Consistent Query Answering for Primary Keys on Path Queries

Paris Koutris \(^1\)  Xiating Ouyang \(^1\)  Jef Wijsen \(^2\)

University of Wisconsin–Madison \(^1\)

University of Mons \(^2\)

PODS, Xi’an Shaanxi, China, June 20–25 2021
Data model

- **Primary key** constraint as the only integrity constraint
- **Inconsistent** relational databases violating the primary constraint
- A **repair** is an inclusion-maximal consistent subinstance

<table>
<thead>
<tr>
<th>Univ</th>
<th><strong>Acronym</strong></th>
<th>City</th>
<th>Weather</th>
</tr>
</thead>
<tbody>
<tr>
<td>UW</td>
<td>Madison</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UW</td>
<td>Seattle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UMONS</td>
<td>Mons</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th><strong>City</strong></th>
<th>Weather</th>
</tr>
</thead>
<tbody>
<tr>
<td>Madison</td>
<td>Snow</td>
<td></td>
</tr>
<tr>
<td>Seattle</td>
<td>Rain</td>
<td></td>
</tr>
<tr>
<td>Mons</td>
<td>Sunny</td>
<td></td>
</tr>
</tbody>
</table>

Q: Is there a university snowing today?

\[ Q() : \neg \text{Univ}(x, y), \text{Weather}(y, \text{'Snow'}) \]
The problem statement

Consistent query answering – CERTAINTY(q)

INPUT: an inconsistent database db

QUESTION: Is the fixed Boolean query q true in all repairs of db?

\[ q_1() : \neg R(x, y), S(y, z) \]

\[ q_2() : \neg R(x, z), S(y, z) \]

\[ \text{CERTAINTY}(q_1) \iff \exists x (\exists y R(x, y) \land \forall y R(x, y) \rightarrow \exists z S(y, z)) \]

\[ \text{coNP-complete} \]

Paris Koutris, Xiating Ouyang, Jef Wijsen
CQA on Path Queries
PODS 2021 3 / 20
The problem statement

Consistent query answering – CERTAINTY($q$)

INPUT: an inconsistent database $db$
QUESTION: Is the fixed Boolean query $q$ true in all repairs of $db$?

$$q_1() : -R(x, y), S(y, z)$$

$$q_2() : -R(x, z), S(y, z)$$

$$\text{CERTAINTY}(q_1) \iff \exists x (\exists y R(x, y) \land \forall y R(x, y) \rightarrow \exists z S(y, z))$$

Paris Koutris, Xiating Ouyang, Jef Wijsen
CQA on Path Queries
PODS 2021
The problem statement

Consistent query answering – CERTAINTY($q$)

**INPUT:** an inconsistent database $db$

**QUESTION:** Is the **fixed** Boolean query $q$ true in all repairs of $db$?

$\text{CERTAINTY}(q_1) \iff \exists x (\exists y \mathcal{R}(x, y) \land \forall y \mathcal{R}(x, y) \rightarrow \exists z \mathcal{S}(y, z))$

$q_1() : \neg \mathcal{R}(x, y), \mathcal{S}(y, z)$

$q_2() : \neg \mathcal{R}(x, z), \mathcal{S}(y, z)$
The problem statement

Consistent query answering – CERTAINTY($q$)

**INPUT:** an inconsistent database $db$

**QUESTION:** Is the **fixed** Boolean query $q$ true in all repairs of $db$?

\[
\text{CERTAINTY}(q_1) \iff \exists x (\exists y R(x, y) \land \forall y R(x, y) \rightarrow \exists z S(y, z))
\]

\[
q_1() : \neg R(x, z), S(y, z)
\]

\[
q_2() : \neg R(x, z), S(y, z)
\]
The problem statement

Consistent query answering – CERTAINTY($q$)

**INPUT:** an inconsistent database $db$

**QUESTION:** Is the **fixed** Boolean query $q$ true in all repairs of $db$?

---

$$q_1() : \neg R(x, y), S(y, z)$$

$$q_2() : \neg R(x, z), S(y, z)$$

$$\text{CERTAINTY}(q_1) \iff \exists x (\exists y R(x, y) \land \forall y R(x, y) \rightarrow \exists z S(y, z))$$
Conjecture

For any BCQ \( q \), \( \text{CERTAINTY}(q) \) is either in PTIME or coNP-complete.


