A Dichotomy Structure Theorem for the Resilience Problem

Cibele Freire

College of Information and Computer Sciences
University of Massachusetts Amherst

joint with work Wolfgang Gatterbauer
& Neil Immerman & Alexandra Meliou

Imagine a world where..

Imagine a world where..

We could selectively delete memories from the brain and by doing that we could change one's opinions or beliefs.

Imagine a world where..

We could selectively delete memories from the brain and by doing that we could change one's opinions or beliefs.

Would you like to know how difficult it would be to choose what memories to delete?

Imagine a world where..

We could selectively delete memories from the brain and by doing that we could change one's opinions or beliefs.

Would you like to know how difficult it would be to choose what memories to delete?

This is what we investigate!

Imagine a world where..

We could selectively delete memories from the brain and by doing that we could change one's opinions or beliefs.

Would you like to know how difficult it would be to choose what memories to delete?

This is what we investigate! Although, in a simpler, safer and more ethical scenario...

Imagine a world where..

We could selectively delete memories from the brain and by doing that we could change one's opinions or beliefs.

Would you like to know how difficult it would be to choose what memories to delete?

This is what we investigate! Although, in a simpler, safer and more ethical scenario...

- Brains = Databases
- Opinions, beliefs = Query answer
- Delete memories = Delete tuples from the database

Resilience problem

Definition (Resilience)

Given a query q and database D, we say that $(D, k) \in \mathtt{RES}(q)$ if and only if $D \models q$ and there exists some $\Gamma \subseteq D$ such that $D - \Gamma \not\models q$ and $|\Gamma| \leq k$.

Resilience problem

Definition (Resilience)

Given a query q and database D, we say that $(D, k) \in \texttt{RES}(q)$ if and only if $D \models q$ and there exists some $\Gamma \subseteq D$ such that $D - \Gamma \not\models q$ and $|\Gamma| \leq k$.

$$q_{vc} :- V(x), E(x, y), V(y)$$

Resilience problem

Definition (Resilience)

Given a query q and database D, we say that $(D, k) \in \mathtt{RES}(q)$ if and only if $D \models q$ and there exists some $\Gamma \subseteq D$ such that $D - \Gamma \not\models q$ and $|\Gamma| \leq k$.

$$q_{vc} :- V(x), E(x, y), V(y)$$

What is the complexity of $\mathtt{RES}(q_{vc})$?

Resilience problem

Definition (Resilience)

Given a query q and database D, we say that $(D, k) \in \texttt{RES}(q)$ if and only if $D \models q$ and there exists some $\Gamma \subseteq D$ such that $D - \Gamma \not\models q$ and $|\Gamma| \leq k$.

$$q_{vc} :- V(x), E(x, y), V(y)$$

What is the complexity of $\texttt{RES}(q_{vc})$? This is exactly vertex cover.

Resilience problem

Definition (Resilience)

Given a query q and database D, we say that $(D, k) \in \texttt{RES}(q)$ if and only if $D \models q$ and there exists some $\Gamma \subseteq D$ such that $D - \Gamma \not\models q$ and $|\Gamma| \leq k$.

$$q_{vc} :- V(x), E(x, y), V(y)$$

What is the complexity of $\texttt{RES}(q_{vc})$? This is exactly vertex cover.

Lemma

$\texttt{RES}(q_{vc})$ *is* NP*-complete.*

Resilience problem

Definition (Resilience)

Given a query q and database D, we say that $(D, k) \in \texttt{RES}(q)$ if and only if $D \models q$ and there exists some $\Gamma \subseteq D$ such that $D - \Gamma \not\models q$ and $|\Gamma| \leq k$.

$$q_{vc} :- V(x), E(x, y), V(y)$$

What is the complexity of $\texttt{RES}(q_{vc})$? This is exactly vertex cover.

