# On First-Order Query Rewriting for Incomplete Database Histories

Alexandre Decan and Jef Wijsen

Université de Mons-Hainaut, Belgium

2008

# Outline

# Plan

# 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** $\Longleftrightarrow$ **db** $\models q'$

# Database histories

- *Database history*
  - A finite sequence of databases
- Example:
  - WorksFor relation at four successive time points $t_0, t_1, t_2, t_3$

$t_0$

| Name | Company |
|---|---|
| Ed | IBM |

$t_1$

| Name | Company |
|---|---|
| Ed | IBM |

$t_2$

| Name | Company |
|---|---|
| Ed | MS |

$t_3$

| Name | Company |
|---|---|
| Ed | MS |

# CQA on database histories under primary keys

Example

*Did MS recruit an IBM employee?*

$$q = \mathsf{WorksFor}(\underline{x}, \mathrm{IBM}) \land \bigcirc \mathsf{WorksFor}(\underline{x}, \mathrm{MS})$$

# CQA on database histories under primary keys

## Example

*Did MS recruit an IBM employee?*

$$q \quad = \quad \mathsf{WorksFor}(\underline{x}, \mathrm{IBM}) \land \bigcirc \mathsf{WorksFor}(\underline{x}, \mathrm{MS})$$

WorksFor relation with primary key Name violated at $t_1$ and $t_2$:

$t_0$

| Name | Company |
|---|---|
| Ed | IBM |

$t_1$

| Name | Company |
|---|---|
| Ed | MS |
| Ed | IBM |

$t_2$

| Name | Company |
|---|---|
| Ed | MS |
| Ed | IBM |

$t_3$

| Name | Company |
|---|---|
| Ed | MS |

# CQA on database histories under primary keys

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

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

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

## Example

$$\varphi_{\{\text{IBM},\text{MS}\}} = \exists y(\text{WorksFor}(\underline{x}, y) \wedge \forall y'(\text{WorksFor}(\underline{x}, y') \rightarrow y' = \text{IBM} \vee y' = \text{MS}))$$

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

# Multiwords abstraction

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

# Multiwords abstraction

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

# 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:

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

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

- $u \Vdash w$ if $w$ is a subword of $u$.
- $M \Vdash_{\mathsf{certain}} w$ if for every $v \in \mathsf{words}(M)$, $v \Vdash w$.

# Example

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

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

# Example

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

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

$$\mathsf{words}(M) = \{ \begin{array}{l} \texttt{abdabca\underline{abdabcab}cab}, \\ \texttt{abdabcaabd\underline{abdabcab}}, \\ \texttt{\underline{abdabcab}bdabcabcab}, \\ \texttt{\underline{abdabcab}bd\underline{abdabcab}} \end{array} \}$$

Hence, $M \Vdash_{\mathsf{certain}} abdabcab$.

# Plan

# Problem: $\mathsf{CERTAIN}(w)$

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

- $\mathsf{CERTAIN}(w) := \{M \text{ a multiword} \mid M \Vdash_{\mathsf{certain}} w\}$

In particular, given $w$, is $\mathsf{CERTAIN}(w)$ FO-definable?

# Plan

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

# Regularity of $\mathsf{CERTAIN}(w)$

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

| M | Concat | Prefixes |
|---|---|---|
| | | $\{\varepsilon\}$ |

# Regularity of CERTAIN($w$)

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

| M | Concat | Prefixes |
|---|---|---|
| | | $\{\varepsilon\}$ |
| $a$ | $\{\varepsilon \cdot a\}$ | $\{a\}$ |

# Regularity of $\mathsf{CERTAIN}(w)$

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

| M | Concat | Prefixes |
|---|---|---|
| | | $\{\varepsilon\}$ |
| $a$ | $\{\varepsilon \cdot a\}$ | $\{a\}$ |
| $b$ | $\{a \cdot b\}$ | $\{ab\}$ |

# Regularity of $\mathsf{CERTAIN}(w)$

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

| M | Concat | Prefixes |
|---|---|---|
| | | $\{\varepsilon\}$ |
| $a$ | $\{\varepsilon \cdot a\}$ | $\{a\}$ |
| $b$ | $\{a \cdot b\}$ | $\{ab\}$ |
| $\{a, b\}$ | $\{ab \cdot a, ab \cdot b\}$ | $\{a\}$ |

# Regularity of CERTAIN($w$)

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

| M | Concat | Prefixes |
|---|---|---|
| | | $\{\varepsilon\}$ |
| $a$ | $\{\varepsilon \cdot a\}$ | $\{a\}$ |
| $b$ | $\{a \cdot b\}$ | $\{ab\}$ |
| $\{a, b\}$ | $\{ab \cdot a, ab \cdot b\}$ | $\{a\}$ |
| $\{a, b\}$ | $\{a \cdot a, a \cdot b\}$ | $\{a, ab\}$ |

# Regularity of $\mathsf{CERTAIN}(w)$

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

| M | Concat | Prefixes |
|---|---|---|
| | | $\{\varepsilon\}$ |
| $a$ | $\{\varepsilon \cdot a\}$ | $\{a\}$ |
| $b$ | $\{a \cdot b\}$ | $\{ab\}$ |
| $\{a, b\}$ | $\{ab \cdot a, ab \cdot b\}$ | $\{a\}$ |
| $\{a, b\}$ | $\{a \cdot a, a \cdot b\}$ | $\{a, ab\}$ |
| $b$ | $\{a \cdot b, ab \cdot b\}$ | $\{ab\}$ |