**Conjecture**

For any BCQ $q$, CERTAINTY($q$) is either in PTIME or coNP-complete.

<table>
<thead>
<tr>
<th>Classification</th>
<th>BCQ Class</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>FO, non-FO</td>
<td>$C_{\text{forest}}$ (SJF_{self-join-free})</td>
<td>[Fuxman and Miller, ICDT’05]</td>
</tr>
<tr>
<td>FO, non-FO</td>
<td>SJF $\alpha$-acyclic queries</td>
<td>[Wijsen, PODS’10]</td>
</tr>
<tr>
<td>P, coNP-comp.</td>
<td>SJF two atoms</td>
<td>[Kolaitis and Pema, 12]</td>
</tr>
<tr>
<td>P, coNP-comp.</td>
<td>SJF simple keys</td>
<td>[Koutris and Suciu, ICDT’14]</td>
</tr>
<tr>
<td>FO, P \ FO, coNP-comp.</td>
<td>SJF</td>
<td>[Koutris and Wijsen, PODS’15]</td>
</tr>
<tr>
<td>FO, L-comp., coNP-comp.</td>
<td>SJF</td>
<td>[Koutris and Wijsen, ICDT’19]</td>
</tr>
<tr>
<td>FO</td>
<td>SJF path</td>
<td>(implied by [KW, PODS’15])</td>
</tr>
</tbody>
</table>
## Conjecture

For any BCQ $q$, $\text{CERTAINTY}(q)$ is either in PTIME or coNP-complete.

<table>
<thead>
<tr>
<th>Classification</th>
<th>BCQ Class</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>FO, non-FO</td>
<td>$C_{forest}$ (SJF$_{(self-join-free)}$)</td>
<td>[Fuxman and Miller, ICDT’05]</td>
</tr>
<tr>
<td>FO, non-FO</td>
<td>SJF $\alpha$-acyclic queries</td>
<td>[Wijsen, PODS’10]</td>
</tr>
<tr>
<td>P, coNP-comp.</td>
<td>SJF two atoms</td>
<td>[Kolaitis and Pema, 12]</td>
</tr>
<tr>
<td>P, coNP-comp.</td>
<td>SJF simple keys</td>
<td>[Koutris and Suciu, ICDT’14]</td>
</tr>
<tr>
<td>FO, P \ FO, coNP-comp.</td>
<td>SJF</td>
<td>[Koutris and Wijsen, PODS’15]</td>
</tr>
<tr>
<td>FO, L-comp., coNP-comp. $\star$</td>
<td>SJF</td>
<td>[Koutris and Wijsen, ICDT’19]</td>
</tr>
<tr>
<td>FO</td>
<td>SJF path</td>
<td>(implied by [KW, PODS’15])</td>
</tr>
</tbody>
</table>

$q() : -R_1(x_1, x_2), R_2(x_2, x_3), \cdots, R_n(x_n, x_{n+1}) \implies R_1R_2 \cdots R_n$

distinct variables $x_i$, relation names $R_i$
### Conjecture

For any BCQ $q$, $\text{CERTAINTY}(q)$ is either in PTIME or coNP-complete.

<table>
<thead>
<tr>
<th>Classification</th>
<th>BCQ Class</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{FO, non-FO}$</td>
<td>$C_{\text{forest}}$ (SJF$_{(\text{self-join-free})}$)</td>
<td>[Fuxman and Miller, ICDT’05]</td>
</tr>
<tr>
<td>$\text{FO, non-FO}$</td>
<td>SJF $\alpha$-acyclic queries</td>
<td>[Wijsen, PODS’10]</td>
</tr>
<tr>
<td>$\text{P, coNP-comp.}$</td>
<td>SJF two atoms</td>
<td>[Kolaitis and Pema, 12]</td>
</tr>
<tr>
<td>$\text{P, coNP-comp.}$</td>
<td>SJF simple keys</td>
<td>[Koutris and Suciu, ICDT’14]</td>
</tr>
<tr>
<td>$\text{FO, P \setminus FO, coNP-comp.}$</td>
<td>SJF</td>
<td>[Koutris and Wijsen, PODS’15]</td>
</tr>
<tr>
<td>$\text{FO, L-comp., coNP-comp. } \star$</td>
<td>SJF</td>
<td>[Koutris and Wijsen, ICDT’19] (implied by [KW, PODS’15])</td>
</tr>
<tr>
<td>$\text{FO \leftarrow}$</td>
<td>SJF path</td>
<td>(Our result)</td>
</tr>
</tbody>
</table>

