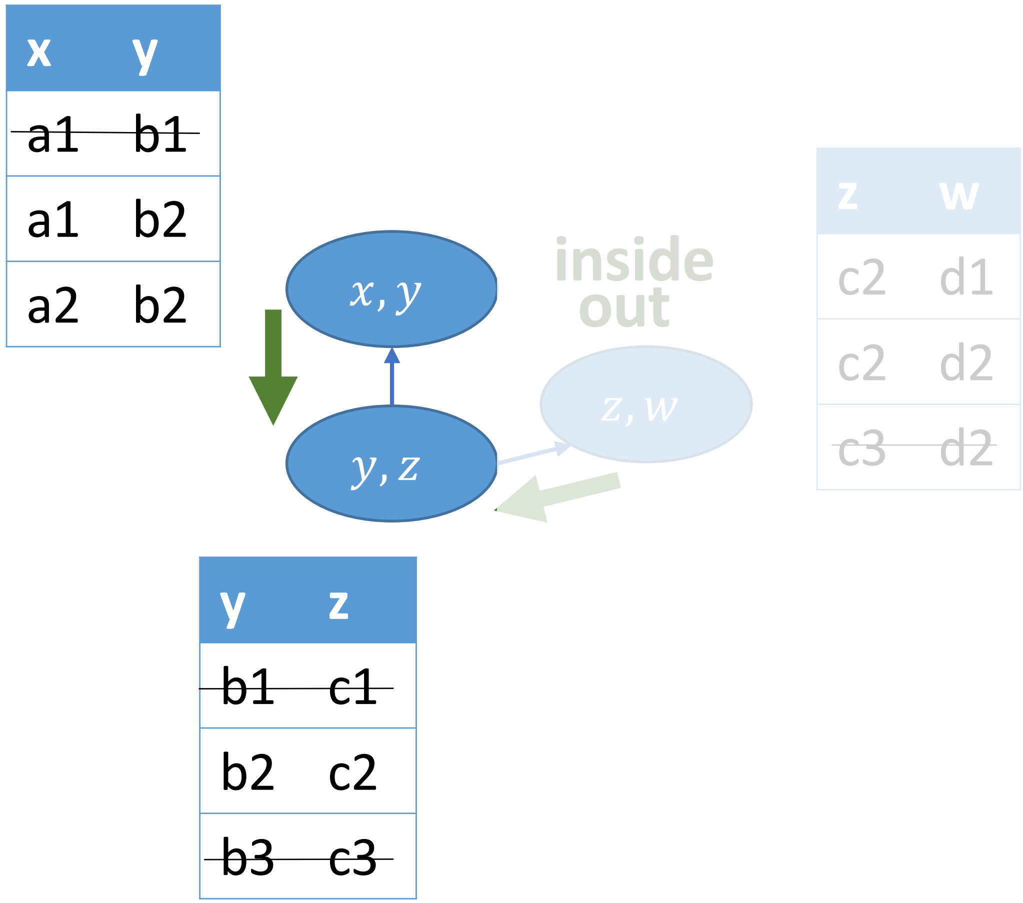
4. a subtree with exactly the free variables

