On First-Order Query Rewriting for Incomplete Database Histories

Alexandre Decan and Jef Wijsen

Université de Mons-Hainaut, Belgium

2008
Outline

1 Motivation

2 Problem Statement

3 Regularity of CERTAIN(\(w\))

4 FO-definability of CERTAIN(\(w\))

5 Words with variables

6 Conclusion
Plan

1 Motivation

2 Problem Statement

3 Regularity of CERTAIN($w$)

4 FO-definability of CERTAIN($w$)

5 Words with variables

6 Conclusion
Motivation

Notions

- An inconsistent database $db$ gives rise to a set of possible databases called repairs.

- **Consistent Query Answering (CQA)**
  - A Boolean query $q$ is *consistently true* on $db$ if for each repair $rep$ of $db$, $rep \models q$

- **CQA by FO query rewriting**
  - Given a Boolean query $q$, find $q'$ such that $q$ is *consistently true* on $db \iff db \models q'$
Database histories

- **Database history**
  - A finite sequence of databases

- **Example:**
  - `WorksFor` relation at four successive time points $t_0$, $t_1$, $t_2$, $t_3$

<table>
<thead>
<tr>
<th>Name</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ed</td>
<td>IBM</td>
</tr>
<tr>
<td>Ed</td>
<td>IBM</td>
</tr>
<tr>
<td>Ed</td>
<td>MS</td>
</tr>
<tr>
<td>Ed</td>
<td>MS</td>
</tr>
</tbody>
</table>
CQA on database histories under primary keys

Example

Did MS recruit an IBM employee?

\[ q = \text{WorksFor}(x, \text{IBM}) \land \text{\textcircled{}} \text{WorksFor}(x, \text{MS}) \]
CQA on database histories under primary keys

Example

Did MS recruit an IBM employee?

\[ q = \text{WorksFor}(x, \text{IBM}) \land \neg \text{WorksFor}(x, \text{MS}) \]

WorksFor relation with primary key Name violated at \( t_1 \) and \( t_2 \):

<table>
<thead>
<tr>
<th>( t_0 )</th>
<th>Name</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ed</td>
<td>IBM</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( t_1 )</th>
<th>Name</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ed</td>
<td>MS</td>
</tr>
<tr>
<td></td>
<td>Ed</td>
<td>IBM</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( t_2 )</th>
<th>Name</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ed</td>
<td>MS</td>
</tr>
<tr>
<td></td>
<td>Ed</td>
<td>IBM</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( t_3 )</th>
<th>Name</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ed</td>
<td>MS</td>
</tr>
<tr>
<td></td>
<td>Ed</td>
<td>MS</td>
</tr>
</tbody>
</table>
CQA on database histories under primary keys

A consistent FOTL-rewriting $q'$ of $q$:

$$q' = \varphi_{\text{IBM}}(x) \land (\varphi_{\{\text{IBM,MS}\}}(x) \text{ until } \varphi_{\text{MS}}(x))$$

where $\varphi_S(x)$ ensures that $S$ is the maximal set of values associated with the primary key $x$.

Example

$$\varphi_{\{\text{IBM,MS}\}} = \exists y (\text{WorksFor}(x, y) \land \forall y' (\text{WorksFor}(x, y') \rightarrow y' = \text{IBM} \lor y' = \text{MS}))$$

We know $\varphi_S$ from earlier work [Fuxman and Miller, 2005] [Wijsen, 2007].
Multiwords abstraction

- Employment history as a sequence:
  - $M_{Ed} = \langle \{IBM\}, \{IBM, MS\}, \{IBM, MS\}, \{MS\} \rangle$
- We call this a *multiword*. 
Multiwords abstraction

- Employment history as a sequence:
  \[ M_{Ed} = \langle \{IBM\}, \{IBM, MS\}, \{IBM, MS\}, \{MS\} \rangle \]
- We call this a *multiword*.
- Gives rise to four possible words:
  - \( \langle IBM, IBM, IBM, MS \rangle \)
  - \( \langle IBM, IBM, MS, MS \rangle \)
  - \( \langle IBM, MS, IBM, MS \rangle \)
  - \( \langle IBM, MS, MS, MS \rangle \)
- MS recruited an IBM employee because the word \( \langle IBM, MS \rangle \) is a subword of each of these four words.
Motivation

Definition of a multiword

Let $\Sigma$ be an alphabet.

**Definition**

A multiword is a sequence $M = \langle A_1, \ldots, A_n \rangle$ where for each $i \in \{1, \ldots, n\}$, $A_i \subseteq \Sigma$ and $A_i \neq \emptyset$.

