Temporal Query Answering in DL-Lite over Inconsistent Data

Camille Bourgaux$^{1,2}$ & Anni-Yasmin Turhan$^1$

$^1$Technische Universität Dresden, $^2$Télécom ParisTech

16$^{th}$ International Semantic Web Conference
October 24$^{th}$ 2017, Vienna
Ontology-based query answering for situation recognition

Situation recognition for context-aware systems

Ontology-based query answering

TBox (Ontology) $\mathcal{T}$
General knowledge about the domain (how the system works...)

ABox (Data) $\mathcal{A}$
Specific knowledge (snapshot of the system)

Conjunctive Query $q$
Description of the situation

$\langle \mathcal{T}, \mathcal{A} \rangle \models q$ ?
situation recognition

Ex: list of servers almost overloaded
Ontology-based query answering for situation recognition

Ontology-based query answering

<table>
<thead>
<tr>
<th>TBox</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebServer ⊑ Server,</td>
</tr>
<tr>
<td>AppServer ⊑ Server,</td>
</tr>
<tr>
<td>WebServer ⊑ ¬AppServer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ABox</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute(a, b), WebServer(a)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conjunctive Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>∃y Server(x) ∧ Execute(x, y)</td>
</tr>
</tbody>
</table>

⟨T, A⟩ |= q ?
Ontology-based query answering for situation recognition

Inconsistency-tolerant query answering

<table>
<thead>
<tr>
<th>TBox</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebServer ⊑ Server,</td>
<td>web servers are servers</td>
</tr>
<tr>
<td>AppServer ⊑ Server,</td>
<td>application servers are servers</td>
</tr>
<tr>
<td>WebServer ⊑ ¬AppServer</td>
<td>web and application servers are disjoint</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ABox</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute(a, b), WebServer(a), AppServer(a)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conjunctive Query</th>
<th>q</th>
</tr>
</thead>
<tbody>
<tr>
<td>∃yServer(x) ∧ Execute(x, y)</td>
<td>retrieve servers that execute something</td>
</tr>
</tbody>
</table>

⟨T, A⟩ |= q?
## Ontology-based query answering for situation recognition

### Temporal query answering

#### TBox

<table>
<thead>
<tr>
<th>WebServer ⊑ Server,</th>
<th>web servers are servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppServer ⊑ Server,</td>
<td>application servers are servers</td>
</tr>
<tr>
<td>WebServer ⊑ ¬AppServer</td>
<td>web and application servers are disjoint</td>
</tr>
</tbody>
</table>

#### Sequence of ABoxes

<table>
<thead>
<tr>
<th>$\mathcal{A}_1$ : Execute($a$, $b$), WebServer($a$)</th>
<th>data at time point 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{A}_2$ : Execute($a$, $c$), WebServer($a$)</td>
<td>data at time point 2</td>
</tr>
</tbody>
</table>

#### Temporal Conjunctive Query

$\text{Server}(x) \land \square \neg \exists y \text{Execute}(x, y)$ retrieve servers that always executed something

$$\langle \mathcal{T}, (\mathcal{A}_i)_{1 \leq i \leq 2} \rangle, 2 \models \phi ?$$ answer the query at time point 2
Goal: inconsistency-tolerant temporal query answering
Outline

1. Introduction
2. Preliminary notions
3. Temporal query answering over inconsistent data
4. Complexity analysis
5. Conclusion and perspectives
Inconsistency-tolerant semantics for knowledge bases

**Repair**

$\subseteq$-maximal subset of the ABox consistent with the TBox

<table>
<thead>
<tr>
<th>TBox</th>
<th>$\mathcal{T}$</th>
<th>ABox</th>
<th>$\mathcal{A}$</th>
<th>Repair 1</th>
<th>Repair 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebServer $\sqsubseteq$ Server, AppServer $\sqsubseteq$ Server, WebServer $\sqsubseteq$ ¬AppServer</td>
<td>WebServer($a$), AppServer($a$), Execute($a$, $b$)</td>
<td>WebServer($a$), Execute($a$, $b$)</td>
<td>AppServer($a$), Execute($a$, $b$)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Inconsistency-tolerant semantics for knowledge bases