$$q() : \neg R_1(x_1, x_2), R_2(x_2, x_3), \ldots, R_n(x_n, x_{n+1}) \quad \Rightarrow \quad R_1R_2 \cdots R_n$$

distinct variables $x_i$, relation names $R_i$

$$q() : \neg R(x, y), R(y, z), X(z, w) \quad \Rightarrow \quad RRX$$
The curse of self-joins

Theorem (Deletion Propagation, TODS 2012)

For every CQ without self-joins, deletion propagation is either APX-hard or solvable (in polynomial time) by the unidimensional algorithm.

Theorem (Pricing, JACM 2015)

Let $Q$ be a CQ without self-joins. The data complexity for $\text{PRICE}(Q)$ is either in PTIME or NP-complete.

Theorem (Query resilience, PODS 2020)

Let $q$ be a single-self-join-CQ with at most two occurrences of the self-join relation. The problem $\text{RES}(q)$ is either in PTIME or NP-complete.

Theorem (Our result)

Let $q$ be a path query. The problem $\text{CERTAINTY}(q)$ is either in FO, NL-complete, PTIME-complete or coNP-complete.
The curse of self-joins

Theorem (Deletion Propagation, TODS 2012)
For every CQ without self-joins, deletion propagation is either APX-hard or solvable (in polynomial time) by the unidimensional algorithm.

Theorem (Pricing, JACM 2015)
Let $Q$ be a CQ without self-joins. The data complexity for $\text{PRICE}(Q)$ is either in $\text{PTIME}$ or $\text{NP}$-complete.

Theorem (Query resilience, PODS 2020)
Let $q$ be a single-self-join-CQ with at most two occurrences of the self-join relation. The problem $\text{RES}(q)$ is either in $\text{PTIME}$ or $\text{NP}$-complete.

Theorem (Our result)
Let $q$ be a path query. The problem $\text{CERTAINTY}(q)$ is either in $\text{FO}$, $\text{NL}$-complete, $\text{PTIME}$-complete or $\text{coNP}$-complete.
Outline

1. Handling self-joins
2. Classification result
3. Proof sketch
1. Handling self-joins

2. Classification result

3. Proof sketch
The notion of “rewinding”

**INPUT** to CERTAINTY(\(RRX\)):

<table>
<thead>
<tr>
<th></th>
<th>(A_1)</th>
<th>(A_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

\[ X \]

<table>
<thead>
<tr>
<th></th>
<th>(B_1)</th>
<th>(B_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

**QUESTION:** Do all repairs contain an \(RRX\) path?

\[ 0 \rightarrow R \leftarrow 1 \rightarrow R \leftarrow 3 \rightarrow X \rightarrow 4 \]

\( RRX \) \( RRX \) \( RRX \)

\[ 0 \rightarrow R \leftarrow 1 \rightarrow R \leftarrow 3 \rightarrow X \rightarrow 4 \]

\( RRX \) \( RRX \) \( RRX \)

\( RRX \) \( RRX \) \( RRX \)
The notion of “rewinding”

Input to CERTAINTY(\textit{RRX}):\[
\begin{array}{c|cc}
R & A_1 & A_2 \\
\hline
0 & 1 \\
1 & 2 \\
1 & 3 \\
2 & 3 \\
\end{array} \quad X \quad \begin{array}{c|cc}
 & B_1 & B_2 \\
\hline
 & 3 & 4 \\
\end{array}
\]

QUESTION: Do all repairs contain an \textit{RRX} path?
The notion of “rewinding”

INPUT to CERTAINTY(\(RRX\)):

<table>
<thead>
<tr>
<th></th>
<th>(A_1)</th>
<th>(A_2)</th>
<th>(B_1)</th>
<th>(B_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(R)</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

QUESTION: Do all repairs contain an \(RRX\) path?

