Uniform Reliability for Unbounded Homomorphism-Closed Graph Queries

Antoine Amarilli

March 29, 2023

Télécom Paris

Uncertain data management

In this talk, we manage data represented as a labeled graph

Uncertain data management

In this talk, we manage **data** represented as a **labeled graph**

WorksAt	
Antoine	Télécom Paris
Antoine	Paris Sud
Benny	Paris Sud
Benny	Technion
İsmail	U. Oxford

Uncertain data management

In this talk, we manage data represented as a labeled graph

WorksAt	
Antoine	Télécom Paris
Antoine	Paris Sud
Benny	Paris Sud
Benny	Technion
İsmail	U. Oxford

MemberOf	
Télécom Paris	ParisTech
ParisTech	IP Paris
IP Paris	Télécom Paris
Paris Sud	IP Paris
Paris Sud	Paris-Saclay
Technion	CESAER

Uncertain data management

In this talk, we manage **data** represented as a **labeled graph**

WorksAt	
Antoine	Télécom Paris
Antoine	Paris Sud
Benny	Paris Sud
Benny	Technion
İsmail	U. Oxford

MemberOf	
Télécom Paris	ParisTech
ParisTech	IP Paris
IP Paris	Télécom Paris
Paris Sud	IP Paris
Paris Sud	Paris-Saclay
Technion	CESAER

A.
Télécom Paris
ParisTech
Paris Sud
IP Paris
B.
Technion
Paris-Saclay
İ.
U. Oxford
CESAER

Uncertain data management

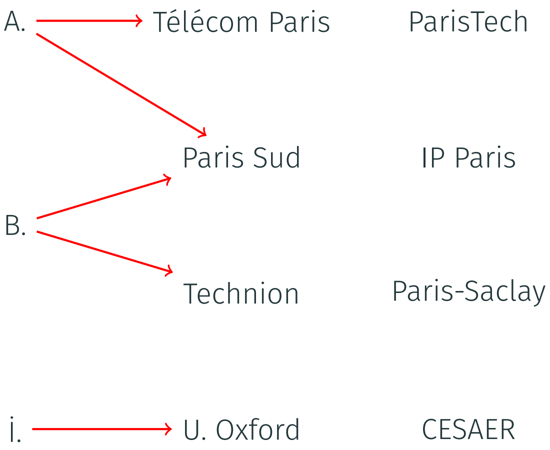
In this talk, we manage **data** represented as a **labeled graph**

WorksAt

Antoine	Télécom Paris
Antoine	Paris Sud
Benny	Paris Sud
Benny	Technion
İsmail	U. Oxford

MemberOf

Télécom Paris	ParisTech
ParisTech	IP Paris
IP Paris	Télécom Paris
Paris Sud	IP Paris
Paris Sud	Paris-Saclay
Technion	CESAER

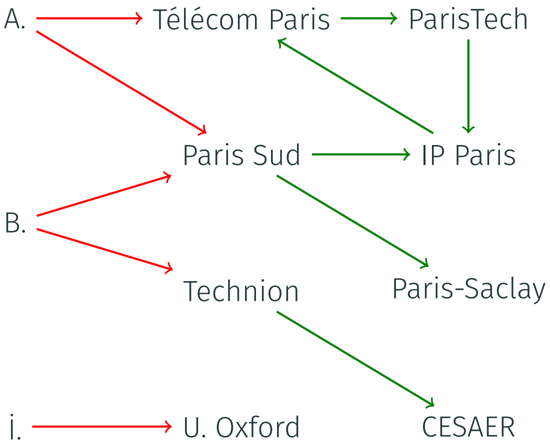


Uncertain data management

In this talk, we manage **data** represented as a **labeled graph**

WorksAt	
Antoine	Télécom Paris
Antoine	Paris Sud
Benny	Paris Sud
Benny	Technion
İsmail	U. Oxford

MemberOf	
Télécom Paris	ParisTech
ParisTech	IP Paris
IP Paris	Télécom Paris
Paris Sud	IP Paris
Paris Sud	Paris-Saclay
Technion	CESAER



Uniform Reliability

We study the uniform reliability problem:

- Fix a query Q

Uniform Reliability

We study the uniform reliability problem:

- Fix a query Q
- Input: a graph database I

Uniform Reliability

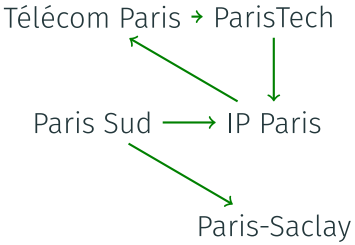
We study the uniform reliability problem:

- Fix a query Q
- Input: a graph database I
- Output: how many subgraphs of I satisfy Q

Uniform Reliability

We study the **uniform reliability** problem:

- **Fix** a query Q
- **Input:** a graph database I
- **Output:** how many **subgraphs** of I satisfy Q

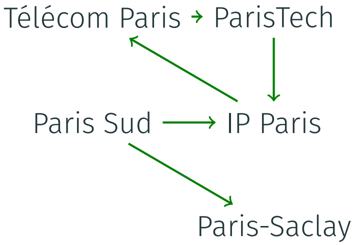


Example query: “is there a cycle of the **MemberOf** relation?

Uniform Reliability

We study the **uniform reliability** problem:

- **Fix** a query Q
- **Input:** a graph database I
- **Output:** how many **subgraphs** of I satisfy Q



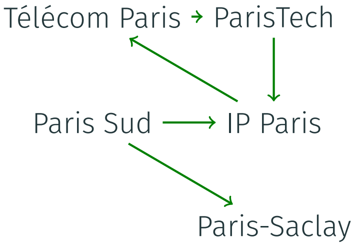
Example query: "is there a cycle of the **MemberOf** relation?

Of the 32 possible subgraphs...

Uniform Reliability

We study the **uniform reliability** problem:

- **Fix** a query Q
- **Input:** a graph database I
- **Output:** how many **subgraphs** of I satisfy Q



Example query: “is there a cycle of the **MemberOf** relation?