**Repair**

\(\subseteq\)-maximal subset of the ABox consistent with the TBox

**AR semantics (ABox Repair)**

AR answer \(\iff\) answer in every repair

<table>
<thead>
<tr>
<th>TBox</th>
<th>ABox</th>
<th>Repair 1</th>
<th>Repair 2</th>
</tr>
</thead>
</table>
| WebServer \(\sqsubseteq\) Server,  
AppServer \(\sqsubseteq\) Server,  
WebServer \(\sqsubseteq\) \neg\ AppServer | WebServer\((a)\),  
AppServer\((a)\),  
Execute\((a, b)\)  | WebServer\((a)\),  
Execute\((a, b)\) | AppServer\((a)\),  
Execute\((a, b)\) |

**True under AR semantics**

- Execute\((a, b)\)
- Server\((a)\)

**False under AR semantics**

- AppServer\((a)\)
- WebServer\((a)\)
Inconsistency-tolerant semantics for knowledge bases

Repair

\( \subseteq \)-maximal subset of the ABox consistent with the TBox

IAR semantics (Intersection ABox Repair)

IAR answer \( \Leftrightarrow \) answer in the intersection of all repairs

<table>
<thead>
<tr>
<th>TBox</th>
<th>ABox</th>
<th>Repair 1</th>
<th>Repair 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebServer ( \sqsubseteq ) Server, AppServer ( \sqsubseteq ) Server, WebServer ( \sqsubseteq ) ( \neg )AppServer</td>
<td>WebServer( (a) ), AppServer( (a) ), Execute( (a, b) )</td>
<td>WebServer( (a) ), Execute( (a,b) )</td>
<td>AppServer( (a) ), Execute( (a,b) )</td>
</tr>
</tbody>
</table>

True under IAR semantics

Execute\( (a, b) \)

False under IAR semantics

Server\( (a) \)
AppServer\( (a) \)
WebServer\( (a) \)
Inconsistency-tolerant semantics for knowledge bases

Repair

\( \subseteq \)-maximal subset of the ABox consistent with the TBox

Brave semantics

brave answer \iff answer in some repair

<table>
<thead>
<tr>
<th>TBox</th>
<th>( T )</th>
<th>ABox</th>
<th>( A )</th>
<th>Repair 1</th>
<th>Repair 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebServer ( \sqsubseteq ) Server, AppServer ( \sqsubseteq ) Server, WebServer ( \sqsubseteq ) \neg \text{AppServer}</td>
<td>WebServer(( a )), AppServer(( a )), Execute(( a, b ))</td>
<td>WebServer(( a )), Execute(( a, b ))</td>
<td>WebServer(( a )), Execute(( a, b ))</td>
<td>AppServer(( a )), Execute(( a, b ))</td>
<td></td>
</tr>
</tbody>
</table>

True under brave semantics

Execute(\( a, b \))
Server(\( a \))
AppServer(\( a \))
WebServer(\( a \))

False under brave semantics
Inconsistency-tolerant semantics

Connection to context awareness

\[ \text{IAR-answers} \subseteq \text{AR-answers} \subseteq \text{brave-answers} \]

- **AR**: more natural, widely accepted
- **IAR**: for situations to be recognised with very high confidence
  - Example: “the server is not used”
- **brave**: for critical situations that have to be handled
  - Example: “the server is almost overloaded and runs a process that has an increasing workload”
Temporal query answering

- **Temporal knowledge base (TKB)** $\langle T, (A_i)_{0 \leq i \leq n} \rangle$
  - **global TBox**: domain knowledge holds eternally
  - **sequence of ABoxes**: data at different time points

- **Rigid** predicates: interpretations are not allowed to change over time

- **Temporal conjunctive query (TCQ)**: CQs + LTL operators

  **Example:**
  \[\phi_1 = \Box \neg \text{AlmostOverloaded}(x): \text{was almost overloaded at previous time point}\]
  \[\phi_2 = \Diamond \neg (\text{Critical}(x) \land \Box \neg \Diamond \neg \text{Critical}(x)): \text{has been in a critical situation twice}\]

  \text{Critical}(x) := \exists y \text{Execute}(x, y) \land \text{IncreasingWorkload}(y) \land \text{AlmostOverloaded}(x)\]
1 Introduction