\(r_1\)

\(r_2\)
### The notion of “rewinding”

#### INPUT to \text{CERTAINTY}(RRX):

<table>
<thead>
<tr>
<th>R</th>
<th>A₁</th>
<th>A₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X</th>
<th>B₁</th>
<th>B₂</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

#### QUESTION: Do all repairs contain an RRX path?

- \( r₁ \)
  - \( 0 \xrightarrow{R} 1 \xrightarrow{R} 3 \xrightarrow{X} 4 \)
  - \( RRX \quad RRX \quad RRX \)

- \( r₂ \)
  - \( 0 \xrightarrow{R} 1 \xrightarrow{R} 3 \xrightarrow{X} 4 \)
  - \( RRX \quad RRX \quad RRX \quad RRX \)
The notion of “rewinding”

**INPUT to CERTAINTY(\textit{RRX}):**

\[
\begin{array}{c|cc}
R & A_1 & A_2 \\
\hline
0 & 1 \\
1 & 2 \\
1 & 3 \\
2 & 3 \\
\end{array}
\quad X \quad \begin{array}{c|cc}
B_1 & B_2 \\
\hline
3 & 4 \\
\end{array}
\]

**QUESTION:** Do all repairs contain an \textit{RRX} path?

\[ \begin{array}{c}
0 \xrightarrow{R} 1 \xrightarrow{R} 3 \xrightarrow{X} 4 \\
R \quad RRX \quad RRX \quad RRX
\end{array} \quad \begin{array}{c}
0 \xrightarrow{R} 1 \xrightarrow{R} 3 \xrightarrow{X} 4 \\
RRX \quad RRX \quad RRX \quad RRX
\end{array} \]
The notion of “rewinding”

INPUT to CERTAINTY( RRX ):

<table>
<thead>
<tr>
<th>$R$</th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$X$</th>
<th>$B_1$</th>
<th>$B_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

QUESTION: Do all repairs contain an $RRX$ path?

$r_1$

0 $\xrightarrow{R}$ 1 $\xrightarrow{R}$ 3 $\xrightarrow{X}$ 4

$RRX$ $RRX$ $RRX$

$r_2$

0 $\xrightarrow{R}$ 1 $\xrightarrow{R}$ 3 $\xrightarrow{X}$ 4

$RRX$ $RRX$ $RRX$
The notion of “rewinding”

INPUT to CERTAINTY(\textit{RRX}): 

\[
\begin{array}{c|cc}
R & A_1 & A_2 \\
\hline
0 & 1 & \\
1 & 2 & \\
1 & 3 & \\
2 & 3 & \\
\end{array}
\quad
\begin{array}{c|cc}
X & B_1 & B_2 \\
\hline
 & 3 & 4 \\
\end{array}
\]

QUESTION: Do all repairs contain an \textit{RRX} path?

\begin{align*}
r_1 & R \quad R \quad X \quad 4 \\
0 \quad 1 \quad 3 \\
R \quad RRX \\
R \quad RRX \\
R \quad RRX
\end{align*}

\begin{align*}
r_2 & R \quad R \quad X \quad 4 \\
0 \quad 1 \quad 3 \\
R \quad RRX \\
R \quad RRX \\
R \quad RRX \quad RRX
\end{align*}
The notion of “rewinding”

INPUT to CERTAINTY(\textit{RRX}):

<table>
<thead>
<tr>
<th></th>
<th>(A_1)</th>
<th>(A_2)</th>
<th>(X)</th>
<th>(B_1)</th>
<th>(B_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

QUESTION: Do all repairs contain an \textit{RRX} path?