Of the 32 possible subgraphs…

… there are 4 that satisfy the query

Motivation and connections

- We think that uniform reliability is a natural problem

Motivation and connections

- We think that uniform reliability is a natural problem
- Connections to computational social choice:
    - the Shapley value [Livshits et al., 2020]
    - the causal effect [Salimi, 2016]

Motivation and connections

- We think that uniform reliability is a natural problem
- Connections to computational social choice:
  - the Shapley value [Livshits et al., 2020]
  - the causal effect [Salimi, 2016]
- Restricted case of probabilistic query evaluation on tuple-independent databases
  $\rightarrow$ All facts have probability $1/2$

Motivation and connections

- We think that uniform reliability is a natural problem

- Connections to computational social choice:
    - the Shapley value [Livshits et al., 2020]
    - the causal effect [Salimi, 2016]

- Restricted case of probabilistic query evaluation on tuple-independent databases

  $\rightarrow$ All facts have probability $1/2$

- Generalization of (two-terminal unweighted directed) network reliability [Valiant, 1979, Provan and Ball, 1983]:
    - Input: a directed graph with a source s and sink t
    - Output: the probability that there is a path from s to t if each edge can fail (independently) with probability $1/2$

Queries

- **Query:** maps a labeled graph to YES/NO

Queries

- **Query:** maps a labeled graph to YES/NO
- **Conjunctive query** (CQ): can I find a match of a **pattern**? e.g., $x \longrightarrow y \longrightarrow z$

Queries

- Query: maps a labeled graph to YES/NO
- Conjunctive query (CQ): can I find a match of a pattern? e.g., $x \longrightarrow y \longrightarrow z$
  $\rightarrow$ We want a homomorphism from the pattern to the graph (not necessarily injective)

Queries

- **Query:** maps a labeled graph to YES/NO
- **Conjunctive query** (CQ): can I find a match of a **pattern**? e.g., $x \longrightarrow y \longrightarrow z$
  $\rightarrow$ We want a **homomorphism** from the pattern to the graph (not necessarily **injective**)
- **Union of conjunctive queries** (UCQ): can I find a match of **some pattern**?

Queries

- **Query:** maps a labeled graph to YES/NO
- **Conjunctive query** (CQ): can I find a match of a **pattern**? e.g., $x \longrightarrow y \longrightarrow z$
  - $\rightarrow$ We want a **homomorphism** from the pattern to the graph (not necessarily **injective**)
- **Union of conjunctive queries** (UCQ): can I find a match of **some pattern**?

$\rightarrow$ **Homomorphism-closed queries** (UCQ$^\infty$):
if I satisfies Q and I has a homomorphism to I' then I' also satisfies Q

Queries

- **Query:** maps a labeled graph to YES/NO
- **Conjunctive query** (CQ): can I find a match of a **pattern**? e.g., $x \longrightarrow y \longrightarrow z$
  $\rightarrow$ We want a **homomorphism** from the pattern to the graph (not necessarily **injective**)
- **Union of conjunctive queries** (UCQ): can I find a match of **some pattern**?

$\rightarrow$ **Homomorphism-closed queries** (UCQ$^\infty$):
if I satisfies Q and I has a homomorphism to I' then I' also satisfies Q

Intuition about homomorphism-closed queries:

Queries

- **Query:** maps a labeled graph to YES/NO
- **Conjunctive query** (CQ): can I find a match of a **pattern**? e.g., $x \longrightarrow y \longrightarrow z$
  $\rightarrow$ We want a **homomorphism** from the pattern to the graph (not necessarily **injective**)
- **Union of conjunctive queries** (UCQ): can I find a match of **some pattern**?

$\rightarrow$ **Homomorphism-closed queries** (UCQ$^\infty$):
if I satisfies Q and I has a homomorphism to I' then I' also satisfies Q

Intuition about homomorphism-closed queries:

- Generalize **CQs** and **UCQs**, but also **regular path queries** (RPQs), **Datalog**, **reliability queries** (source-to-sink path), existence of a **cycle**, etc.

Queries

- **Query:** maps a labeled graph to YES/NO
- **Conjunctive query** (CQ): can I find a match of a **pattern**? e.g., $x \longrightarrow y \longrightarrow z$
  $\rightarrow$ We want a **homomorphism** from the pattern to the graph (not necessarily **injective**)
- **Union of conjunctive queries** (UCQ): can I find a match of **some pattern**?

$\rightarrow$ **Homomorphism-closed queries** (UCQ$^\infty$):
if I satisfies Q and I has a homomorphism to I' then I' also satisfies Q

Intuition about homomorphism-closed queries:

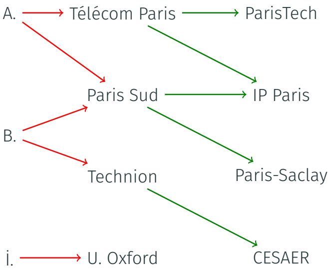
- Generalize **CQs** and **UCQs**, but also **regular path queries** (RPQs), **Datalog**, **reliability queries** (source-to-sink path), existence of a **cycle**, etc.
- Not supported: **inequalities**, **negation**

Problem statement: Uniform reliability (UR)

- We fix a homomorphism-closed query Q
  For instance the CQ: $x \longrightarrow y \longrightarrow z$

Problem statement: Uniform reliability (UR)

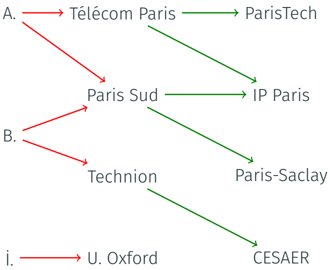
- We fix a homomorphism-closed query Q
  For instance the CQ: $x \longrightarrow y \longrightarrow z$
- The input is a labeled graph I:



Problem statement: Uniform reliability (UR)

- We **fix** a homomorphism-closed query Q
  For instance the CQ: $x \longrightarrow y \longrightarrow z$

- The **input** is a labeled graph I:

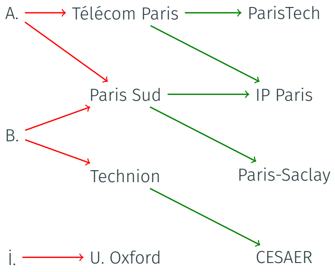


- The **output** is the **number** of subgraphs of I satisfying Q

Problem statement: Uniform reliability (UR)

- We **fix** a homomorphism-closed query Q
  For instance the CQ: $x \longrightarrow y \longrightarrow z$

- The **input** is a labeled graph I:



- The **output** is the **number** of subgraphs of I satisfying Q

$\rightarrow$ What is the **complexity** of the problem $\mathrm{UR}(Q)$, depending on the query Q?