The set of possible words:

$$\text{words}(M) = \{a_1 a_2 \ldots a_n \mid \forall i \in \{1, \ldots, n\} : a_i \in A_i\}$$

If $u, w$ are words, we write:

- $u \models w$ if $w$ is a subword of $u$.
- $M \models_{\text{certain}} w$ if for every $v \in \text{words}(M)$, $v \models w$. 

Alexandre Decan and Jef Wijsen (UMH)

On First-Order Query Rewriting

2008 9 / 26
Let $w = abdabcab$.
Consider the following multiword $M$:

$$M = \langle a, b, d, a, b, c, a, \{a, b\}, b, d, a, b, \{c, d\}, a, b, c, a, b \rangle$$

Hence, $M \models certain ab dab cab$. 

Alexandre Decan and Jef Wijsen (UMH) On First-Order Query Rewriting 2008 10 / 26
Example

Let $w = abdabcab$.
Consider the following multiword $M$:

$$M = \langle a, b, d, a, b, c, a, \{a, b\}, b, d, a, b, \{c, d\}, a, b, c, a, b \rangle$$

$$\text{words}(M) = \{ \text{abdabca} \text{abdabcabcab}, \text{abdabca} \text{abdabdabcab}, \text{abdabcab} \text{babcabcab}, \text{abdabcab} \text{babcabcab} \}$$

Hence, $M \models_{\text{certain}} abdabcab$. 
Plan

1. Motivation
2. Problem Statement
3. Regularity of $\text{CERTAIN}(w)$
4. FO-definability of $\text{CERTAIN}(w)$
5. Words with variables
6. Conclusion
Problem: \textsc{CERTAIN}(w)

We are interested in (the complexity of) the following language:

- \textsc{CERTAIN}(w) := \{ M \text{ a multiword} \mid M \models_{\text{certain}} w \}

In particular, given \( w \), is \textsc{CERTAIN}(w) \text{ FO-definable}?
Plan

1. Motivation
2. Problem Statement
3. Regularity of \textsc{CERTAIN}(w)
4. FO-definability of \textsc{CERTAIN}(w)
5. Words with variables
6. Conclusion
Regularity of \textit{CERTAIN}(w)

- Simple algorithm for deciding whether \( M \in \text{CERTAIN}(w) \).
- Example with \( w = a \cdot b \cdot b \) and \( M = \langle a, b, \{a, b\}, \{a, b\}, b, b \rangle \):

\[
\begin{array}{c|c|c}
M & \text{Concat} & \text{Prefixes} \\
\hline
& & \{\varepsilon\} \\
\end{array}
\]
Regularity of $\text{CERTAIN}(w)$

- Simple algorithm for deciding whether $M \in \text{CERTAIN}(w)$.
- Example with $w = abb$ and $M = \langle a, b, \{a, b\}, \{a, b\}, b, b \rangle$:

<table>
<thead>
<tr>
<th>M</th>
<th>Concat</th>
<th>Prefixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>${\epsilon \cdot a}$</td>
<td>${\epsilon}$</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td></td>
<td>${a}$</td>
</tr>
</tbody>
</table>
Regularity of $\text{CERTAIN}(w)$

- Simple algorithm for deciding whether $M \in \text{CERTAIN}(w)$.
- Example with $w = abb$ and $M = \langle a, b, \{a, b\}, \{a, b\}, b, b \rangle$:

<table>
<thead>
<tr>
<th>$M$</th>
<th>Concated</th>
<th>Prefixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>$\epsilon \cdot a$</td>
<td>${a}$</td>
</tr>
<tr>
<td>$b$</td>
<td>$a \cdot b$</td>
<td>${ab}$</td>
</tr>
</tbody>
</table>
Regularity of $\text{CERTAIN}(w)$

- Simple algorithm for deciding whether $M \in \text{CERTAIN}(w)$.
- Example with $w = abb$ and $M = \langle a, b, \{a, b\}, \{a, b\}, b, b \rangle$:

<table>
<thead>
<tr>
<th>M</th>
<th>Concat</th>
<th>Prefixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>${\varepsilon \cdot a}$</td>
<td>${\varepsilon}$</td>
</tr>
<tr>
<td>$b$</td>
<td>${a \cdot b}$</td>
<td>${a}$</td>
</tr>
<tr>
<td>${a, b}$</td>
<td>${a b \cdot a, a b \cdot b}$</td>
<td>${a}$</td>
</tr>
</tbody>
</table>
Regularity of $\text{CERTAIN}(w)$

