The Complexity of Resilience and Responsibility for Conjunctive Queries

Neil Immerman
College of Information and Computer Sciences
UMass Amherst

to appear in VLDB 2016
joint with Cibele Freire
& Wolfgang Gatterbauer & Alexandra Meliou
The **resilience** of a boolean query with respect to a database, $D$, is the minimum number of tuples that must be removed from $D$ to make the query false.
The **resilience** of a boolean query with respect to a database, $D$, is the minimum number of tuples that must be removed from $D$ to make the query false.

Resilience is crucial to figuring out **why** a certain tuple, $t$, occurs in the answer to a query or view, $q$, on a database, $D$. 

Neil Immerman

**Resilience and Responsibility**
The **resilience** of a boolean query with respect to a database, $D$, is the minimum number of tuples that must be removed from $D$ to make the query false.

Resilience is crucial to figuring out **why** a certain tuple, $t$, occurs in the answer to a query or view, $q$, on a database, $D$. and to computing the **minimum change** needed to remove $t$ from the view.
The resilience of a boolean query with respect to a database, $D$, is the minimum number of tuples that must be removed from $D$ to make the query false.

Resilience is crucial to figuring out why a certain tuple, $t$, occurs in the answer to a query or view, $q$, on a database, $D$, and to computing the minimum change needed to remove $t$ from the view.

Often $D = D^x \cup D^n$ is partly exogenous and partly endogenous.
The **resilience** of a boolean query with respect to a database, \( D \), is the minimum number of tuples that must be removed from \( D \) to make the query false.

Resilience is crucial to figuring out why a certain tuple, \( t \), occurs in the answer to a query or view, \( q \), on a database, \( D \), and to computing the **minimum change** needed to remove \( t \) from the view.

Often \( D = D^x \cup D^n \) is partly **exogenous** and partly **endogenous**.

Treat exogenous part as fixed, beyond our control; only consider possible changes to the endogenous part.
Resilience as a decision problem

\[
\text{RES}(q) = \{(D, k) \mid \exists \Gamma \subseteq D^n (D - \Gamma) \not\models q \& |\Gamma| \leq k\}
\]

Example: \(q_{vc}: V(x)E(x, y)V(y)\)

Prop: \(\text{RES}(q_{vc})\) is NP complete.

Proof. \(\text{RES}(q_{vc})\) is exactly the vertex cover problem: how many vertices need we remove so that no edges remain.

\(q_{vc}\) has a self join.

Goal: Characterize the complexity of resilience for sj-free conjunctive queries.
Resilience as a decision problem

\[
\text{RES}(q) = \{ (D, k) \mid \exists \Gamma \subseteq D^n (D - \Gamma) \not\models q \& |\Gamma| \leq k \}\]

Example: \[q_{vc} ::= V(x) E(x, y) V(y)\]
Resilience as a decision problem

\[
\text{RES}(q) = \{ (D, k) \mid \exists \Gamma \subseteq D^n (D - \Gamma) \not\models q & |\Gamma| \leq k \}
\]

**Example:** \( q_{vc} := V(x) E(x, y) V(y) \)

**Prop:** \( \text{RES}(q_{vc}) \) is NP complete.
Resilience as a decision problem

\[
\text{RES}(q) = \{(D, k) \mid \exists \Gamma \subseteq D^n (D - \Gamma) \not\models q \& |\Gamma| \leq k\}
\]

**Example:** \( q_{vc} \):\( V(x) E(x, y) V(y) \)

**Prop:** \( \text{RES}(q_{vc}) \) is NP complete.

**Proof.**
\( \text{RES}(q_{vc}) \) is exactly the vertex cover problem: how many vertices need we remove so that no edges remain. \( \square \)
Resilience as a decision problem

\[ \text{RES}(q) = \{(D, k) \mid \exists \Gamma \subseteq D^n (D - \Gamma) \not\models q \land |\Gamma| \leq k\} \]

Example: \( q_{vc} := V(x) E(x, y) V(y) \)

**Prop:** \( \text{RES}(q_{vc}) \) is NP complete.

**Proof.** \( \text{RES}(q_{vc}) \) is exactly the vertex cover problem: how many vertices need we remove so that no edges remain.