# Free-Connex CQs

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

Can be answered efficiently

1. Find a join tree
2. Remove dangling tuples [Yannakakis81]
3. Ignore existential variables
4. Join



| x | y |
|---|---|
| ~~a1~~ | ~~b1~~ |
| a1 | b2 |
| a2 | b2 |

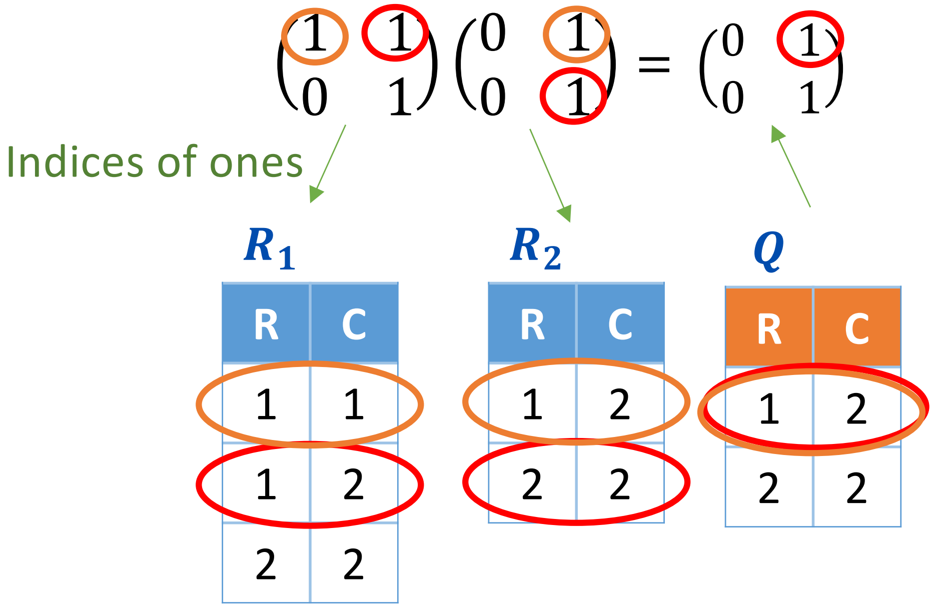
| z | w |
|---|---|
| c2 | d1 |
| c2 | d2 |
| ~~c3~~ | ~~d2~~ |

| y | z |
|---|---|
| ~~b1~~ | ~~c1~~ |
| b2 | c2 |
| ~~b3~~ | ~~c3~~ |

# Acyclic non-free-connex CQs [BaganDurandGrandjean CSL'2007]

Assumption: Boolean matrices cannot be multiplied in time $O(m^{1+o(1)})$

$m$ = number of ones in the input and output



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

Indices of ones

$R_1$

| R | C |
|---|---|
| 1 | 1 |
| 1 | 2 |
| 2 | 2 |

$R_2$

| R | C |
|---|---|
| 1 | 2 |
| 2 | 2 |

$Q$

| R | C |
|---|---|
| 1 | 2 |
| 2 | 2 |

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

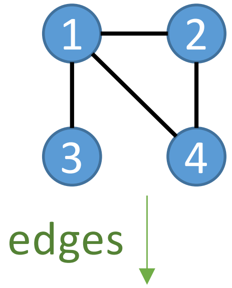
$O(m)$ preprocessing + $O(\log(m))$ delay = $O(m\log(m))$ total $\implies$ not possible

# Cyclic CQs

[Brault-Baron 2013]

Assumption: $k$-Hypercliques cannot be found in time $O(m)$

$m$ = number of edges of size $k-1$

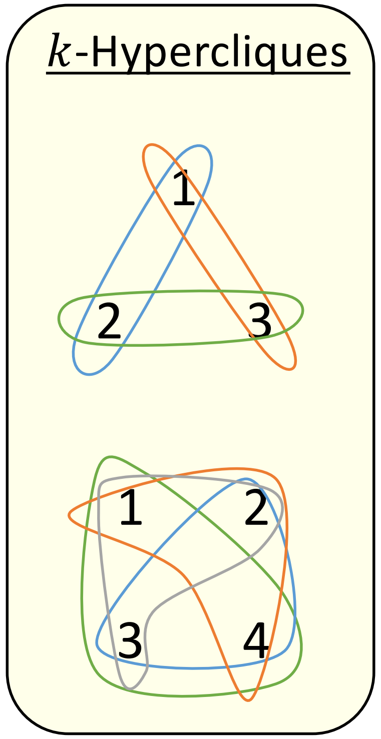


$\boldsymbol{Q}$

| $x$ | $y$ | $z$ |
|---|---|---|
| 1 | 2 | 4 |

$\boldsymbol{R_1 = R_2 = R_3}$

| | |
|---|---|
| 1 | 2 |
| 1 | 3 |
| 1 | 4 |
| 2 | 4 |



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

first answer in $O(m)$ time $\implies$ not possible

# CQs Dichotomy

After linear preprocessing

| Acyclic<br>Free Connex | Acyclic<br>Not Free Connex | Cyclic |
|---|---|---|
| random access<br>$O(\log n)$ | random access<br>~~$O(\log n)$~~ | random access<br>~~$O(\log n)$~~ |
| enumeration<br>$O(1)$ delay | enumeration<br>~~$O(1)$ delay~~ | enumeration<br>~~$O(1)$ delay~~ |
| random permutation<br>$O(\log n)$ delay | random permutation<br>~~$O(\log n)$ delay~~ | random permutation<br>~~$O(\log n)$ delay~~ |
| Also efficient counting, membership testing, etc. | Assuming the hardness of Boolean matrix multiplication | Cannot find any answer in $O(n)$ time<br>Assuming the hardness of finding hypercliques |