- Simple algorithm for deciding whether $M \in \text{CERTAIN}(w)$.
- Example with $w = abb$ and $M = \langle a, b, \{a, b\}, \{a, b\}, b, b \rangle$:

<table>
<thead>
<tr>
<th>$M$</th>
<th>Concat</th>
<th>Prefixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>${\varepsilon \cdot a}$</td>
<td>${\varepsilon}$</td>
</tr>
<tr>
<td>$b$</td>
<td>${a \cdot b}$</td>
<td>${a}$</td>
</tr>
<tr>
<td>${a, b}$</td>
<td>${ab \cdot a, ab \cdot b}$</td>
<td>${ab}$</td>
</tr>
<tr>
<td>${a, b}$</td>
<td>${a \cdot a, a \cdot b}$</td>
<td>${a, ab}$</td>
</tr>
</tbody>
</table>
Regularity of $\text{CERTAIN}(w)$

- Simple algorithm for deciding whether $M \in \text{CERTAIN}(w)$.
- Example with $w = abb$ and $M = \langle a, b, \{a, b\}, \{a, b\}, b, b \rangle$:

<table>
<thead>
<tr>
<th>$M$</th>
<th>Concat</th>
<th>Prefixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>${\varepsilon \cdot a}$</td>
<td>${\varepsilon}$</td>
</tr>
<tr>
<td>$b$</td>
<td>${a \cdot b}$</td>
<td>${a}$</td>
</tr>
<tr>
<td>${a, b}$</td>
<td>${ab \cdot a, ab \cdot b}$</td>
<td>${ab}$</td>
</tr>
<tr>
<td>${a, b}$</td>
<td>${a \cdot a, a \cdot b}$</td>
<td>${a, ab}$</td>
</tr>
<tr>
<td>$b$</td>
<td>${a \cdot b, ab \cdot b}$</td>
<td>${ab}$</td>
</tr>
</tbody>
</table>

This algorithm leads to the construction of a DFA.
Regularity of $\text{CERTAIN}(w)$

- Simple algorithm for deciding whether $M \in \text{CERTAIN}(w)$.
- Example with $w = abb$ and $M = \langle a, b, \{a, b\}, \{a, b\}, b, b \rangle$:

<table>
<thead>
<tr>
<th>$M$</th>
<th>Concat</th>
<th>Prefixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>${\varepsilon \cdot a}$</td>
<td>${\varepsilon}$</td>
</tr>
<tr>
<td>$b$</td>
<td>${a \cdot b}$</td>
<td>${a}$</td>
</tr>
<tr>
<td>${a, b}$</td>
<td>${ab \cdot a, ab \cdot b}$</td>
<td>${ab}$</td>
</tr>
<tr>
<td>${a, b}$</td>
<td>${a \cdot a, a \cdot b}$</td>
<td>${a, ab}$</td>
</tr>
<tr>
<td>$b$</td>
<td>${a \cdot b, ab \cdot b}$</td>
<td>${ab}$</td>
</tr>
<tr>
<td>$b$</td>
<td>${ab \cdot b}$</td>
<td>${}$</td>
</tr>
</tbody>
</table>
Regularity of $\text{CERTAIN}(w)$

- Simple algorithm for deciding whether $M \in \text{CERTAIN}(w)$.
- Example with $w = abb$ and $M = \langle a, b, \{a, b\}, \{a, b\}, b, b \rangle$:

\[
\begin{array}{ccc}
M & \text{Concat} & \text{Prefixes} \\
\hline
a & \{\varepsilon \cdot a\} & \{\varepsilon\} \\
b & \{a \cdot b\} & \{a\} \\
\{a, b\} & \{ab \cdot a, ab \cdot b\} & \{a\} \\
\{a, b\} & \{a \cdot a, a \cdot b\} & \{a, ab\} \\
b & \{a \cdot b, ab \cdot b\} & \{ab\} \\
b & \{ab \cdot b\} & \{\varepsilon\} \\
\end{array}
\]

- $M \models_{\text{CERTAIN}} w$ if and only if this sequence ends with $\{\}$. 
- This algorithm leads to the construction of a DFA.
Finite State Automaton