Results on Uniform Reliability

Known results: SJFCQs

A **self-join-free CQ** (SJFCQ) is a CQ with no **repeated relations**,
i.e., **all colors are different**:

- E.g., $x \longrightarrow y \longrightarrow z$ but not $x \longrightarrow y \longrightarrow z \longleftarrow w$

Known results: SJFCQs

A **self-join-free CQ** (SJFCQ) is a CQ with no **repeated relations**,
i.e., **all colors are different**:

- E.g., $\begin{tikzcd} x \arrow[r] \arrow[rr, bend left] & y \arrow[r] & z \end{tikzcd}$ but not $x \longrightarrow y \longrightarrow z \longleftarrow w$

The following **dichotomy** is known for a class of **hierarchical SJFCQs**
for the **probabilistic query evaluation** problem (PQE) on tuple-independent databases:

Theorem [Dalvi and Suciu, 2007]

- *For any **hierarchical** SJFCQ $\mathbf{Q}$, the problem PQE$(\mathbf{Q})$ is in **PTIME***
- *For any **non-hierarchical** SJFCQ $\mathbf{Q}$, the problem PQE$(\mathbf{Q})$ is **#P-hard***

Known results: SJFCQs

A **self-join-free CQ** (SJFCQ) is a CQ with no **repeated relations**,
i.e., **all colors are different**:

- E.g., $\begin{tikzcd} x \arrow[r] \arrow[rr, bend left] & y \arrow[r] & z \end{tikzcd}$ but not $x \longrightarrow y \longrightarrow z \longleftarrow w$

The following **dichotomy** is known for a class of **hierarchical SJFCQs**
for the **probabilistic query evaluation** problem (PQE) on tuple-independent databases:

Theorem [Dalvi and Suciu, 2007]

- *For any **hierarchical** SJFCQ $\mathbf{Q}$, the problem PQE$(\mathbf{Q})$ is in **PTIME***
- *For any **non-hierarchical** SJFCQ $\mathbf{Q}$, the problem PQE$(\mathbf{Q})$ is **#P-hard***

Theorem [A. and Kimelfeld, 2022]

*The **same dichotomy** holds for the UR problem*

Known results: UCQs

For UCQs, a dichotomy on PQE is also known for a class of **safe queries**:

Theorem [Dalvi and Suciu, 2012]

- *For any **safe** UCQ Q, the problem PQE(Q) is in **PTIME***
- *For any **unsafe** UCQ Q, the problem PQE(Q) is **#P-hard***

Known results: UCQs

For UCQs, a dichotomy on PQE is also known for a class of **safe queries**:

Theorem [Dalvi and Suciu, 2012]

- *For any **safe** UCQ Q, the problem PQE(Q) is in **PTIME***
- *For any **unsafe** UCQ Q, the problem PQE(Q) is **#P-hard***

The **upper bound** for PQE holds in particular for UR, but not the **lower bound**...

Known results: UCQs

For UCQs, a dichotomy on PQE is also known for a class of **safe queries**:

Theorem [Dalvi and Suciu, 2012]

- *For any **safe** UCQ Q, the problem PQE(Q) is in **PTIME***
- *For any **unsafe** UCQ Q, the problem PQE(Q) is **#P-hard***

The **upper bound** for PQE holds in particular for UR, but not the **lower bound**...

Theorem [Kenig and Suciu, 2021]

- *For any **unsafe** UCQ Q, the **#P-hardness** of PQE holds even when we only use probabilities 0, 1/2, and 1*
- *For **Type-I forbidden queries**, the UR problem is **#P-hard***

Known results: UCQ$^\infty$s

- Some UCQ$^\infty$s are equivalent to UCQs (aka bounded): see previous slide

Known results: UCQ$^\infty$s

- Some UCQ$^\infty$s are **equivalent to UCQs** (aka **bounded**): see previous slide
- For the other UCQ$^\infty$s (**unbounded**), hardness is known for **PQE**

Known results: UCQ$^\infty$s

- Some UCQ$^\infty$s are **equivalent to UCQs** (aka **bounded**): see previous slide
- For the other UCQ$^\infty$s (**unbounded**), hardness is known for **PQE**

Theorem [A. and Ceylan, 2022]

*For any **unbounded UCQ$^\infty$** Q, the problem PQE(Q) is **#P-hard***

Known results: UCQ$^\infty$s

- Some UCQ$^\infty$s are **equivalent to UCQs** (aka **bounded**): see previous slide
- For the other UCQ$^\infty$s (**unbounded**), hardness is known for **PQE**

Theorem [A. and Ceylan, 2022]

*For any **unbounded UCQ$^\infty$** Q, the problem PQE(Q) is **#P-hard***

For UR, the only known results are those on **reliability** (S–T CONNECTEDNESS):

Known results: UCQ$^\infty$s

- Some UCQ$^\infty$s are **equivalent to UCQs** (aka **bounded**): see previous slide
- For the other UCQ$^\infty$s (**unbounded**), hardness is known for **PQE**

Theorem [A. and Ceylan, 2022]

For any ***unbounded UCQ$^\infty$*** Q*, the problem PQE(Q) is* ***#P-hard***

For UR, the only known results are those on **reliability** (S–T CONNECTEDNESS):

Theorem [Valiant, 1979]

The UR problem for the query Q*:* $\longrightarrow (\longrightarrow)^* \longrightarrow$ *is* ***#P-hard***

Summary of known results, and result statement

Query class	UR	PQE

Summary of known results, and result statement

Query class	UR	PQE
Hierarchical SJFCQs	PTIME [DS07]	

Summary of known results, and result statement

<table>
<tr><th>Query class</th><th>UR</th><th>PQE</th></tr>
<tr><td>Hierarchical SJFCQs</td><td colspan="2">PTIME [DS07]</td></tr>
<tr><td>Safe UCQs</td><td colspan="2">PTIME [DS12]</td></tr>
</table>

Summary of known results, and result statement

Query class	UR	PQE
Hierarchical SJFCQs	PTIME [DS07]	
Safe UCQs	PTIME [DS12]	
Non-hierarchical SJFCQs	#P-hard [AK22]	#P-hard [DS07]

Summary of known results, and result statement

Query class	UR	PQE
Hierarchical SJFCQs	PTIME [DS07]	
Safe UCQs	PTIME [DS12]	
Non-hierarchical SJFCQs	#P-hard [AK22]	#P-hard [DS07]
Unsafe UCQs	some are #P-hard [KS21]	#P-hard [DS12]