\(R\):

\[\begin{array}{ccc}
0 & R & 1 \\
1 & R & 3 \\
2 & X & 4 \\
\end{array}\]

\(R\) \textit{RRX} \(R\) \textit{RRX} \(R\) \textit{RRX}

\(X\)

\[\begin{array}{ccc}
0 & R & 1 \\
1 & R & 3 \\
3 & X & 4 \\
\end{array}\]

\(R\) \textit{RRX} \(R\) \textit{RRX} \(R\) \textit{RRX} \(R\) \textit{RRX} \(R\) \textit{RRX}
The notion of “rewinding”

INPUT to CERTAINTY($RRX$):

\[
\begin{array}{c|cc}
R & A_1 & A_2 \\
0 & 1 & . \\
1 & 2 & . \\
1 & 3 & . \\
2 & 3 & . \\
\end{array}
\]

\[
\begin{array}{c|cc}
X & B_1 & B_2 \\
2 & 3 & 4 \\
\end{array}
\]

QUESTION: Do all repairs contain an $RRX$ path?

\[
\begin{array}{c|cc}
r_1 & & \\
0 & R & 1 \\
1 & R & 3 \\
3 & X & 4 \\
\end{array}
\]

\[
\begin{array}{c|cc}
r_2 & & \\
0 & R & 1 \\
1 & R & 3 \\
3 & X & 4 \\
\end{array}
\]

$RRX$ $RRX$ $RRX$ $RRX$ $RRX$ $RRX$
Proposition

The following statements are equivalent:

1. db is a “yes”-instance for CERTAINTY(RRX); and
2. ∃c such that in all repairs, there exists a path of $RR \cdot R^* \cdot X$ starting at c.

“Reachability”, “NL-complete”  How to find the regular expression?
The notion of “rewinding” (cont.)

Proposition

The following statements are equivalent:

1. $db$ is a “yes”-instance for $\text{CERTAINTY}(RRX)$; and
2. $\exists c$ such that in all repairs, there exists a path of $RR \cdot R^* \cdot X$ starting at $c$.

“Reachability”, “$\text{NL}$-complete”
The notion of “rewinding” (cont.)

Proposition

The following statements are equivalent:

1. $db$ is a “yes”-instance for $\text{CERTAINTY}(RRX)$; and
2. $\exists c$ such that in all repairs, there exists a path of $RR \cdot R^* \cdot X$ starting at $c$.

“Reachability”, “$\text{NL}$-complete”

How to find the regular expression?
From path query to NFA

\[ \varepsilon \xrightarrow{R} R \xrightarrow{R} RR \xrightarrow{X} RRX \]

NFA(\(RRX\))
From path query to NFA

\[ \varepsilon \xrightarrow{R} R \xrightarrow{R} RR \xrightarrow{X} RRX \]

NFA(RRX)
From path query to NFA

ε → R → RR → RRX

NFA(RRX)
From path query to NFA

NFA(\text{RRX})
From path query to NFA

NFA(\textit{RRX})
From path query to NFA

NFA($RRX$)
From path query to NFA (cont.)

NFA($RXRRR$)
Outline

1. Handling self-joins
2. Classification result
3. Proof sketch
Our result

$$q_2 = RXRY$$
$$RXRXRY \in \text{NFA}(q_2)$$

C_1: \( q \) is a prefix of every word in NFA(\( q \))

$$q_1 = RXRX$$
$$RXRXRX \in \text{NFA}(q_1)$$

**NL-hard**
Our result

**coNP-complete**

\[ q_4 = RXRXRYRY \quad RXRXRYRXRYRY \in NFA(q_4) \]

**C_3:** \( q \) is a factor of every word in \( NFA(q) \)

**PTIME**

**NL-hard**

\[ q_2 = RXRY \quad RXRXRY \in NFA(q_2) \]

**C_1:** \( q \) is a prefix of every word in \( NFA(q) \)

**FO-rewritable**

\[ q_1 = RXRX \quad RXRXRX \in NFA(q_1) \]
Our result

\[ q_4 = RXRXRYRY \quad RXRXRYRXRYRY \in \text{NFA}(q_4) \]

**coNP-complete**

\[ q_4 = RXRXRYRY \quad RXRXRYRXRYRY \in \text{NFA}(q_4) \]

\[ q_3 = RXRYRY \]

**PTIME**

\[ q_3 = RXRYRY \]

\[ q_2 = RXRY \quad RXRXRY \in \text{NFA}(q_2) \]

**NL-hard**

\[ q_2 = RXRY \quad RXRXRY \in \text{NFA}(q_2) \]

\[ q_1 = RXRX \quad RXRXRX \in \text{NFA}(q_1) \]

**C_1:** \( q \) is a prefix of every word in \( \text{NFA}(q) \)

**C_2:** Whenever \( q = uRvRw \), \( q \) is a factor of \( uRvRvRw \); and whenever \( q = uRv_1Rv_2Rw \) for consecutive occurrences of \( R \), \( v_1 = v_2 \) or \( Rw \) is a prefix of \( Rv_1 \).