Example with $w = abb$ and $\Sigma = \{a, b\}$:
Monadic Second Order

- $\text{CERTAIN}(w)$ can be described in MSO (thanks to Véronique Bruyère).
- Example with $w = abb$ and $\Sigma = \{a, b\}$:

$$
\forall X_a \forall X_b \left( \begin{array}{l}
\forall y (X_a(y) \lor X_b(y)) \\
\land \\
\lor \exists y (X_a(y) \land X_b(y)) \\
\land \\
\forall y (X_a(y) \implies P_{\{a\}}(y) \lor P_{\{a, b\}}(y)) \\
\land \\
\forall y (X_b(y) \implies P_{\{b\}}(y) \lor P_{\{a, b\}}(y)) \\
\end{array} \right)
$$

$$
\implies \exists w_1 \exists w_2 \exists w_3 \left( \begin{array}{l}
S(w_1, w_2) \land S(w_2, w_3) \\
\land \\
X_a(w_1) \land X_b(w_2) \land X_b(w_3) \\
\end{array} \right)
$$

$\text{CERTAIN}(w)$ is regular.
Plan

1. Motivation
2. Problem Statement
3. Regularity of CERTAIN(\(w\))
4. FO-definability of CERTAIN(\(w\))
5. Words with variables
6. Conclusion
Recall: results on standard words

Fundamental results on standard words:

- Over words, linear temporal logic has the same expressive power as FO [Gabbay et al., 1980] [Kamp, 1968].
- A regular language is FO-definable if and only if it is aperiodic [McNaughton and Papert, 1971] [Schützenberger, 1965].
Aperiodicity of \text{CERTAIN}(w)

These notions are equivalent:

\text{FO-definability} \iff \text{Star-freeness} \iff \text{Aperiodicity}

Definition (aperiodicity)

\exists k \geq 0 \text{ such that for all multiwords } P, U, Q:

\[ P \cdot U^k \cdot Q \in \text{CERTAIN}(w) \text{ if and only if } P \cdot U^{k+1} \cdot Q \in \text{CERTAIN}(w). \]
Sufficient condition for aperiodicity

Theorem

If the first (or the last) symbol of $w$ occurs only once in $w$, then $\text{CERTAIN}(w)$ is aperiodic (and hence FO-definable).
Sufficient condition for aperiodicity

**Theorem**

*If the first (or the last) symbol of $w$ occurs only once in $w$, then $\text{CERTAIN}(w)$ is aperiodic (and hence FO-definable).*

$\text{CERTAIN}(aa)$ is also FO-definable:

$$\exists i \exists j (\text{Successor}(i,j) \land P_{\{a\}}(i) \land P_{\{a\}}(j))$$
Words with variables

- A v-word $v$ is a word in which variables can appear.
- Variables are placeholders for constants.
- $\models_{\text{certain}}$ naturally carries over the v-words.
- For example, $\langle a, \{a, b\}, \{a, b\}, a \rangle \models_{\text{certain}} xx$
  - aaaa
  - aaba
  - abaa
  - abba
- Motivation: larger class of queries
  - Eg.: "Did someone work in the same company at two consecutive times?"
v-words: results

**Theorem** (not in our paper, thanks to Véronique Bruyère)

*If* $v$ *is a v-word, then* $\text{CERTAIN}(v)$ *is regular.*
v-words: results

**Theorem** (not in our paper, thanks to Véronique Bruyère)

*If* $v$ *is a v-word, then* $\text{CERTAIN}(v)$ *is regular.*

**Theorem**

*There exists some v-word $v$ such that:*

1. $\text{CERTAIN}(v)$ *is in* $\text{PTIME};$
2. $\text{CERTAIN}(v)$ *is not FO-definable.*

Example: $xx.$
Plan

1. Motivation
2. Problem Statement
3. Regularity of $\text{CERTAIN}(w)$
4. FO-definability of $\text{CERTAIN}(w)$
5. Words with variables
6. Conclusion
The study of multiwords is motivated by FOTL query rewriting under primary key constraints.

CERTAIN(\nu) is defined as the set of multiwords that certainly contain the "pattern" \nu.
The study of multiwords is motivated by FOTL query rewriting under primary key constraints.

CERTAIN($\nu$) is defined as the set of multiwords that certainly contain the "pattern" $\nu$.

We showed the following results:

- CERTAIN($\nu$) is always regular.
- CERTAIN(xx) is regular but not FO-definable.
- If $w$ is variable-free and the first (or the last) symbol of $w$ occurs only once, then CERTAIN($w$) is aperiodic.
Questions

We are currently addressing the following tasks:

1. Show the conjecture that \( \text{CERTAIN}(w) \) is FO-definable if \( w \) is variable-free.

2. Find a syntactic characterization of the \( v \)-words \( v \) for which \( \text{CERTAIN}(v) \) is FO-definable.

Second question on page 12 has been solved.