Summary of known results, and result statement

Query class	UR	PQE
Hierarchical SJFCQs	PTIME [DS07]	
Safe UCQs	PTIME [DS12]	
Non-hierarchical SJFCQs	#P-hard [AK22]	#P-hard [DS07]
Unsafe UCQs	some are #P-hard [KS21]	#P-hard [DS12]
Reliability queries	#P-hard [V79]	

Summary of known results, and result statement

<table>
<tr><th>Query class</th><th>UR</th><th>PQE</th></tr>
<tr><td>Hierarchical SJFCQs</td><td colspan="2">PTIME [DS07]</td></tr>
<tr><td>Safe UCQs</td><td colspan="2">PTIME [DS12]</td></tr>
<tr><td>Non-hierarchical SJFCQs</td><td>#P-hard [AK22]</td><td>#P-hard [DS07]</td></tr>
<tr><td>Unsafe UCQs</td><td>some are #P-hard [KS21]</td><td>#P-hard [DS12]</td></tr>
<tr><td>Reliability queries</td><td colspan="2">#P-hard [V79]</td></tr>
<tr><td>Other unbounded UCQ$^{\infty}$s</td><td>???</td><td>#P-hard [AC21]</td></tr>
</table>

Summary of known results, and result statement

<table>
<tr><th>Query class</th><th>UR</th><th>PQE</th></tr>
<tr><td>Hierarchical SJFCQs</td><td colspan="2">PTIME [DS07]</td></tr>
<tr><td>Safe UCQs</td><td colspan="2">PTIME [DS12]</td></tr>
<tr><td>Non-hierarchical SJFCQs</td><td>#P-hard [AK22]</td><td>#P-hard [DS07]</td></tr>
<tr><td>Unsafe UCQs</td><td>some are #P-hard [KS21]</td><td>#P-hard [DS12]</td></tr>
<tr><td>Reliability queries</td><td colspan="2">#P-hard [V79]</td></tr>
<tr><td>Other unbounded UCQ^{∞}s</td><td>#P-hard</td><td>#P-hard [AC21]</td></tr>
</table>

Summary of known results, and result statement

<table>
<tr><th>Query class</th><th>UR</th><th>PQE</th></tr>
<tr><td>Hierarchical SJFCQs</td><td colspan="2">PTIME [DS07]</td></tr>
<tr><td>Safe UCQs</td><td colspan="2">PTIME [DS12]</td></tr>
<tr><td>Non-hierarchical SJFCQs</td><td>#P-hard [AK22]</td><td>#P-hard [DS07]</td></tr>
<tr><td>Unsafe UCQs</td><td>some are #P-hard [KS21]</td><td>#P-hard [DS12]</td></tr>
<tr><td>Reliability queries</td><td colspan="2">#P-hard [V79]</td></tr>
<tr><td>Other unbounded UCQ$^\infty$s</td><td>#P-hard</td><td>#P-hard [AC21]</td></tr>
</table>

Theorem

For any unbounded UCQ$^\infty$ Q on graphs, the uniform reliability problem for Q is #P-hard

Proof Techniques

Starting point: Hardness of a non-hierarchical query

- Let us study uniform reliability for SJFCQ Q : $x \longrightarrow y \longrightarrow z \longrightarrow w$

Starting point: Hardness of a non-hierarchical query

- Let us study uniform reliability for SJFCQ Q : $x \longrightarrow y \longrightarrow z \longrightarrow w$
- Idea: Reduce from the #P-hard problem of counting satisfying valuations of a positive partitioned 2-DNF formula

Starting point: Hardness of a non-hierarchical query

- Let us study uniform reliability for SJFCQ Q : $x \xrightarrow{} y \xrightarrow{} z \xrightarrow{} w$
- Idea: Reduce from the **#P-hard** problem of **counting satisfying valuations** of a **positive partitioned 2-DNF formula**
- From $\phi : (X_1 \wedge Y_1) \vee (X_1 \wedge Y_2) \vee (X_2 \wedge Y_2) \vee (X_3 \wedge Y_1) \vee (X_3 \wedge Y_2)$, build **graph database** I_ϕ:

Starting point: Hardness of a non-hierarchical query

- Let us study uniform reliability for SJFCQ Q : $x \longrightarrow y \longrightarrow z \longrightarrow w$
- Idea: Reduce from the **#P-hard** problem of **counting satisfying valuations** of a **positive partitioned 2-DNF formula**
- From $\phi : (X_1 \wedge Y_1) \vee (X_1 \wedge Y_2) \vee (X_2 \wedge Y_2) \vee (X_3 \wedge Y_1) \vee (X_3 \wedge Y_2)$, build **graph database** I_ϕ:

$$a'_1 \longrightarrow a_1$$

$$a'_2 \longrightarrow a_2$$

$$a'_3 \longrightarrow a_3$$

Starting point: Hardness of a non-hierarchical query

- Let us study uniform reliability for SJFCQ Q : $x \longrightarrow y \longrightarrow z \longrightarrow w$
- Idea: Reduce from the **#P-hard** problem of **counting satisfying valuations** of a **positive partitioned 2-DNF formula**
- From $\phi : (X_1 \wedge Y_1) \vee (X_1 \wedge Y_2) \vee (X_2 \wedge Y_2) \vee (X_3 \wedge Y_1) \vee (X_3 \wedge Y_2)$, build **graph database** I_ϕ:

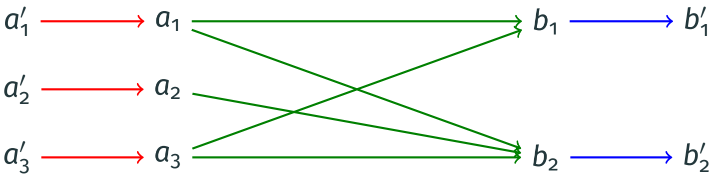
$$a'_1 \longrightarrow a_1 \qquad\qquad b_1 \longrightarrow b'_1$$

$$a'_2 \longrightarrow a_2$$

$$a'_3 \longrightarrow a_3 \qquad\qquad b_2 \longrightarrow b'_2$$

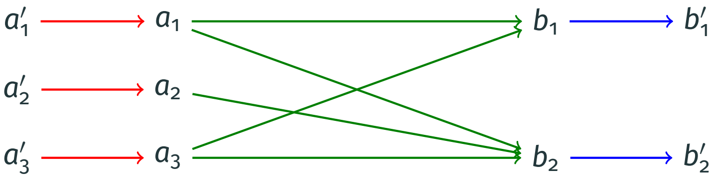
Starting point: Hardness of a non-hierarchical query