**C_3:** \( q \) is a factor of every word in \( \text{NFA}(q) \)

**FO-rewritable**

\[ q_1 = RXRX \quad RXRXRX \in \text{NFA}(q_1) \]
Our result

**coNP-complete**

\[ q_4 = RXRXRYRY \quad RXRXRYRXRYRY \in \text{NFA}(q_4) \]

**PTIME-complete**

\[ q_3 = RXRYRY \]

**C_3:** \( q \) is a factor of every word in NFA\((q)\)

**NL-complete**

\[ q_2 = RXRY \quad RXRXRY \in \text{NFA}(q_2) \]

**C_2:** Whenever \( q = uRvRw \), \( q \) is a factor of \( uRvRvRw \); and whenever \( q = uRv_1Rv_2Rw \) for consecutive occurrences of \( R \), \( v_1 = v_2 \) or \( Rw \) is a prefix of \( Rv_1 \).

**FO-rewritable**

\[ q_1 = RXRX \quad RXRXRX \in \text{NFA}(q_1) \]

**C_1:** \( q \) is a prefix of every word in NFA\((q)\)
$C_1, C_2$ and $C_3$ are decidable

$C_1$: $q$ is a prefix of every word in $\text{NFA}(q)$
\[
\iff \text{Whenever } q = u \cdot Rv \cdot Rw, \text{ } q \text{ is a prefix of } u \cdot Rv \cdot Rv \cdot Rw.
\]

$C_3$: $q$ is a factor of every word in $\text{NFA}(q)$
\[
\iff \text{Whenever } q = u \cdot Rv \cdot Rw, \text{ } q \text{ is a factor of } u \cdot Rv \cdot Rv \cdot Rw.
\]
Outline

1. Handling self-joins

2. Classification result

3. Proof sketch
Lemma \((\text{PTIME})\)

Let \(q\) be a path query satisfying \(C_3\). The following statements are equivalent:

1. \(\text{db}\) is a “yes”-instance for \(\text{CERTAINTY}(q)\); and
2. \(\exists c\) such that in all repairs, there exists a path accepted by \(\text{NFA}(q)\) starting at \(c\).

Moreover, item 2 can be decided in \(\text{PTIME}\) using dynamic programming/least fixedpoint logic.

\(C_3:\ q\) is a factor of every word in \(\text{NFA}(q)\)
Lemma (PTIME)

Let $q$ be a path query satisfying $C_3$. The following statements are equivalent:

1. $db$ is a “yes”-instance for $CERTAINTY(q)$; and
2. $\exists c$ such that in all repairs, there exists a path accepted by $NFA(q)$ starting at $c$.

Moreover, item 2 can be decided in $\text{PTIME}$ using dynamic programming/least fixedpoint logic.

$C_3$: $q$ is a factor of every word in $NFA(q)$. 
Lemma (**PTIME**)

Let $q$ be a path query satisfying $C_3$. The following statements are equivalent:

1. $db$ is a “yes”-instance for $\text{CERTAINTY}(q)$; and
2. $\exists c$ such that in all repairs, there exists a path accepted by NFA($q$) starting at $c$.

Moreover, item 2 can be decided in **PTIME** using dynamic programming/least fixedpoint logic.

$$C_3: \quad q \text{ is a factor of every word in } \text{NFA}(q)$$
Lemma (\textsc{PTIME})

Let \( q \) be a path query satisfying \( C_3 \). The following statements are equivalent:

1. \( db \) is a “yes”-instance for \( \text{CERTAINTY}(q) \); and
2. \( \exists c \) such that in all repairs, there exists a path accepted by \( \text{NFA}(q) \) starting at \( c \).

Moreover, item 2 can be decided in \textsc{PTIME} using dynamic programming/least fixedpoint logic.