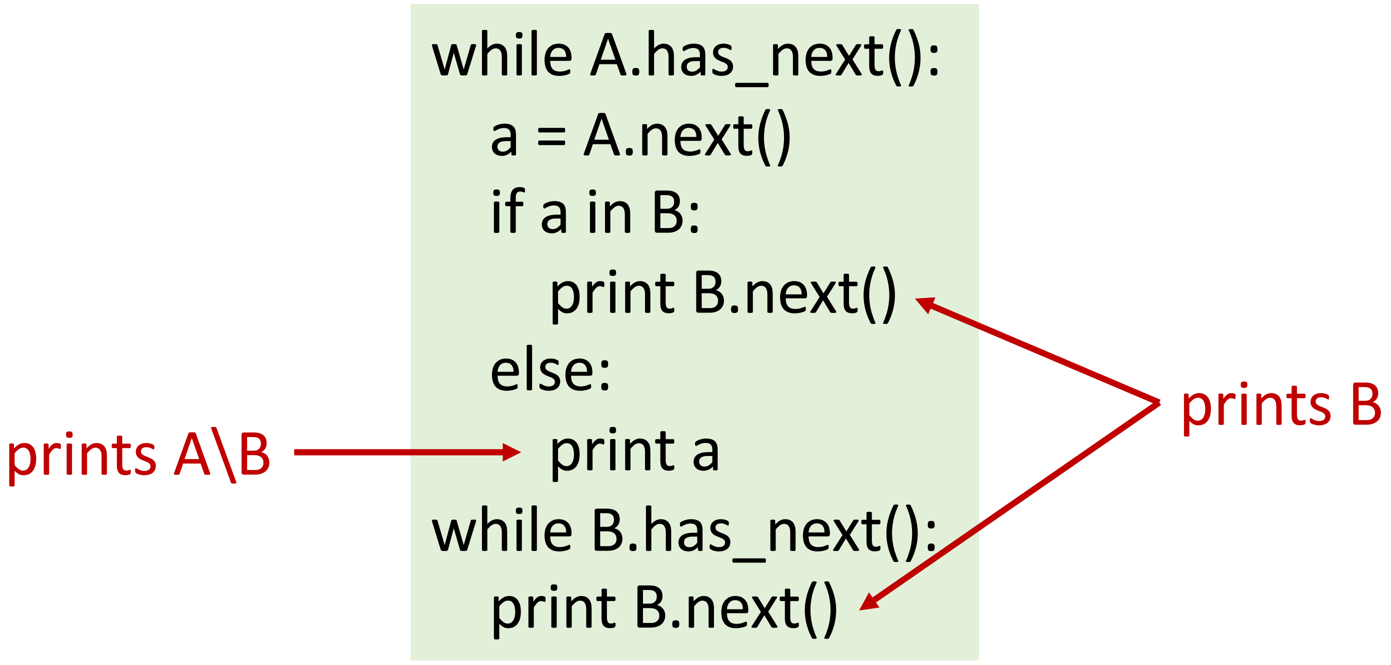
Goal

CQs

# UCQs

# Enumeration: Easy ∪ Easy = Easy

[DurandStrozecki CSL’2011]



```
while A.has_next():
    a = A.next()
    if a in B:
        print B.next()
    else:
        print a
while B.has_next():
    print B.next()
```

prints A\B

prints B

A\B and B are a partition of A∪B

# Access: Easy ∪ Easy = Sometimes Hard

Proof (Example):