Lemma

$\texttt{RES}(q_{vc})$ *is* NP*-complete.*

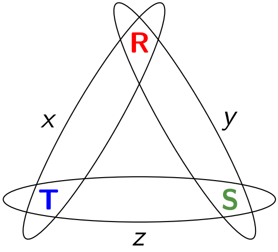
- Conjunctive queries **without self-joins**
- Deleted tuples $\rightarrow$ Contingency set Γ
- Endogenous (changeable) vs exogenous (non changeable) tuples

Dual hypergraph

$q_{\triangle} \mathrel{:-} R(x, y), S(y, z), T(z, x)$ $\qquad$ $q_{\mathrm{T}} \mathrel{:-} A(x), B(y), C(z), W(x, y, z)$

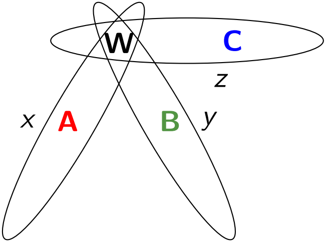
Dual hypergraph

$q_{\triangle} :- R(x, y), S(y, z), T(z, x)$



Triangle query

$q_{\mathrm{T}} :- A(x), B(y), C(z), W(x, y, z)$

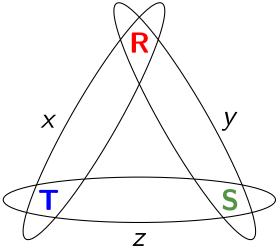


Tripod query

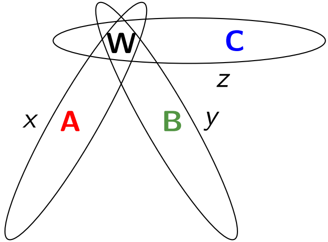
Dual hypergraph

$q_{\triangle} :- R(x,y), S(y,z), T(z,x)$

$q_{\mathrm{T}} :- A(x), B(y), C(z), W(x,y,z)$



Triangle query



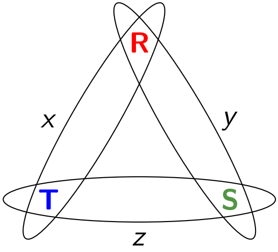
Tripod query

What is the complexity of resilience for those queries?

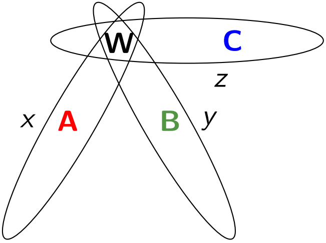
Dual hypergraph

$q_{\triangle}$:– $R(x, y), S(y, z), T(z, x)$

q_{T} :– $A(x), B(y), C(z), W(x, y, z)$



Triangle query



Tripod query

Lemma

$\mathtt{RES}(q_{\triangle})$ *and* $\mathtt{RES}(q_{\mathrm{T}})$ *are NP-complete.*

$\mathtt{RES}(q_\triangle)$ is NP-complete.

$3\text{SAT} \le \mathtt{RES}(q_\triangle)$. Let $\psi = C_1 \wedge \cdots \wedge C_m$ be a 3-CNF formula, $\text{var}(\psi) = \{v_1, \ldots, v_n\}$

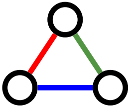
Map $\psi \mapsto (D_\psi, k_\psi)$ s.t. $\quad \psi \in 3\mathrm{SAT} \Leftrightarrow (D_\psi, k_\psi) \in \mathtt{RES}(q_\triangle)$

$\texttt{RES}(q_\triangle)$ is NP-complete.

3SAT $\leq \texttt{RES}(q_\triangle)$. Let $\psi = C_1 \wedge \cdots \wedge C_m$ be a 3-CNF formula, $\mathsf{var}(\psi) = \{v_1, \ldots, v_n\}$

Map $\psi \mapsto (D_\psi, k_\psi)$ s.t. $\quad \psi \in \mathrm{3SAT} \Leftrightarrow (D_\psi, k_\psi) \in \texttt{RES}(q_\triangle)$

$$q_\triangle \coloneq R(x,y), S(y,z), T(z,x)$$

$(D_\psi, k_\psi) \in \texttt{RES}(q_\triangle) \Leftrightarrow \exists \Gamma\ (|\Gamma| = k_\psi) \wedge (D_\psi - \Gamma)$ has no 

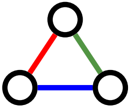
$\mathrm{RES}(q_\triangle)$ is NP-complete.