\( q_{vc} \) has a **self join**.
Resilience as a decision problem

\[
\text{RES}(q) = \{(D, k) \mid \exists \Gamma \subseteq D^n (D - \Gamma) \not\models q \& |\Gamma| \leq k\}
\]

**Example:** \(q_{vc} := V(x) E(x,y) V(y)\)

**Prop:** \(\text{RES}(q_{vc})\) is NP complete.

**Proof.**
\(\text{RES}(q_{vc})\) is exactly the vertex cover problem: how many vertices need we remove so that no edges remain. \(\square\)

\(q_{vc}\) has a **self join**.

**Goal:** Characterize the **complexity** of **resilience** for **sj-free conjunctive queries**.
q_\triangle :— R(x, y), S(y, z), T(z, x)
Triangle Query

\[ q_\triangle \; :\; - \quad R(x, y), \quad S(y, z), \quad T(z, x) \]

Query hypergraph: relations are vertices; variables are hyperedges
The triangle query, $q_\triangle$, is hard.

**Prop.** $\text{RES}(q_\triangle)$ is NP-complete.
The triangle query, $q_{\triangle}$, is hard.

**Prop.** $\text{RES}(q_{\triangle})$ is NP-complete.

**Proof:** Reduce 3SAT to $\text{RES}(q_{\triangle})$. Let $\psi = C_1 \land \cdots \land C_m$ be a 3-CNF formula, $\text{var}(\psi) = \{v_1, \ldots, v_n\}$

Map $\psi \mapsto (D_\psi, k_\psi)$ s.t. $\psi \in \text{3SAT} \iff (D_\psi, k_\psi) \in \text{RES}(q_{\triangle})$
\[ \psi = C_1 \land \cdots \land C_m \quad \text{var}(\psi) = \{v_1, \ldots, v_n\} \quad \psi \mapsto (D_\psi, k_\psi) \]

\[ q_\triangle := R(x, y), S(y, z), T(z, x) \]

\[ (D_\psi, k_\psi) \in \text{RES}(q_\triangle) \iff \exists \Gamma \mid |\Gamma| = k_\psi \land D_\psi - \Gamma \text{ has no} \]
\[
\psi = C_1 \land \cdots \land C_m \quad \text{var}(\psi) = \{v_1, \ldots, v_n\} \quad \psi \mapsto (D_\psi, k_\psi)
\]

\[
q_\triangle := R(x, y), S(y, z), T(z, x)
\]

\[
(D_\psi, k_\psi) \in \text{RES}(q_\triangle) \iff \exists \Gamma \mid |\Gamma| = k_\psi \land D_\psi - \Gamma \text{ has no } D_\psi \text{ has one circular gadget } G_i \text{ for each variable } v_i.
\]

\[
\begin{array}{ccccccc}
a_1^i & \overset{v_i}{\longrightarrow} & b_1^i & \overset{\overline{v}_i}{\longrightarrow} & c_1^i & \overset{v_i}{\longrightarrow} & a_2^i & \overset{\overline{v}_i}{\longrightarrow} & b_2^i & \overset{v_i}{\longrightarrow} & c_2^i
\end{array}
\]
In $G_i$ must choose all $v_i$’s or all $\overline{v}_i$’s
For each clause, e.g., $C_j = (v_1 \lor \overline{v_2} \lor v_3)$, pick the $j$th 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. □
Tripod Query

\[ q_T := A(x), B(y), C(z), W(x, y, z) \]
Tripod Query

\[ q_T := A(x), B(y), C(z), W(x, y, z) \]

**Prop.** \( \text{RES}(q_T) \) is NP complete.
RES\(\left(q_T\right)\) is NP complete.

\[ q_T := A(x), B(y), C(z), W(x, y, z) \]
RES\( (q_T) \) is NP complete.

\[
q_T ::= A(x), B(y), C(z), W(x, y, z)
\]

\[\text{var}(A) \subseteq \text{var}(W).\]

\(A\) dominates \(W\).
\( \text{RES}(q_T) \) is NP complete.

\[
q_T \defined A(x), B(y), C(z), W(x, y, z)
\]

\( \text{var}(A) \subseteq \text{var}(W) \).

A dominates \( W \).