2 Preliminary notions

3 Temporal query answering over inconsistent data

4 Complexity analysis

5 Conclusion and perspectives
Repairs of a temporal knowledge base

Temporal behaviour of predicates

---

### Repair of a temporal knowledge base

“maximal subset of the data consistent with the TBox”

sequence of ABoxes $\approx$ set of **timed-assertions**: $\{(\alpha, i) \mid \alpha \in \mathcal{A}_i\}$

<table>
<thead>
<tr>
<th>$\mathcal{T}$</th>
<th>$\mathcal{A}_1$ (time point 1)</th>
<th>$\mathcal{A}_2$ (time point 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebServer $\sqsubseteq$ Server, AppServer $\sqsubseteq$ Server, WebServer $\sqsubseteq \neg$ AppServer</td>
<td>WebServer($a$), Execute($a$, $b$)</td>
<td>WebServer($a$), AppServer($a$), Execute($a$, $c$)</td>
</tr>
</tbody>
</table>
## Repairs of a temporal knowledge base
### Temporal behaviour of predicates

<table>
<thead>
<tr>
<th>$\mathcal{T}$</th>
<th>$A_1$ (time point 1)</th>
<th>$A_2$ (time point 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebServer $\sqsubseteq$ Server, AppServer $\sqsubseteq$ Server, WebServer $\sqsubseteq \neg$ AppServer</td>
<td>WebServer($a$), Execute($a$, $b$)</td>
<td>WebServer($a$), AppServer($a$), Execute($a$, $c$)</td>
</tr>
</tbody>
</table>

**First case:** independent time points, no rigid predicates

→ contradictions only between timed-assertions with the same time point
Repairs of a temporal knowledge base
Temporal behaviour of predicates

First case: independent time points, no rigid predicates
→ contradictions only between timed-assertions with the same time point
→ repairs of TKB = sequences of repairs of KBs
Repairs of a temporal knowledge base
Temporal behaviour of predicates

<table>
<thead>
<tr>
<th>T</th>
<th>A₁ (time point 1)</th>
<th>A₂ (time point 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebServer ⊆ Server, AppServer ⊆ Server, WebServer ⊆ ¬AppServer</td>
<td>WebServer(a), Execute(a, b)</td>
<td>WebServer(a), AppServer(a), Execute(a, c)</td>
</tr>
</tbody>
</table>

Second case: some rigid predicates are not allowed to change over time
Repairs of a temporal knowledge base
Temporal behaviour of predicates

<table>
<thead>
<tr>
<th>T</th>
<th>A₁ (time point 1)</th>
<th>A₂ (time point 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebServer ⊑ Server, AppServer ⊑ Server, WebServer ⊑ ¬AppServer</td>
<td>WebServer(a), Execute(a, b)</td>
<td>WebServer(a), AppServer(a), Execute(a, c)</td>
</tr>
</tbody>
</table>

**Second case:** some rigid predicates are not allowed to change over time
→ contradictions between data at different time points possible
Repairs of a temporal knowledge base
Temporal behaviour of predicates

\[ T \]
- WebServer \( \sqsubseteq \) Server,
- AppServer \( \sqsubseteq \) Server,
- WebServer \( \sqsubseteq \neg \) AppServer

\[ A_1 \) (time point 1)\]
- WebServer\( (a) \),
- Execute\( (a, b) \)

\[ A_2 \) (time point 2)\]
- WebServer\( (a) \),
- AppServer\( (a) \),
- Execute\( (a, c) \)

Second case: some rigid predicates are not allowed to change over time
→ contradictions between data at different time points possible
→ repairs of TKB \( \neq \) sequences of repairs of KBs

\[ A_1' \]
- WebServer\( (a) \),
- Execute\( (a, b) \)

\[ A_2' \]
- WebServer\( (a) \),
- Execute\( (a, c) \)

\[ A_1'' \]
- Execute\( (a, b) \)

\[ A_2'' \]
- AppServer\( (a) \),
- Execute\( (a, c) \)
AR, IAR and brave semantics defined in the natural way

<table>
<thead>
<tr>
<th>AR semantics</th>
<th>AR answer ⟷ answer in every repair</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAR semantics</td>
<td>IAR answer ⟷ answer in the intersection of all repairs</td>
</tr>
<tr>
<td>Brave semantics</td>
<td>brave answer ⟷ answer in some repair</td>
</tr>
</tbody>
</table>
Temporal query answering over inconsistent data