$3\mathrm{SAT} \le \mathrm{RES}(q_\triangle)$. Let $\psi = C_1 \wedge \cdots \wedge C_m$ be a 3-CNF formula, $\mathrm{var}(\psi) = \{v_1, \ldots, v_n\}$

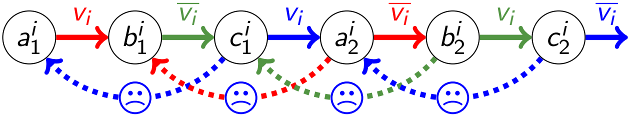
Map $\psi \mapsto (D_\psi, k_\psi)$ s.t. $\quad \psi \in 3\mathrm{SAT} \Leftrightarrow (D_\psi, k_\psi) \in \mathrm{RES}(q_\triangle)$

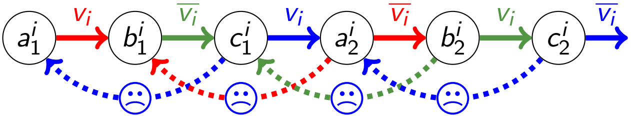
$$q_\triangle \ :-\ R(x,y), S(y,z), T(z,x)$$

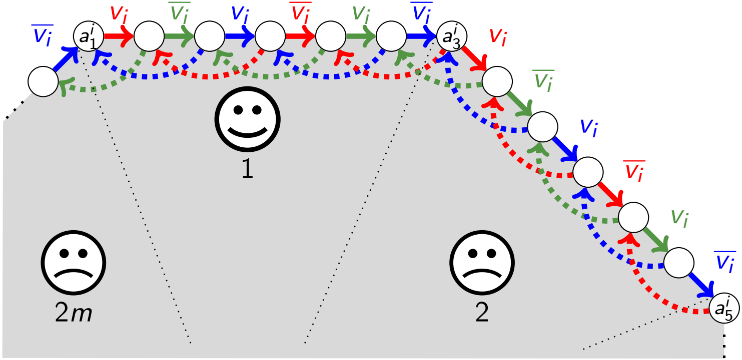
$(D_\psi, k_\psi) \in \mathrm{RES}(q_\triangle) \Leftrightarrow \exists \Gamma\ (|\Gamma| = k_\psi) \wedge (D_\psi - \Gamma)$ has no



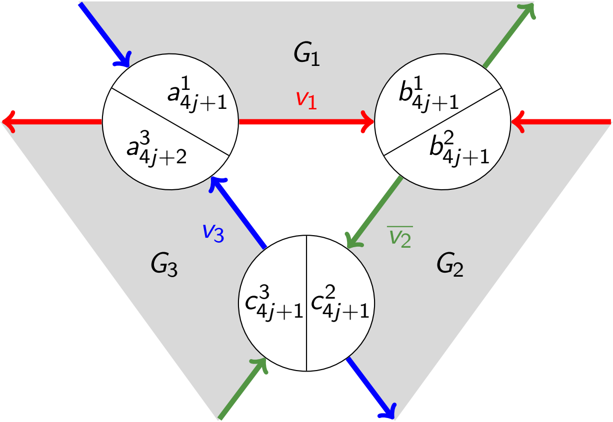
D_ψ has one circular gadget G_i for each variable v_i.



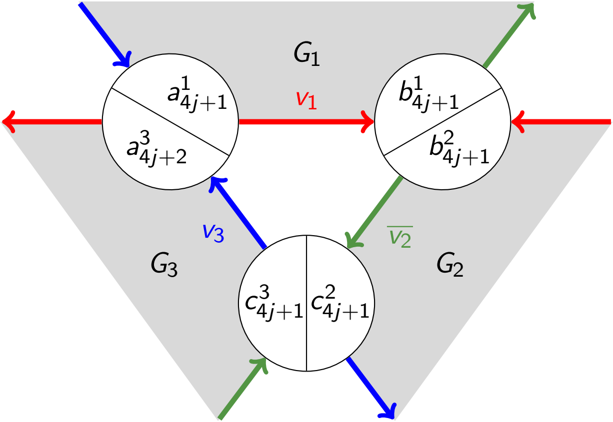

a_1^i
v_i
b_1^i
$\overline{v_i}$
c_1^i
v_i
a_2^i
$\overline{v_i}$
b_2^i
v_i
c_2^i
$\overline{v_i}$