- Let us study uniform reliability for SJFCQ Q : $x \xrightarrow{} y \xrightarrow{} z \xrightarrow{} w$
- Idea: Reduce from the **#P-hard** problem of **counting satisfying valuations** of a **positive partitioned 2-DNF formula**
- From $\phi : (X_1 \wedge Y_1) \vee (X_1 \wedge Y_2) \vee (X_2 \wedge Y_2) \vee (X_3 \wedge Y_1) \vee (X_3 \wedge Y_2)$, build **graph database** I_ϕ:



Starting point: Hardness of a non-hierarchical query

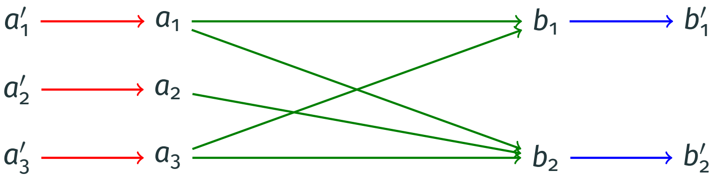
- Let us study uniform reliability for SJFCQ Q : $x \longrightarrow y \longrightarrow z \longrightarrow w$
- Idea: Reduce from the **#P-hard** problem of **counting satisfying valuations** of a **positive partitioned 2-DNF formula**
- From $\phi : (X_1 \wedge Y_1) \vee (X_1 \wedge Y_2) \vee (X_2 \wedge Y_2) \vee (X_3 \wedge Y_1) \vee (X_3 \wedge Y_2)$, build **graph database** I_ϕ:



- If the green edges are **always here**: clear correspondence between satisfying assignments and subsets satisfying Q

Starting point: Hardness of a non-hierarchical query

- Let us study uniform reliability for SJFCQ Q : $x \longrightarrow y \longrightarrow z \longrightarrow w$
- Idea: Reduce from the **#P-hard** problem of **counting satisfying valuations** of a **positive partitioned 2-DNF formula**
- From $\phi : (X_1 \wedge Y_1) \vee (X_1 \wedge Y_2) \vee (X_2 \wedge Y_2) \vee (X_3 \wedge Y_1) \vee (X_3 \wedge Y_2)$, build **graph database** I_ϕ:



- If the green edges are **always here**: clear correspondence between satisfying assignments and subsets satisfying Q
- Otherwise, **#P-hardness** via **interpolation technique** (already in [A., Kimelfeld, 2022])

Extending to an unbounded query

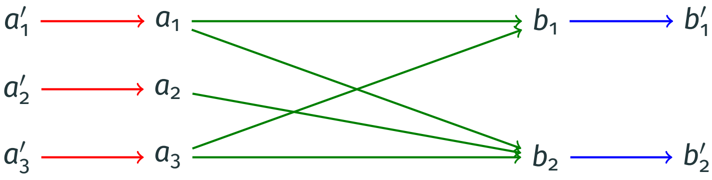
What about the unbounded UCQ$^\infty$ Q: $\color{red}{\longrightarrow}\left(\color{green}{\longrightarrow}\right)^*\color{blue}{\longrightarrow}$

Extending to an unbounded query

What about the unbounded UCQ$^\infty$ Q: $\textcolor{red}{\longrightarrow}\left(\textcolor{green}{\longrightarrow}\right)^{*}\textcolor{blue}{\longrightarrow}$

Same proof!

$$\phi : (X_1 \land Y_1) \lor (X_1 \land Y_2) \lor (X_2 \land Y_2) \lor (X_3 \land Y_1) \lor (X_3 \land Y_2)$$

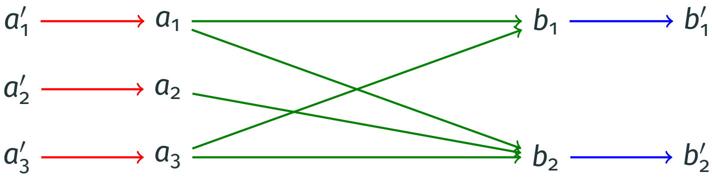


Extending to an unbounded query

What about the unbounded UCQ$^\infty$ Q: $\textcolor{red}{\longrightarrow}\left(\textcolor{green}{\longrightarrow}\right)^*\textcolor{blue}{\longrightarrow}$

Same proof!

$$\phi : (X_1 \wedge Y_1) \vee (X_1 \wedge Y_2) \vee (X_2 \wedge Y_2) \vee (X_3 \wedge Y_1) \vee (X_3 \wedge Y_2)$$



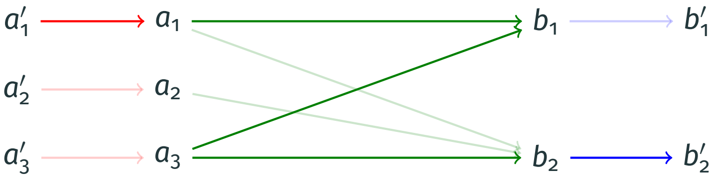
However, **the same would not work** for Q': $\textcolor{red}{\longrightarrow}\left(\textcolor{green}{\longrightarrow}\ \textcolor{green}{\longleftarrow}\right)^*\textcolor{green}{\longrightarrow}\ \textcolor{blue}{\longrightarrow}$

Extending to an unbounded query

What about the unbounded UCQ$^\infty$ Q: $\textcolor{red}{\longrightarrow}\left(\textcolor{green}{\longrightarrow}\right)^*\textcolor{blue}{\longrightarrow}$

Same proof!

$$\phi : (X_1 \wedge Y_1) \vee (X_1 \wedge Y_2) \vee (X_2 \wedge Y_2) \vee (X_3 \wedge Y_1) \vee (X_3 \wedge Y_2)$$



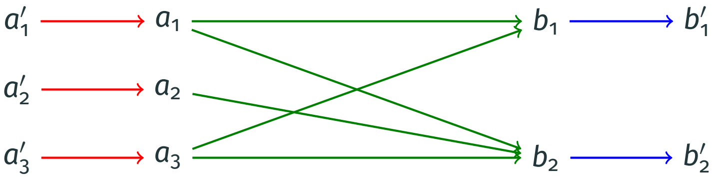
However, the same would not work for Q': $\textcolor{red}{\longrightarrow}\left(\textcolor{green}{\longrightarrow}\ \textcolor{green}{\longleftarrow}\right)^*\textcolor{green}{\longrightarrow}\ \textcolor{blue}{\longrightarrow}$