**Prop.** 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.
$\text{RES}(q_T)$ is NP complete.

$q_T : A(x), B(y), C(z), W(x, y, z)$

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

$A$ dominates $W$.

**Prop.** 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.

$q_T : A(x), B(y), C(z), W^\times(x, y, z)$
RES\left(q_T\right) is NP complete.

\[ q_\Delta := R(x, y), S(y, z), T(z, x) \]
\[ q_T := A(x), B(y), C(z), W^x(x, y, z) \]

**Proof:** Show RES\left(q_\Delta\right) \leq RES\left(q_T\right)
\( \text{RES}(q_T) \) is NP complete.

\[ q_\triangle := R(x, y), S(y, z), T(z, x) \]

\[ q_T := A(x), B(y), C(z), W^x(x, y, z) \]

**Proof:** Show \( \text{RES}(q_\triangle) \leq \text{RES}(q_T) \)

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

\((D, k) \mapsto (D', k) \quad D' \stackrel{\text{def}}{=} (A, B, C, W^x)\)
\( \text{RES}(q_T) \) is NP complete.

\[
\begin{align*}
q_\triangle & := R(x, y), S(y, z), T(z, x) \\
q_T & := A(x), B(y), C(z), W^x(x, y, z)
\end{align*}
\]

**Proof:** Show \( \text{RES}(q_\triangle) \leq \text{RES}(q_T) \)

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

\((D, k) \mapsto (D', k) \quad D' \overset{\text{def}}{=} (A, B, C, W^x)\)

\[
\begin{align*}
A & = \{ \langle ab \rangle \mid R(a, b) \in D \} \\
B & = \{ \langle bc \rangle \mid S(b, c) \in D \} \\
C & = \{ \langle ca \rangle \mid T(c, a) \in D \} \\
W^x & = \{ (\langle ab \rangle, \langle bc \rangle, \langle ca \rangle) \mid a, b, c \in \text{dom}(D) \}
\end{align*}
\]
**RES**\((q_T)\) is NP complete.

\[
q_{\triangle} \,:= \, R(x, y), S(y, z), T(z, x) \\
q_T \,:= \, A(x), B(y), C(z), W^x(x, y, z)
\]

**Proof:** Show \(\text{RES}(q_{\triangle}) \leq \text{RES}(q_T)\)

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

\((D, k) \mapsto (D', k) \quad D' \overset{\text{def}}{=} (A, B, C, W^x)\)

\[
A = \{ \langle ab \rangle \mid R(a, b) \in D \} \\
B = \{ \langle bc \rangle \mid S(b, c) \in D \} \\
C = \{ \langle ca \rangle \mid T(c, a) \in D \} \\
W^x = \{ ((\langle ab \rangle, \langle bc \rangle, \langle ca \rangle) \mid a, b, c \in \text{dom}(D) \}
\]

**Claim** \((D, k) \in \text{RES}(q_{\triangle}) \iff (D', k) \in \text{RES}(q_T)\). \(\square\)
**Def.** A query is **linear** if all of the vertices of its hypergraph can be drawn along a straight line with all of its hyperedges convex.

For example, the following query is linear:

\[ q :\ A(x), \ R(x, y), \ S(y, z) \]
**Prop.** For any linear sj-free conjunctive query $q$, $\text{RES}(q) \in P$. 

Use Network Flow. $\text{RES}(D, q)$ is the min cut of corresponding network.
**Prop.** For any linear sj-free conjunctive query $q$, $\text{RES}(q) \in \mathbb{P}$.

**Proof:** Use Network Flow.

$\text{RES}(D, q)$ is the min cut of corresponding network.

$q :\!-\! A(x) \qquad R(x, y) \qquad S(y, z)$
\[ q_{\text{rats}} \defeq A(x), R(x, y), S(y, z), T(z, x) \]
Is Rats hard or easy?