$\overline{v_i}$
a_1^i
v_i
$\overline{v_i}$
v_i
$\overline{v_i}$
v_i
$\overline{v_i}$
a_3^i
v_i
$\overline{v_i}$
v_i
$\overline{v_i}$
v_i
$\overline{v_i}$
a_5^i
1
$2m$
2

For each clause, e.g., $C_j = (v_1 \vee \overline{v_2} \vee v_3)$, pick the jth occurrences of $v_1 \in G_1$, $\overline{v_2} \in G_2$ and $v_3 \in G_3$. Identify head of v_1 with tail of $\overline{v_2}$, head of $\overline{v_2}$ with tail of v_3, head of v_3 with tail of v_1



For each clause, e.g., $C_j = (v_1 \vee \overline{v_2} \vee v_3)$, pick the jth occurrences of $v_1 \in G_1$, $\overline{v_2} \in G_2$ and $v_3 \in G_3$. Identify head of v_1 with tail of $\overline{v_2}$, head of $\overline{v_2}$ with tail of v_3, head of v_3 with tail of v_1



This new RGB triangle is automatically removed iff one of the literals in C_j is chosen true. $\square$

$\texttt{RES}(q_{\mathrm{T}})$ is NP-complete.

$$q_{\mathrm{T}} \ :\!-\ A(x), B(y), C(z), W(x, y, z)$$

$\mathrm{RES}(q_{\mathrm{T}})$ is NP-complete.

$$q_{\mathrm{T}} \text{ :- } A(x), B(y), C(z), W(x, y, z)$$

$\mathrm{var}(A) \subseteq \mathrm{var}(W)$.

$\mathrm{RES}(q_{\mathrm{T}})$ is NP-complete.

$$q_{\mathrm{T}} \text{ :- } A(x), B(y), C(z), W(x, y, z)$$

$\mathrm{var}(A) \subseteq \mathrm{var}(W)$.

A **dominates** W.

$\mathrm{RES}(q_{\mathrm{T}})$ is NP-complete.

$$q_{\mathrm{T}} :- A(x), B(y), C(z), W(x, y, z)$$

$\mathsf{var}(A) \subseteq \mathsf{var}(W)$.

A **dominates** W.

Proposition

If A dominates W, then we can assume that W is **exogenous**, i.e., rewrite as W^{x}, tuples from W^{x} are **never chosen**.

$\mathrm{RES}(q_{\mathrm{T}})$ is NP-complete.

$$q_{\mathrm{T}} \ :- \ A(x), B(y), C(z), W(x, y, z)$$

$\mathsf{var}(A) \ \subseteq \ \mathsf{var}(W)$.

A **dominates** W.

Proposition

If A dominates W, then we can assume that W is **exogenous**, i.e., rewrite as $W^{\times}$, tuples from $W^{\times}$ are **never chosen**.

$$q_{\mathrm{T}} \ :- \ A(x), B(y), C(z), W^{\times}(x, y, z)$$

$\mathtt{RES}(q_{\mathrm{T}})$ is NP-complete.

$$\begin{aligned} q_{\triangle} &:- \quad R(x,y), S(y,z), T(z,x) \\ q_{\mathrm{T}} &:- \quad A(x), B(y), C(z), W^{x}(x,y,z) \end{aligned}$$

Show $\mathtt{RES}(q_{\triangle}) \leq \mathtt{RES}(q_{\mathrm{T}})$

$\mathtt{RES}(q_{\mathrm{T}})$ is NP-complete.

$$q_{\triangle} \quad :- \quad R(x,y), S(y,z), T(z,x)$$
$$q_{\mathrm{T}} \quad :- \quad A(x), B(y), C(z), W^{\times}(x,y,z)$$

Show $\mathtt{RES}(q_{\triangle}) \leq \mathtt{RES}(q_{\mathrm{T}})$

Let (D,k) be an instance of $\mathtt{RES}(q_{\triangle})$.

$(D,k) \mapsto (D',k) \qquad D' \stackrel{\mathrm{def}}{=} (A,B,C,W^{\times})$

$\mathtt{RES}(q_{\mathrm{T}})$ is NP-complete.

$$\begin{aligned} q_{\triangle} &\;\text{:–}\; R(x,y), S(y,z), T(z,x) \\ q_{\mathrm{T}} &\;\text{:–}\; A(x), B(y), C(z), W^{\times}(x,y,z) \end{aligned}$$

Show $\mathtt{RES}(q_{\triangle}) \leq \mathtt{RES}(q_{\mathrm{T}})$

Let (D,k) be an instance of $\mathtt{RES}(q_{\triangle})$.