Extending to an unbounded query

What about the unbounded UCQ$^\infty$ Q: $\longrightarrow \left(\longrightarrow\right)^{*} \longrightarrow$

Same proof!

$$\phi : (X_1 \wedge Y_1) \vee (X_1 \wedge Y_2) \vee (X_2 \wedge Y_2) \vee (X_3 \wedge Y_1) \vee (X_3 \wedge Y_2)$$



However, the same would not work for Q': $\longrightarrow \left(\longrightarrow \ \longleftarrow\right)^{*} \longrightarrow \ \longrightarrow$

$\rightarrow$ We call this an **iterable** unbounded query

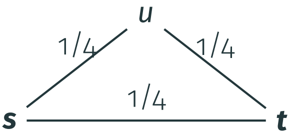
Hardness for iterable unbounded queries

How can we show hardness for $\boldsymbol{Q'}$: $\textcolor{red}{\longrightarrow}\left(\textcolor{green}{\longrightarrow}\ \textcolor{green}{\longleftarrow}\right)^{*}\textcolor{green}{\longrightarrow}\ \textcolor{blue}{\longrightarrow}$

Hardness for iterable unbounded queries

How can we show hardness for Q': $\longrightarrow \left(\longrightarrow \ \longleftarrow \right)^{*} \longrightarrow \ \longrightarrow$

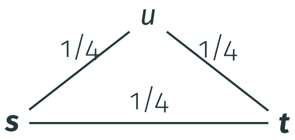
Reduce from the **#P-hard** source-to-target reliability problem on undirected graphs



Hardness for iterable unbounded queries

How can we show hardness for Q': $\color{red}{\longrightarrow}\color{black}{(}\color{green}{\longrightarrow\ \longleftarrow}\color{black}{)^*}\color{green}{\longrightarrow}\ \color{blue}{\longrightarrow}$

Reduce from the #P-hard source-to-target reliability problem on undirected graphs

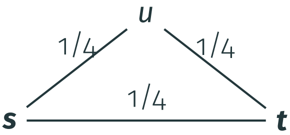


is coded as

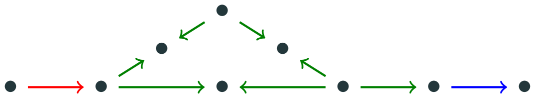
Hardness for iterable unbounded queries

How can we show hardness for Q': $\textcolor{red}{\longrightarrow}(\textcolor{green}{\longrightarrow}\ \textcolor{green}{\longleftarrow})^{*}\textcolor{green}{\longrightarrow}\ \textcolor{blue}{\longrightarrow}$

Reduce from the #P-hard source-to-target reliability problem on undirected graphs



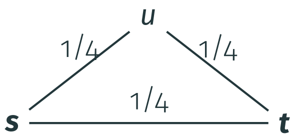
is coded as



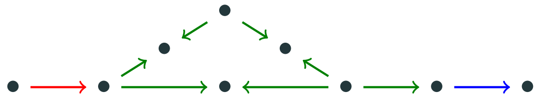
Hardness for iterable unbounded queries

How can we show hardness for Q': $\color{red}{\longrightarrow}\color{black}{(}\color{green}{\longrightarrow\ \longleftarrow}\color{black}{)^*}\color{green}{\longrightarrow}\ \color{blue}{\longrightarrow}$

Reduce from the **#P-hard** source-to-target reliability problem on undirected graphs



is coded as

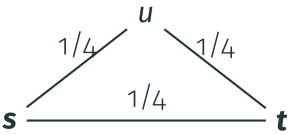


- **Idea:** There is a path connecting s and t in a possible world of the graph at the left iff the query Q' is **satisfied** in the corresponding subgraph of the graph database

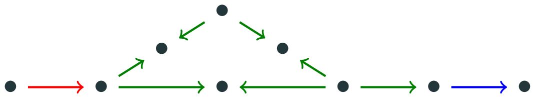
Hardness for iterable unbounded queries

How can we show hardness for Q': $\longrightarrow (\longrightarrow \ \longleftarrow)^* \longrightarrow \ \longrightarrow$

Reduce from the **#P-hard** **source-to-target reliability problem on undirected graphs**



is coded as

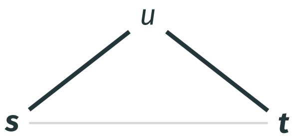


- **Idea:** There is a **path connecting s and t** in a possible world of the graph at the left iff the query Q' is **satisfied** in the corresponding subgraph of the graph database
- Only consider subgraphs where the red and blue edges are present, the others violate Q'

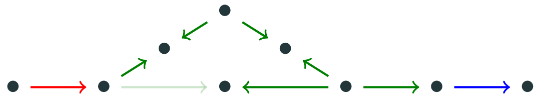
Hardness for iterable unbounded queries

How can we show hardness for Q': $\color{red}{\longrightarrow}\color{black}(\color{green}{\longrightarrow}\ \longleftarrow\color{black})^{*}\color{green}{\longrightarrow}\ \color{blue}{\longrightarrow}$

Reduce from the **#P-hard** **source-to-target reliability problem on undirected graphs**



is coded as

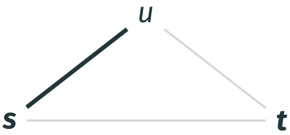


- **Idea:** There is a **path connecting s and t** in a possible world of the graph at the left iff the query Q' is **satisfied** in the corresponding subgraph of the graph database
- Only consider subgraphs where the red and blue edges are present, the others violate Q'

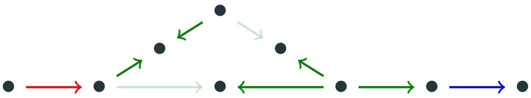
Hardness for iterable unbounded queries

How can we show hardness for Q': $\longrightarrow (\longrightarrow \ \longleftarrow)^* \longrightarrow \ \longrightarrow$

Reduce from the **#P-hard** **source-to-target reliability problem on undirected graphs**



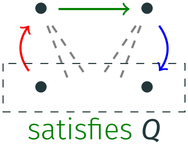
is coded as



- **Idea:** There is a **path connecting s and t** in a possible world of the graph at the left iff the query Q' is **satisfied** in the corresponding subgraph of the graph database
- Only consider subgraphs where the red and blue edges are present, the others violate Q'