$q_{\text{rats}} :=$ $A(x), R(x, y), S(y, z), T(z, x)$
Is Rats **hard** or **easy**?

\[
q_{\text{rats}} \coloneqq A(x), R(x, y), S(y, z), T(z, x)
\]

\[
q_1 \equiv A(x), R^x(x, y), S(y, z), T^x(z, x)
\]

\[
\text{RES}(q_{\text{rats}}) \equiv \text{RES}(q_1)
\]
Is Rats hard or easy?

\[
q_{\text{rats}} := A(x), R(x, y), S(y, z), T(z, x)
\]
\[
q_1 \equiv A(x), R^x(x, y), S(y, z), T^x(z, x) \quad \text{Domination}
\]
\[
\text{RES}(q_{\text{rats}}) \equiv \text{RES}(q_1)
\]
\[
q_2 \equiv A(x), R^x(x, y, z), S(y, z), T^x(z, x) \quad \text{Dissociation}
\]
\[
\text{RES}(q_1) \leq \text{RES}(q_2)
\]
Is Rats **hard** or **easy**?

\[ q_{rats} := A(x), R(x, y), S(y, z), T(z, x) \]
\[ q_1 \equiv A(x), R^x(x, y), S(y, z), T^x(z, x) \quad \text{Domination} \]
\[ \text{RES}(q_{rats}) \equiv \text{RES}(q_1) \]
\[ q_2 \equiv A(x), R^x(x, y, z), S(y, z), T^x(z, x) \quad \text{Dissociation} \]
\[ \text{RES}(q_1) \leq \text{RES}(q_2) \]
\[ q_3 \equiv A(x), R^x(x, y, z), S(y, z), T^x(z, x, y) \quad \text{Dissociation} \]
\[ \text{RES}(q_2) \leq \text{RES}(q_3) \]
Is Rats **hard** or **easy**?

\[
\begin{align*}
q_{\text{rats}} & := A(x), R(x, y), S(y, z), T(z, x) \\
q_1 & \equiv A(x), R^x(x, y), S(y, z), T^x(z, x) \quad \text{Domination} \\
\text{RES}(q_{\text{rats}}) & \equiv \text{RES}(q_1) \\
q_2 & \equiv A(x), R^x(x, y, z), S(y, z), T^x(z, x) \quad \text{Dissociation} \\
\text{RES}(q_1) & \leq \text{RES}(q_2) \\
q_3 & \equiv A(x), R^x(x, y, z), S(y, z), T^x(z, x, y) \quad \text{Dissociation} \\
\text{RES}(q_2) & \leq \text{RES}(q_3) \\
q_4 & \equiv A(x), R^x(x, y, z), S(y, z) \quad \text{Repetition} \\
\text{RES}(q_3) & \equiv \text{RES}(q_4)
\end{align*}
\]
Is Rats hard or easy?

\[
q_{\text{rats}} := A(x), R(x, y), S(y, z), T(z, x)
\]

\[
q_1 := A(x), R^x(x, y), S(y, z), T^x(z, x)
\]

Domination

\[
\text{RES}(q_{\text{rats}}) \equiv \text{RES}(q_1)
\]

\[
q_2 := A(x), R^x(x, y, z), S(y, z), T^x(z, x)
\]

Dissociation

\[
\text{RES}(q_1) \leq \text{RES}(q_2)
\]

\[
q_3 := A(x), R^x(x, y, z), S(y, z), T^x(z, x, y)
\]

Dissociation

\[
\text{RES}(q_2) \leq \text{RES}(q_3)
\]

\[
q_4 := A(x), R^x(x, y, z), S(y, z)
\]

Repetition

\[
\text{RES}(q_3) \equiv \text{RES}(q_4)
\]

\[
q_4 \text{ is linear and therefore easy!}
\]
Is Rats hard or easy?

\[ q_{\text{rats}} \equiv A(x), R(x, y), S(y, z), T(z, x) \]
\[ q_1 \equiv A(x), R^x(x, y), S(y, z), T^x(z, x) \quad \text{Domination} \]
\[ \text{RES}(q_{\text{rats}}) \equiv \text{RES}(q_1) \]
\[ q_2 \equiv A(x), R^x(x, y, z), S(y, z), T^x(z, x) \quad \text{Dissociation} \]
\[ \text{RES}(q_1) \leq \text{RES}(q_2) \]
\[ q_3 \equiv A(x), R^x(x, y, z), S(y, z), T^x(z, x, y) \quad \text{Dissociation} \]
\[ \text{RES}(q_2) \leq \text{RES}(q_3) \]
\[ q_4 \equiv A(x), R^x(x, y, z), S(y, z) \quad \text{Repetition} \]
\[ \text{RES}(q_3) \equiv \text{RES}(q_4) \]
\[ q_4 \text{ is linear and therefore easy!} \]
\[ \text{RES}(q_{\text{rats}}) \leq \text{RES}(q_5) \quad q_{\text{rats}} \text{ is easy!} \]
What do the triangle and the tripod have in common?