$(D,k) \mapsto (D',k) \qquad D' \stackrel{\mathrm{def}}{=} (A,B,C,W^{\times})$

$$\begin{aligned} A &= \left\{ \langle ab \rangle \;\middle|\; R(a,b) \in D \right\} \\ B &= \left\{ \langle bc \rangle \;\middle|\; S(b,c) \in D \right\} \\ C &= \left\{ \langle ca \rangle \;\middle|\; T(c,a) \in D \right\} \\ W^{\times} &= \left\{ (\langle ab \rangle, \langle bc \rangle, \langle ca \rangle) \;\middle|\; a,b,c \in \mathrm{dom}(D) \right\} \end{aligned}$$

$\mathtt{RES}(q_{\mathrm{T}})$ is NP-complete.

$$
\begin{aligned}
q_\triangle &\;:-\; R(x,y), S(y,z), T(z,x) \\
q_{\mathrm{T}} &\;:-\; A(x), B(y), C(z), W^{\times}(x,y,z)
\end{aligned}
$$

Show $\mathtt{RES}(q_\triangle) \leq \mathtt{RES}(q_{\mathrm{T}})$

Let (D,k) be an instance of $\mathtt{RES}(q_\triangle)$.

$(D,k) \mapsto (D',k) \qquad D' \stackrel{\text{def}}{=} (A,B,C,W^{\times})$

$$
\begin{aligned}
A &= \{\langle ab\rangle \mid R(a,b) \in D\} \\
B &= \{\langle bc\rangle \mid S(b,c) \in D\} \\
C &= \{\langle ca\rangle \mid T(c,a) \in D\} \\
W^{\times} &= \{(\langle ab\rangle, \langle bc\rangle, \langle ca\rangle) \mid a,b,c \in \mathrm{dom}(D)\}
\end{aligned}
$$

Claim

$$(D,k) \in \mathtt{RES}(q_\triangle) \quad \Leftrightarrow \quad (D',k) \in \mathtt{RES}(q_{\mathrm{T}}). \qquad \square$$

Triads

What do the triangle query and the tripod query have in common?

Triads

What do the triangle query and the tripod query have in common?

Definition (triad)

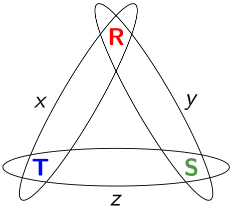
A **triad** is a set of three *endogenous atoms*, $\mathcal{T} = \{S_0, S_1, S_2\}$ such that for every pair i, j, there is a path from S_i to S_j that uses no variable occurring in the other atom of $\mathcal{T}$.

Triads

What do the triangle query and the tripod query have in common?

Definition (triad)

A **triad** is a set of three *endogenous atoms*, $\mathcal{T} = \{S_0, S_1, S_2\}$ such that for every pair i, j, there is a path from S_i to S_j that uses no variable occurring in the other atom of $\mathcal{T}$.

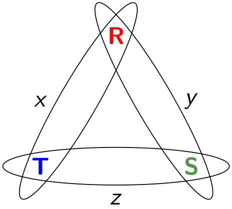


Triads

What do the triangle query and the tripod query have in common?

Definition (triad)

A **triad** is a set of three *endogenous atoms*, $\mathcal{T} = \{S_0, S_1, S_2\}$ such that for every pair i, j, there is a path from S_i to S_j that uses no variable occurring in the other atom of $\mathcal{T}$.



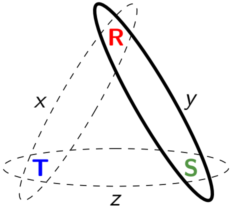
Where is the triad?

Triads

What do the triangle query and the tripod query have in common?

Definition (triad)

A **triad** is a set of three *endogenous atoms*, $\mathcal{T} = \{S_0, S_1, S_2\}$ such that for every pair i, j, there is a path from S_i to S_j that uses no variable occurring in the other atom of $\mathcal{T}$.