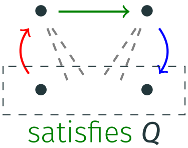
Extending to general unbounded queries

- We show in [A., Ceylan, 2022] that any unbounded UCQ$^\infty$ has a **tight pattern**: a graph with three distinguished edges $\color{red}{\rightarrow}$ $\color{green}{\rightarrow}$ $\color{blue}{\rightarrow}$ such that:



Extending to general unbounded queries

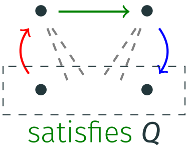
- We show in [A., Ceylan, 2022] that any unbounded UCQ$^{\infty}$ has a **tight pattern**: a graph with three distinguished edges $\textcolor{red}{\rightarrow}$ $\textcolor{green}{\rightarrow}$ $\textcolor{blue}{\rightarrow}$ such that:



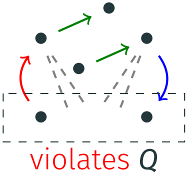
but

Extending to general unbounded queries

- We show in [A., Ceylan, 2022] that any unbounded UCQ$^\infty$ has a **tight pattern**: a graph with three distinguished edges $\color{red}{\rightarrow}$ $\color{green}{\rightarrow}$ $\color{blue}{\rightarrow}$ such that:

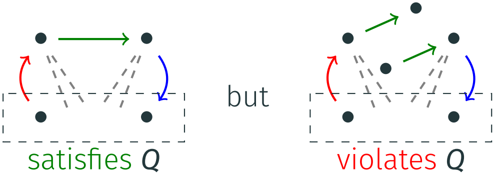


but



Extending to general unbounded queries

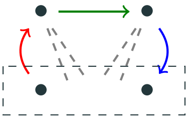
- We show in [A., Ceylan, 2022] that any unbounded UCQ$^\infty$ has a **tight pattern**: a graph with three distinguished edges $\color{red}{\rightarrow}$ $\color{green}{\rightarrow}$ $\color{blue}{\rightarrow}$ such that:



- Use the **first hardness proof** if it is non-iterable, the **second** if it is

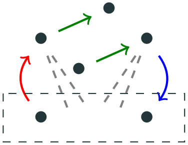
Extending to general unbounded queries

- We show in [A., Ceylan, 2022] that any unbounded UCQ$^\infty$ has a **tight pattern**: a graph with three distinguished edges $\color{red}{\rightarrow}$ $\color{green}{\rightarrow}$ $\color{blue}{\rightarrow}$ such that:

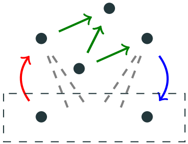


satisfies Q

but



violates Q

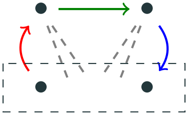


non-iterable case: violates Q
iterable case: satisfies Q

- Use the **first hardness proof** if it is non-iterable, the **second** if it is

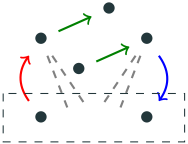
Extending to general unbounded queries

- We show in [A., Ceylan, 2022] that any unbounded UCQ$^\infty$ has a **tight pattern**: a graph with three distinguished edges $\rightarrow$ $\rightarrow$ $\rightarrow$ such that:

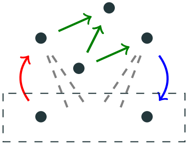


satisfies Q

but



violates Q



non-iterable case: violates Q
iterable case: satisfies Q

- Use the **first hardness proof** if it is non-iterable, the **second** if it is

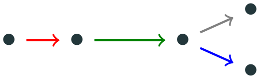
$\rightarrow$ What **breaks down** in the unweighted case?

Uniform reliability: Technical challenges

- We must impose subinstance-minimality: strict subsets of the tight pattern do not satisfy the query

Uniform reliability: Technical challenges

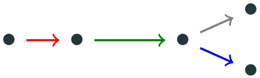
- We must impose **subinstance-minimality**: strict subsets of the tight pattern do not satisfy the query
- We cannot **choose** specific left and right edges! For instance, non-iterable case:



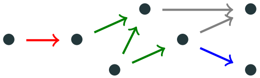
satisfies Q

Uniform reliability: Technical challenges

- We must impose **subinstance-minimality**: strict subsets of the tight pattern do not satisfy the query
- We cannot **choose** specific left and right edges! For instance, non-iterable case:



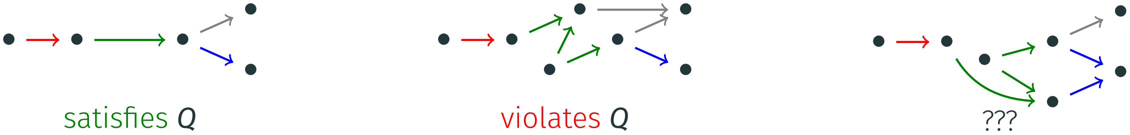
satisfies Q



violates Q

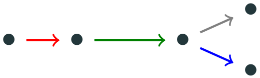
Uniform reliability: Technical challenges

- We must impose **subinstance-minimality**: strict subsets of the tight pattern do not satisfy the query
- We cannot **choose** specific left and right edges! For instance, non-iterable case:

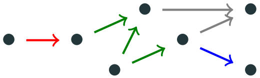


Uniform reliability: Technical challenges

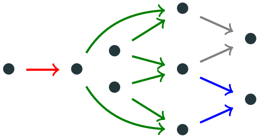
- We must impose subinstance-minimality: strict subsets of the tight pattern do not satisfy the query
- We cannot choose specific left and right edges! For instance, non-iterable case:



satisfies Q



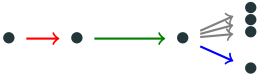
violates Q



???

Tool: The saturation technique

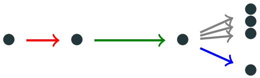
Idea: make the presence of facts **very certain** by copying them enough times:



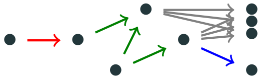
satisfies Q

Tool: The saturation technique

Idea: make the presence of facts **very certain** by copying them enough times:



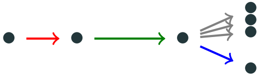
satisfies Q



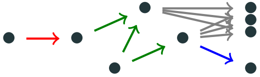
violates Q

Tool: The saturation technique

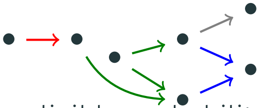
Idea: make the presence of facts **very certain** by copying them enough times:



satisfies Q



violates Q



negligible probability

But serious **limitations**:

- Only applicable in the **non-iterable** case
  (query matches must have constant size to make the probability negligible)
- Only works with **binary facts**!