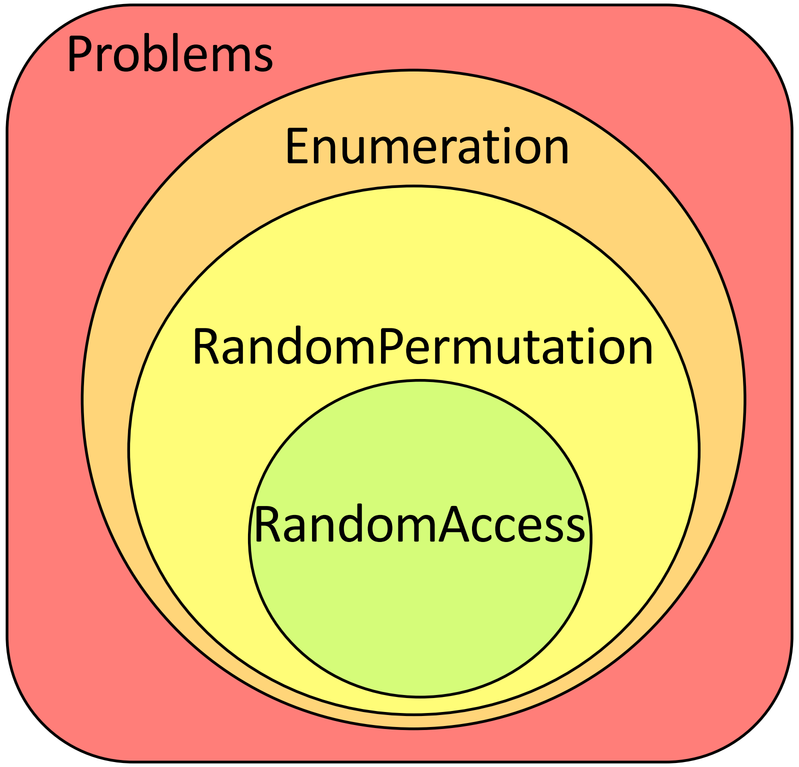
- $Q_1(x,y,z) \leftarrow R(x,y), S(y,z)$ free-connex
- $Q_2(x,y,z) \leftarrow S(y,z), T(x,z)$ free-connex
- $Q_1 \cap Q_2(x,y,z) \leftarrow R(x,y), S(y,z), T(x,z)$ cyclic
  - Cannot determine whether $|Q_1 \cap Q_2| > 0$ in linear time
    assumption: cannot find a triangle in a graph in linear time.

- Assume by contradiction $Q_1 \cup Q_2 \in$ RandomAccess
  - Ask for answer number $|Q_1| + |Q_2|$
  - This checks if $|Q_1 \cup Q_2| < |Q_1| + |Q_2|$ in linear time
  - This determines whether $|Q_1 \cap Q_2| = |Q_1| + |Q_2| - |Q_1 \cup Q_2| > 0$

Contradiction!

# Open Problem!

- How hard is random permutation?
- First Step:
  Can this example be solved in log delay?



Example:

$$Q_1(x,y,z) \leftarrow R(x,y), S(y,z)$$
$$Q_2(x,y,z) \leftarrow S(y,z), T(x,z)$$

Answer $Q_1 \cup Q_2$

# Unions with Hard CQs

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

non free − connex

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

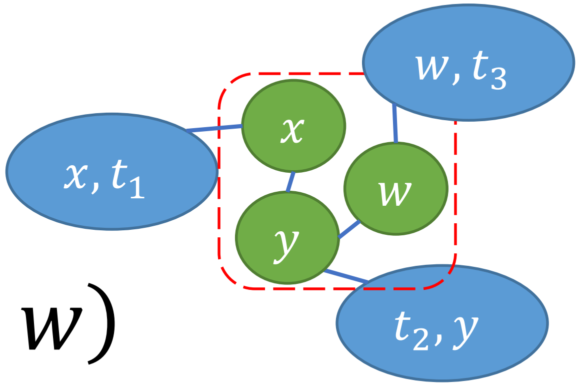
free − connex

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

- Previous claim:
  - Non-redundant unions with a hard CQ are always hard
- We show:
  - They are sometimes easy
  - Even if they contain **only** hard CQs
- Example: $Q_1(x,z,w,u), Q_2(u,z,y,x) \leftarrow$
  $R_1(x,y), R_2(y,z), R_3(z,w), R_4(w,u)$

# Open Problem!

- What characterizes easy to enumerate UCQs?
- First Step:
  What is the complexity for the examples?
  [Carmeli, Kröll: *On the Enumeration Complexity of Unions of Conjunctive Queries*. PODS 2019]



$$Q_1(x, y, w) \leftarrow R_1(x, z), R_2(z, y), R_3(y, w)$$
$$Q_2(x, y, w) \leftarrow R_1(x, t_1), R_2(t_2, y), R_3(w, t_3)$$

Goal
CQs
UCQs

# Thank You.