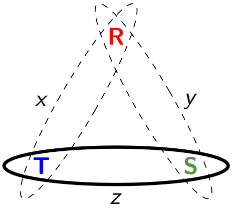
Where is the triad?
Path from R to S.

Triads

What do the triangle query and the tripod query have in common?

Definition (triad)

A **triad** is a set of three *endogenous atoms*, $\mathcal{T} = \{S_0, S_1, S_2\}$ such that for every pair i, j, there is a path from S_i to S_j that uses no variable occurring in the other atom of $\mathcal{T}$.



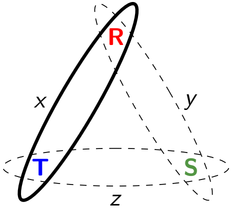
Where is the triad?
Path from S to T.

Triads

What do the triangle query and the tripod query have in common?

Definition (triad)

A **triad** is a set of three *endogenous atoms*, $\mathcal{T} = \{S_0, S_1, S_2\}$ such that for every pair i, j, there is a path from S_i to S_j that uses no variable occurring in the other atom of $\mathcal{T}$.



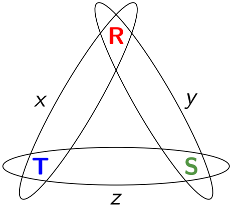
Where is the triad?
Path from T to R.

Triads

What do the triangle query and the tripod query have in common?

Definition (triad)

A **triad** is a set of three *endogenous atoms*, $\mathcal{T} = \{S_0, S_1, S_2\}$ such that for every pair i, j, there is a path from S_i to S_j that uses no variable occurring in the other atom of $\mathcal{T}$.



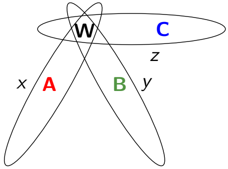
Where is the triad?
$\mathcal{T} = \{R, S, T\}$

Triads

What do the triangle query and the tripod query have in common?

Definition (triad)

A **triad** is a set of three *endogenous atoms*, $\mathcal{T} = \{S_0, S_1, S_2\}$ such that for every pair i, j, there is a path from S_i to S_j that uses no variable occurring in the other atom of $\mathcal{T}$.



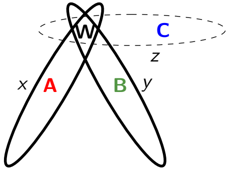
Where is the triad?

Triads

What do the triangle query and the tripod query have in common?

Definition (triad)

A **triad** is a set of three *endogenous atoms*, $\mathcal{T} = \{S_0, S_1, S_2\}$ such that for every pair i, j, there is a path from S_i to S_j that uses no variable occurring in the other atom of $\mathcal{T}$.



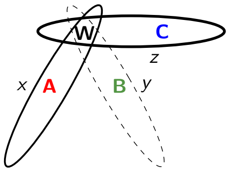
Where is the triad?
Path from A to B.

Triads

What do the triangle query and the tripod query have in common?

Definition (triad)

A **triad** is a set of three *endogenous atoms*, $\mathcal{T} = \{S_0, S_1, S_2\}$ such that for every pair i, j, there is a path from S_i to S_j that uses no variable occurring in the other atom of $\mathcal{T}$.



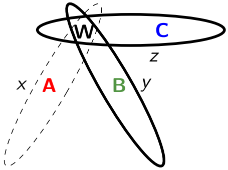
Where is the triad?
Path from A to C.

Triads

What do the triangle query and the tripod query have in common?

Definition (triad)

A **triad** is a set of three *endogenous atoms*, $\mathcal{T} = \{S_0, S_1, S_2\}$ such that for every pair i, j, there is a path from S_i to S_j that uses no variable occurring in the other atom of $\mathcal{T}$.



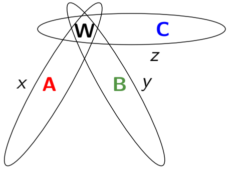
Where is the triad?
Path from B to C.

Triads

What do the triangle query and the tripod query have in common?

Definition (triad)

A **triad** is a set of three *endogenous atoms*, $\mathcal{T} = \{S_0, S_1, S_2\}$ such that for every pair i, j, there is a path from S_i to S_j that uses no variable occurring in the other atom of $\mathcal{T}$.