Overall proof strategy

- Choose the tight pattern **very carefully**:

Overall proof strategy

- Choose the tight pattern very carefully:
  - Subinstance-minimality

Overall proof strategy

- Choose the tight pattern **very carefully**:
  - Subinstance-minimality
  - Minimal number of facts in the middle (green) edge

Overall proof strategy

- Choose the tight pattern **very carefully**:
  - Subinstance-minimality
  - Minimal number of facts in the middle (green) edge
  - Minimal number of non-green left and right edges (extra facts)

Overall proof strategy

- Choose the tight pattern **very carefully**:
  - Subinstance-minimality
  - Minimal number of facts in the middle (green) edge
  - Minimal number of non-green left and right edges (extra facts)
  - Minimal number of green left and right edges (copy facts)

Overall proof strategy

- Choose the tight pattern **very carefully**:
  - Subinstance-minimality
  - Minimal number of facts in the middle (green) edge
  - Minimal number of non-green left and right edges (extra facts)
  - Minimal number of green left and right edges (copy facts)
- Non-iterable case:
  - Use **saturation technique** on copy facts
  - Argue that extra facts are **necessary**

Overall proof strategy

- Choose the tight pattern **very carefully**:
  - Subinstance-minimality
  - Minimal number of facts in the middle (green) edge
  - Minimal number of non-green left and right edges (extra facts)
  - Minimal number of green left and right edges (copy facts)
- Non-iterable case:
  - Use **saturation technique** on copy facts
  - Argue that extra facts are **necessary**
- Iterable case:
  - Show that there are **only copy facts**
  - Minimize their number **lexicographically** (left, then right)
  - Tweak coding to make all copy facts **necessary**

Conclusion and Future Work

Conclusion and future work

Theorem

For any unbounded UCQ^{∞} Q on graphs, uniform reliability for Q is #P-hard

Conclusion and future work

Theorem

For any unbounded UCQ^∞ Q on graphs, uniform reliability for Q is #P-hard

<table>
<tr><th>Query class</th><th>UR</th><th>PQE</th></tr>
<tr><td>Hierarchical SJFCQs</td><td colspan="2">PTIME [DS07]</td></tr>
<tr><td>Safe UCQs</td><td colspan="2">PTIME [DS12]</td></tr>
<tr><td>Non-hierarchical SJFCQs</td><td>#P-hard [AK22]</td><td>#P-hard [DS07]</td></tr>
<tr><td>Unsafe UCQs</td><td>some are #P-hard [KS21]</td><td>#P-hard [DS12]</td></tr>
<tr><td>Reliability queries</td><td colspan="2">#P-hard [V79]</td></tr>
<tr><td>Other unbounded UCQ^∞s</td><td>#P-hard</td><td>#P-hard [AC21]</td></tr>
</table>

Conclusion and future work

Theorem

For any unbounded UCQ^{∞} Q on graphs, uniform reliability for Q is #P-hard

Query class	UR	PQE
Hierarchical SJFCQs	PTIME [DS07]	
Safe UCQs	PTIME [DS12]	
Non-hierarchical SJFCQs	#P-hard [AK22]	#P-hard [DS07]
Unsafe UCQs	some are #P-hard [KS21]	#P-hard [DS12]
Reliability queries	#P-hard [V79]	
Other unbounded UCQ^{∞}s	#P-hard	#P-hard [AC21]

- Does intractability hold for **all unsafe UCQs**? (left open by [KS21], looks challenging)

Conclusion and future work

Theorem

For any unbounded UCQ^{∞} Q on graphs, uniform reliability for Q is #P-hard

Query class	UR	PQE
Hierarchical SJFCQs	PTIME [DS07]	
Safe UCQs	PTIME [DS12]	
Non-hierarchical SJFCQs	#P-hard [AK22]	#P-hard [DS07]
Unsafe UCQs	some are #P-hard [KS21]	#P-hard [DS12]
Reliability queries	#P-hard [V79]	
Other unbounded UCQ^{∞}s	#P-hard	#P-hard [AC21]

- Does intractability hold for **all unsafe UCQs**? (left open by [KS21], looks challenging)
- Does intractability for **unbounded UCQ^{∞}** extend to higher arity? (plausible, but technical)

Conclusion and future work

Theorem

For any unbounded UCQ^∞ Q on graphs, uniform reliability for Q is #P-hard

Query class	UR	PQE
Hierarchical SJFCQs	PTIME [DS07]	
Safe UCQs	PTIME [DS12]	
Non-hierarchical SJFCQs	#P-hard [AK22]	#P-hard [DS07]
Unsafe UCQs	some are #P-hard [KS21]	#P-hard [DS12]
Reliability queries	#P-hard [V79]	
Other unbounded UCQ^∞s	#P-hard	#P-hard [AC21]

- Does intractability hold for **all unsafe UCQs**? (left open by [KS21], looks challenging)
- Does intractability for **unbounded UCQ^∞** extend to higher arity? (plausible, but technical)

Thanks for your attention!

References i

Amarilli, A. and Ceylan, I. I. (2022).
The dichotomy of evaluating homomorphism-closed queries on probabilistic graphs.
LMCS.

Amarilli, A. and Kimelfeld, B. (2022).
Uniform Reliability of Self-Join-Free Conjunctive Queries.
LMCS.

Dalvi, N. and Suciu, D. (2007).
Efficient query evaluation on probabilistic databases.
VLDB Journal, 16(4).

References ii

Dalvi, N. and Suciu, D. (2012).
The dichotomy of probabilistic inference for unions of conjunctive queries.
J. ACM, 59(6).

Kenig, B. and Suciu, D. (2021).
A dichotomy for the generalized model counting problem for unions of conjunctive queries.
In *Proc. PODS*.

Livshits, E., Bertossi, L., Kimelfeld, B., and Sebag, M. (2020).
The Shapley value of tuples in query answering.
In *ICDT*.

Provan, J. S. and Ball, M. O. (1983).
The complexity of counting cuts and of computing the probability that a graph is connected.
SIAM Journal on Computing, 12(4).

Salimi, B. (2016).
Quantifying causal effects on query answering in databases.
In *TaPP*.

Valiant, L. G. (1979).
The complexity of computing the permanent.
TCS, 8(2):189–201.