\[ q_\triangle := R(x, y), S(y, z), T(z, x) \]

\[ q_T := A(x), B(y), C(z), W^x(x, y, z) \]
What do the triangle and the tripod have in common?

\[ q_\triangle := R(x, y), S(y, z), T(z, x) \quad q_T := A(x), B(y), C(z), W^x(x, y, z) \]

**Def.** 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 \( \mathcal{T} \).
What do the triangle and the tripod have in common?

$\{R(x, y), S(y, z), T(z, x)\}$ is a triad in $q_\triangle$.

$\{A(x), B(y), C(z), W^x(x, y, z)\}$ is a triad in $q_T$.

**Def.** A **triad** is a set of three endogenous atoms, $T = \{S_0, S_1, S_2\}$ 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 $T$.

$\{R, S, T\}$ is a triad in $q_\triangle$. 

Neil Immerman  
Resilience and Responsibility
What do the triangle and the tripod have in common?

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

**Def.** 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 $\mathcal{T}$.

$\{R, S, T\}$ is a triad in $q_{\triangle}$.

$\{A, B, C\}$ is a triad in $q_T$. 
Lemma Let $q$ be an sj-free conjunctive query where all dominated atoms are exogenous. If $q$ has a triad, then $\text{RES}(q)$ is NP-complete.
Lemma Let $q$ be an sj-free conjunctive query where all dominated atoms are exogenous. If $q$ has a triad, then $\text{RES}(q)$ is NP-complete.

Proof: Show $\text{RES}(q_{\triangle}) \leq \text{RES}(q)$
**Lemma** Let $q$ be an $sj$-free conjunctive query that has no triad. Then $\text{RES}(q) \in P$. 
**Lemma** Let $q$ be an sj-free conjunctive query that has no triad. Then $\text{RES}(q) \in P$.

**Proof:** By induction on the number of *endogenous* atoms in $q$ that we can transform it into a linear query by using dissociations.
Lemma Let $q$ be an sj-free conjunctive query that has no triad. Then $\text{RES}(q) \in P$.

Proof: By induction on the number of endogenous atoms in $q$ that we can transform it into a linear query by using dissociations.

Inductive case: assume true for triad-free queries with $n$ endogenous atoms. Let $q_{n+1}$ be triad-free and have $n + 1$ endogenous atoms.
Lemma Let $q$ be an sj-free conjunctive query that has no triad. Then $\text{RES}(q) \in \mathbb{P}$.

Proof: By induction on the number of endogenous atoms in $q$ that we can transform it into a linear query by using dissociations.

Inductive case: assume true for triad-free queries with $n$ endogenous atoms. Let $q_{n+1}$ be triad-free and have $n+1$ endogenous atoms.

Since there is no triad, we can linearize the endogenous atoms:

\[
\begin{align*}
S_1 & \quad E_1^x & \quad S_2 & \quad E_2^x & \quad \cdots \\
& \quad c_1 & \quad \vdots & \quad \vdots & \quad \vdots \\
& \quad \vdots & \quad c_2 & \quad \vdots & \quad \vdots & \vdots \\
& \quad \vdots & \quad \vdots & \quad \vdots & \quad \vdots & \vdots \\
S_n & \quad E_{n-1}^x & \quad S_{n+1} & \quad E_n^x & \quad c_n & \quad c_{n+1}
\end{align*}
\]
**Dichotomy Theorem for Resilience:** Let $q$ be a sj-free conjunctive query all of whose dominated atoms are exogenous. If $q$ has a triad then $\text{RES}(q)$ is NP complete. Otherwise, $\text{RES}(q) \in \mathbb{P}$. 
Extend to Databases with Functional Dependencies

Let \( q^* \) be \( q \) after all possible induced rewrites have been applied.