Where is the triad?
$\mathcal{T} = \{A, B, C\}$

Triads → hardness

Lemma

If q has a triad, then $\texttt{RES}(q)$ *is* NP*-complete.*

Triads $\rightarrow$ hardness

Lemma

If q has a triad, then $\mathtt{RES}(q)$ is NP-complete.

Important remark

It is easy to check if a query has a triad!

Triads → hardness

Lemma

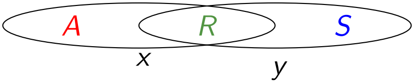
If q has a triad, then $\texttt{RES}(q)$ is NP-complete.

Important remark

It is easy to check if a query has a triad!

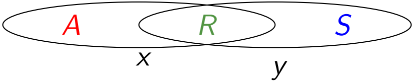
What if a query does not have a triad?

Linear queries



$$q :- A(x), R(x, y), S(y)$$

Linear queries

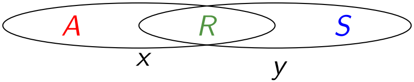


$$q :\!- A(x), R(x,y), S(y)$$

Definition

A query q is **linear** if its atoms may be arranged in a linear order such that each variable occurs in a contiguous sequence of atoms.

Linear queries



$$q :- A(x), R(x, y), S(y)$$

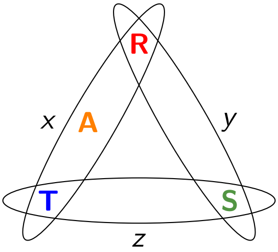
Definition

A query q is **linear** if its atoms may be arranged in a linear order such that each variable occurs in a contiguous sequence of atoms.

Fact [Meliou et.al., VLDB10]

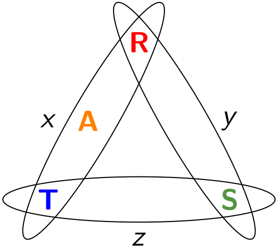
For any linear sj-free CQ q, $\texttt{RES}(q)$ is in P. (Reduction to network flow.)

Is there a triad in the following query?



$$q_{\mathrm{rats}} :- A(x), R(x,y), S(y,z), T(z,x)$$

Is there a triad in the following query?



$$q_{\mathrm{rats}} \coloneq A(x), R(x, y), S(y, z), T(z, x)$$

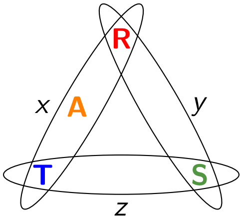
$\mathtt{RES}(q_{\mathrm{rats}})$ is in P!

Domination

T and R are **dominated** by A in q_{rats}. This guarantees we don't need tuples from R, T in a minimum contingency set.

Domination

T and R are **dominated** by A in q_{rats}. This guarantees we don't need tuples from R, T in a minimum contingency set.



$$q_{\mathrm{rats}} :\!-\, A(x), R^x(x,y), S(y,z), T^x(z,x)$$

No triads $\rightarrow$ easy

Lemma

If q is a query in normal form with no triads, we can transform it into a linear query q', such that $\texttt{RES}(q) \leq \texttt{RES}(q')$. Therefore $\texttt{RES}(q)$ is in P.

Dichotomy for Resilience - sj-free CQ

Theorem

Let q be an sj-free CQ and $\texttt{nf}(q)$ its normal form.

- *If* $\texttt{nf}(q)$ *has a triad, then* $\texttt{RES}(q)$ *is* NP*-complete*
- *If* $\texttt{nf}(q)$ *does* **not** *have a triad, then* $\texttt{RES}(q)$ *is in* P.