\( C_3 \): \( q \) is a factor of every word in \( \text{NFA}(q) \)
Lemma (PTIME)

Let \( q \) be a path query satisfying \( C_3 \). The following statements are equivalent:

1. \( \text{db} \) is a “yes”-instance for \( \text{CERTAINTY}(q) \); and
2. \( \exists c \) such that in all repairs, there exists a path accepted by \( \text{NFA}(q) \) starting at \( c \).

Moreover, item 2 can be decided in **PTIME** using dynamic programming/least fixedpoint logic.

\[ C_3: \quad q \text{ is a factor of every word in } \text{NFA}(q) \]

\[ \text{NFA}(q) \]
Lemma (PTIME)

Let $q$ be a path query satisfying $C_3$. The following statements are equivalent:

1. $db$ is a “yes”-instance for $\text{CERTAINTY}(q)$; and
2. $\exists c$ such that in all repairs, there exists a path accepted by $\text{NFA}(q)$ starting at $c$.

Moreover, item 2 can be decided in PTIME using dynamic programming/least fixedpoint logic.

$C_3$: $q$ is a factor of every word in $\text{NFA}(q)$
C₁: q is a prefix of every word in NFA(q)
$C_1$: $q$ is a prefix of every word in $\text{NFA}(q)$
FO-rewritability

$C_1$: $q$ is a prefix of every word in $\text{NFA}(q)$

**Lemma (FO)**

Let $q$ be a path query satisfying $C_1$. The following statements are equivalent:

1. db is a “yes”-instance for $\text{CERTAINTY}(q)$; and
2. $\exists c$ such that in all repairs, there exists a path of $q$ starting at $c$.

Moreover, item 2 can be decided in $\text{FO}$. 
Lemma

Let $q$ be a path query. Then we have

- if $q$ does not satisfy $C_1$, then $\text{CERTAINTY}(q)$ is $\text{NL}$-hard (via Reachability);
- if $q$ does not satisfy $C_2$, then $\text{CERTAINTY}(q)$ is $\text{PTIME}$-hard (via Monotone Circuit Value); and
- if $q$ does not satisfy $C_3$, then $\text{CERTAINTY}(q)$ is $\text{coNP}$-hard (via Satisfiability).

$q = \begin{array}{c}
\bigwedge \\
\bigvee \\
\bigwedge \\
\bigvee \\
\bigvee \\
\end{array}
\begin{array}{c}
RX \\
\downarrow \\
RY \\
\downarrow \\
RY \\
\end{array}
\begin{array}{c}
u \\
\downarrow \\
Rv_1 \\
\downarrow \\
Rv_2 \\
\downarrow \\
Rw \\
\end{array}$

The output of $C$ is 0 iff $db$ contains a falsifying repair.
Future works

**Conjecture**

Let $q$ be a CQ. Then we have

- CERTAINTY($q$) is either in PTIME or coNP-complete, and it is decidable which of the two cases applies; and

- it is decidable whether or not CERTAINTY($q$) is in FO.

- Acyclic queries with self-joins
- Multiple key constraints, negated atom, aggregation etc.
### Conclusion

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>coNP-complete</strong></td>
<td>$q_4 = RXRXRYRY \quad RXRXRYRXRYRY \in NFA(q_4)$</td>
<td></td>
</tr>
<tr>
<td><strong>PTIME-complete</strong></td>
<td>$q_3 = RXRYRY$</td>
<td></td>
</tr>
<tr>
<td><strong>NL-complete</strong></td>
<td>$q_2 = RXRY \quad RXRXRY \in NFA(q_2)$</td>
<td></td>
</tr>
<tr>
<td><strong>FO-rewritable</strong></td>
<td>$q_1 = RXRX \quad RXRXRX \in NFA(q_1)$</td>
<td></td>
</tr>
</tbody>
</table>

C₃: $q$ is a factor of every word in $\text{NFA}(q)$

C₂: Whenever $q = uRvRw$, $q$ is a factor of $uRvRvRw$; and whenever $q = uRv_1Rv_2Rw$ for consecutive occurrences of $R$, $v_1 = v_2$ or $Rw$ is a prefix of $Rv_1$.

C₁: $q$ is a prefix of every word in $\text{NFA}(q)$