# Regularity of CERTAIN($w$)

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

| M | Concat | Prefixes |
|---|---|---|
| | | $\{\varepsilon\}$ |
| $a$ | $\{\varepsilon \cdot a\}$ | $\{a\}$ |
| $b$ | $\{a \cdot b\}$ | $\{ab\}$ |
| $\{a, b\}$ | $\{ab \cdot a, ab \cdot b\}$ | $\{a\}$ |
| $\{a, b\}$ | $\{a \cdot a, a \cdot b\}$ | $\{a, ab\}$ |
| $b$ | $\{a \cdot b, ab \cdot b\}$ | $\{ab\}$ |
| $b$ | $\{ab \cdot b\}$ | $\{\}$ |

# Regularity of CERTAIN($w$)

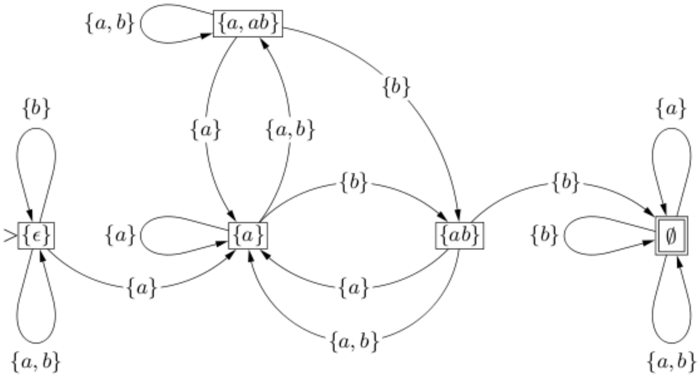
- Simple algorithm for deciding whether $M \in \mathsf{CERTAIN}(w)$.
- Example with $w = abb$ and $M = \langle a, b, \{a, b\}, \{a, b\}, b, b\rangle$:

| M | Concat | Prefixes |
|---|---|---|
| | | $\{\varepsilon\}$ |
| $a$ | $\{\varepsilon \cdot a\}$ | $\{a\}$ |
| $b$ | $\{a \cdot b\}$ | $\{ab\}$ |
| $\{a, b\}$ | $\{ab \cdot a, ab \cdot b\}$ | $\{a\}$ |
| $\{a, b\}$ | $\{a \cdot a, a \cdot b\}$ | $\{a, ab\}$ |
| $b$ | $\{a \cdot b, ab \cdot b\}$ | $\{ab\}$ |
| $b$ | $\{ab \cdot b\}$ | $\{\}$ |

- $M \Vdash_{\mathsf{certain}} w$ 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

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

$$\forall X_a \forall X_b \left[ \left( \begin{array}{ll} & \forall y (X_a(y) \vee X_b(y)) \\ \wedge & \neg \exists y (X_a(y) \wedge X_b(y)) \\ \wedge & \forall y (X_a(y) \implies P_{\{a\}}(y) \vee P_{\{a,b\}}(y)) \\ \wedge & \forall y (X_b(y) \implies P_{\{b\}}(y) \vee P_{\{a,b\}}(y)) \end{array} \right) \right.$$

$$\left. \implies \exists w_1 \exists w_2 \exists w_3 \left( \begin{array}{ll} & S(w_1, w_2) \wedge S(w_2, w_3) \\ \wedge & X_a(w_1) \wedge X_b(w_2) \wedge X_b(w_3) \end{array} \right) \right]$$

$\mathsf{CERTAIN}(w)$ is regular.

# Plan

# 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 $\mathsf{CERTAIN}(w)$

- These notions are equivalent:

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

## Definition (aperiodicity)

$\exists k \geq 0$ such that for all multiwords $P, U, Q$:

$$P \cdot U^k \cdot Q \in \mathsf{CERTAIN}(w) \text{ if and only if } P \cdot U^{k+1} \cdot Q \in \mathsf{CERTAIN}(w).$$

# Sufficient condition for aperiodicity

**Theorem**

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

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

$$\exists i \exists j (Successor(i,j) \wedge P_{\{a\}}(i) \wedge P_{\{a\}}(j))$$

# Plan

# Words with variables

- A *v-word* $v$ is a word in which variables can appear.
- Variables are placeholders for constants.
- $\Vdash_{\mathsf{certain}}$ naturally carries over the v-words.
- For example, $\langle a, \{a, b\}, \{a, b\}, a \rangle \Vdash_{\mathsf{certain}} xx$
  - <u>aa</u>aa
  - <u>aa</u>ba
  - ab<u>aa</u>
  - a<u>bb</u>a
- 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* $\mathsf{CERTAIN}(v)$ *is regular.*

# v-words: results

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

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

**Theorem**

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

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

Example: $xx$.

# Plan

# Conclusion

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

# Conclusion

- The study of multiwords is motivated by FOTL query rewriting under primary key constraints.
- $\mathsf{CERTAIN}(v)$ is defined as the set of multiwords that certainly contain the "pattern" $v$.
- We showed the following results:
  - $\mathsf{CERTAIN}(v)$ is always regular.
  - $\mathsf{CERTAIN}(xx)$ is regular but not FO-definable.
  - If $w$ is variable-free and the first (or the last) symbol of $w$ occurs only once, then $\mathsf{CERTAIN}(w)$ is aperiodic.

# Questions

We are currently addressing the following tasks:

1. Show the conjecture that $\mathsf{CERTAIN}(w)$ is FO-definable if $w$ is variable-free.
2. Find a syntactic characterization of the v-words $v$ for which $\mathsf{CERTAIN}(v)$ is FO-definable.

Second question on page 12 has been solved.