Adding functional dependencies

$q_{\mathrm{T}} :- A(x), B(y), C(z), W(x, y, z)$

A

	X
a_1	1
a_2	3

B

	Y
b_1	2
b_2	4

C

	Z
c_1	5
c_2	6

W

	X	Y	Z
w_1	1	2	5
w_2	1	2	6
w_3	3	4	5
w_4	1	4	5

Adding functional dependencies

$q_{\mathrm{T}} :- A(x), B(y), C(z), W(x, y, z)$,$\texttt{fd} = W : x \to y$

A

	X
a_1	1
a_2	3

B

	Y
b_1	2
b_2	4

C

	Z
c_1	5
c_2	6

W

	X	Y	Z
w_1	**1**	2	5
w_2	**1**	2	6
w_3	3	4	5
w_4	**1**	4	5

Adding functional dependencies

$q_{\mathrm{T}} :- A(x), B(y), C(z), W(x, y, z)$,$\texttt{fd} = W : x \to y$

A

	X
a_1	1
a_2	3

B

	Y
b_1	2
b_2	4

C

	Z
c_1	5
c_2	6

W

	X	Y	Z
w_1	**1**	2	5
w_2	**1**	2	6
w_3	3	4	5
w_4	**1**	4	5

- FDs constrain the databases we can consider

Adding functional dependencies

$q_{\mathrm{T}} :- A(x), B(y), C(z), W(x, y, z)$,$\texttt{fd} = W : x \to y$

A

	X
a_1	1
a_2	3

B

	Y
b_1	2
b_2	4

C

	Z
c_1	5
c_2	6

W

	X	Y	Z
w_1	**1**	2	5
w_2	**1**	2	6
w_3	3	4	5
w_4	**1**	4	5

- FDs constrain the databases we can consider
- FDs can reduce the complexity of resilience

Adding functional dependencies

Transform the query based on FDs

Adding functional dependencies

Transform the query based on FDs $\rightarrow$ *induced rewrites*

Adding functional dependencies

Transform the query based on FDs $\rightarrow$ *induced rewrites*

$$q_{\mathrm{T}} = A(x), B(y), C(z), W(x, y, z), W : x \rightarrow y$$

$$\downarrow$$

$$q^*_{\mathrm{T}} = \mathbf{A}'(\mathbf{x}, \mathbf{y}), B(y), C(z), W(x, y, z), W : x \rightarrow y$$

$$\downarrow$$

$$q^*_{\mathrm{T}} = A'(x, y), B(y), C(z), W(x, y, z)$$

Adding functional dependencies

Transform the query based on FDs $\rightarrow$ *induced rewrites*

$$q_{\mathrm{T}} = A(x), B(y), C(z), W(x,y,z), W : x \rightarrow y$$

$$\downarrow$$

$$q_{\mathrm{T}}^{*} = \mathbf{A}'(\mathbf{x}, \mathbf{y}), B(y), C(z), W(x,y,z), W : x \rightarrow y$$

$$\downarrow$$

$$q_{\mathrm{T}}^{*} = A'(x,y), B(y), C(z), W(x,y,z)$$

$\texttt{RES}(q_{\mathrm{T}}, \varphi)$ is in P.

Adding functional dependecies

Induced rewrites

Let q be a query and $\overline{v} \to u \in \Phi$ be a functional dependency. We write $(q; \Phi) \rightsquigarrow (q'; \Phi)$ to mean that q' is the result of adding the dependent variable u to some relation that contains all the determinant variables $\overline{v}$. After applying all possible rewrites, we obtain query q^*, which we call **closed query**.

Adding functional dependecies

Induced rewrites

Let q be a query and $\overline{v} \to u \in \Phi$ be a functional dependency. We write $(q; \Phi) \rightsquigarrow (q'; \Phi)$ to mean that q' is the result of adding the dependent variable u to some relation that contains all the determinant variables $\overline{v}$. After applying all possible rewrites, we obtain query q^*, which we call **closed query**.

Lemma

Let q^* be q after all possible induced rewrites have been applied. Then $\mathrm{RES}(q; \Phi) \equiv \mathrm{RES}(q^*; \Phi) \equiv \mathrm{RES}(q^*)$.

Dichotomy for resilience - sj-free CQ + FD

Theorem

Let $(q; \Phi)$ be a sf-free CQ with functional dependencies. Let (q^*, Φ) be its closure under induced rewrites, and such that all dominated atoms of q^* are exogenous.

- *If q^* has a triad then $\mathtt{RES}(q; \Phi)$ is NP-complete.*
- *If q^* does* **not** *have a triad, then $\mathtt{RES}(q; \Phi)$ is in P.*

Future Directions

- Resilience for CQ with self-joins - (q_{vc})
- Deletion propagation: view side-effects for CQ with self-joins
  - Dichotomy results for CQ without self-joins [Kimelfeld et.al., PODS11]
  - Extended to functional dependencies [Kimelfeld, PODS12]
- Characterize the complexity of the parts of the problem that are in P, cf. [Allender, et. al.]