Lemma: \( \text{RES}(q) \equiv \text{RES}(q^*) \)

Dichotomy Theorem for Resilience with FD's

Let \( q^* \) be an sj-free conjunctive query with FD's, all possible induced rewrites applied and all dominated atoms are exogenous. If \( q^* \) has a triad then \( \text{RES}(q) \) is NP complete. Otherwise, \( \text{RES}(q) \in P \).

Corollary: Induced rewrites characterized the effect of FD's:

\[
\text{RES}(q; \Phi) \equiv \text{RES}(q^*; \Phi) \equiv \text{RES}(q^*)
\]
induced rewrites preserve complexity of resilience:
q :− R(x, y), S(y, z), T(z, x); x → y

q* :− R(x, y), S(y, z), T(z, x, y); x → y

Let q* be q after all possible induced rewrites have been applied.
induced rewrites preserve complexity of resilience:

$$q : - R(x, y), S(y, z), T(z, x); \ x \mapsto y$$

$$q^* : - R(x, y), S(y, z), T(z, x, y); \ x \mapsto y$$

Let $q^*$ be $q$ after all possible induced rewrites have been applied.

**Lemma:** $\text{RES}(q) \equiv \text{RES}(q^*)$
induced rewrites preserve complexity of resilience:

\[ q :\neg R(x, y), S(y, z), T(z, x); x \mapsto y \]

\[ q^* :\neg R(x, y), S(y, z), T(z, x, y); x \mapsto y \]

Let \( q^* \) be \( q \) after all possible induced rewrites have been applied.

**Lemma:** \( \text{RES}(q) \equiv \text{RES}(q^*) \)

**Dichotomy Theorem for Resilience with FD’s** Let \( q^* \) be an sj-free conjunctive query with FD’s, all possible induced rewrites applied and all dominated atoms are exogenous. If \( q^* \) has a triad then \( \text{RES}(q) \) is NP complete. Otherwise, \( \text{RES}(q) \in \mathbb{P} \).
induced rewrites preserve complexity of resilience:

\[ q : \neg R(x, y), S(y, z), T(z, x); \ x \mapsto y \]

\[ q^* : \neg R(x, y), S(y, z), T(z, x, y); \ x \mapsto y \]

Let \( q^* \) be \( q \) after all possible induced rewrites have been applied.

**Lemma:** \( \text{RES}(q) \equiv \text{RES}(q^*) \)

**Dichotomy Theorem for Resilience with FD’s** Let \( q^* \) be an sj-free conjunctive query with FD’s, all possible induced rewrites applied and all dominated atoms are exogenous. If \( q^* \) has a triad then \( \text{RES}(q) \) is NP complete. Otherwise, \( \text{RES}(q) \in P \).

**Corollary** Induced rewrites characterized the effect of FD’s:

\[ \text{RES}(q; \Phi) \equiv \text{RES}(q^*; \Phi) \equiv \text{RES}(q^*) \]
Future Directions

- Extend characterization of complexity of resilience to conjunctive queries with self joins.

- Extend to joins with FD's.

- Extend to the complexity of "view side-effects" problem.

- Characterize the complexity of the parts of the problem that are in P, cf. [Allender, et. al.]

- Understand & explain Dichotomy Phenomenon.
Future Directions

▶ Extend characterization of complexity of resilience to conjunctive queries with self joins.
▶ Extend to sj’s with FD’s.
Future Directions

▶ Extend characterization of complexity of resilience to conjunctive queries with self joins.
▶ Extend to sj’s with FD’s.
▶ Extend to the complexity of “view side-effects” problem.
Future Directions

- Extend characterization of complexity of resilience to conjunctive queries with self joins.
- Extend to sj’s with FD’s.
- Extend to the complexity of “view side-effects” problem.
- Characterize the complexity of the parts of the problem that are in P, cf. [Allender, et. al.]
Future Directions

- Extend characterization of complexity of resilience to conjunctive queries with self joins.
- Extend to sj’s with FD’s.
- Extend to the complexity of “view side-effects” problem.
- Characterize the complexity of the parts of the problem that are in P, cf. [Allender, et. al.]
- **Understand & explain Dichotomy Phenomenon**