Example

<table>
<thead>
<tr>
<th>$T$</th>
<th>$\mathcal{A}_1$ (time point 1)</th>
<th>$\mathcal{A}_2$ (time point 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebServer $\sqsubseteq$ Server, AppServer $\sqsubseteq$ Server, WebServer $\sqsubseteq$ $\neg$AppServer</td>
<td>WebServer($a$), Execute($a$, $b$)</td>
<td>WebServer($a$), AppServer($a$), Execute($a$, $c$)</td>
</tr>
</tbody>
</table>

True under semantics at time point 1

$\Box \exists y \text{Execute}(a, y)$

$\Box \text{Server}(a)$

$\Box \text{AppServer}(a)$

False under semantics at time point 1

$\Box \exists y \text{Execute}(a, y)$

$\Box \text{Server}(a)$

$\Box \text{AppServer}(a)$
Temporal query answering over inconsistent data

Example

Case without rigid predicates:

\( T \)
- WebServer \( \sqsubseteq \) Server,
- AppServer \( \sqsubseteq \) Server,
- WebServer \( \sqsubseteq \) \( \neg \)AppServer

\( A_1 \) (time point 1)
- WebServer\((a)\), Execute\((a, b)\)

\( A_2 \) (time point 2)
- WebServer\((a)\), AppServer\((a)\), Execute\((a, c)\)

\( A'_1 \)
- WebServer\((a)\), Execute\((a, b)\)

\( A'_2 \)
- WebServer\((a)\), Execute\((a, c)\)

\( A''_1 \)
- WebServer\((a)\), Execute\((a, b)\)

\( A''_2 \)
- AppServer\((a)\), Execute\((a, c)\)
Temporal query answering over inconsistent data

Example

\( T \)
- WebServer \( \sqsubseteq \) Server,
- AppServer \( \sqsubseteq \) Server,
- WebServer \( \sqsubseteq \neg \) AppServer

\( A_1 \) (time point 1)
- WebServer\((a)\),
- Execute\((a, b)\)

\( A_2 \) (time point 2)
- WebServer\((a)\),
- AppServer\((a)\),
- Execute\((a, c)\)

Case without rigid predicates: AR semantics

\( A'_1 \)
- WebServer\((a)\),
- Execute\((a, b)\)

\( A'_2 \)
- WebServer\((a)\),
- Execute\((a, c)\)

\( A''_1 \)
- WebServer\((a)\),
- Execute\((a, b)\)

\( A''_2 \)
- AppServer\((a)\),
- Execute\((a, c)\)

True under AR semantics at time point 1
- \( \Box \exists y \text{ Execute}(a, y) \)
- \( \Box \text{ Server}(a) \)

False under AR semantics at time point 1
- \( \Box \text{ AppServer}(a) \)
Temporal query answering over inconsistent data

Example

\( T \)
- WebServer \( \sqsubseteq \) Server,
- AppServer \( \sqsubseteq \) Server,
- WebServer \( \sqsubseteq \neg \) AppServer

\( A_1 \) (time point 1)
- WebServer(\( a \)), Execute(\( a, b \))

\( A_2 \) (time point 2)
- WebServer(\( a \)), AppServer(\( a \)), Execute(\( a, c \))

Case without rigid predicates: IAR semantics

\( A'_1 \)
- WebServer(\( a \)), Execute(\( a, b \))

\( A'_2 \)
- WebServer(\( a \)), Execute(\( a, c \))

\( A''_1 \)
- WebServer(\( a \)), Execute(\( a, b \))

\( A''_2 \)
- AppServer(\( a \)), Execute(\( a, c \))

True under IAR semantics at time point 1
\( \Box \exists y \) Execute(\( a, y \))

False under IAR semantics at time point 1
\( \Box \) Server(\( a \))
\( \Box \) AppServer(\( a \))
Temporal query answering over inconsistent data

Example

<table>
<thead>
<tr>
<th>( \mathcal{T} )</th>
<th>( \mathcal{A}_1 ) (time point 1)</th>
<th>( \mathcal{A}_2 ) (time point 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebServer \sqsupseteq Server, AppServer \sqsupseteq Server, WebServer \sqsubseteq \neg \text{AppServer}</td>
<td>WebServer(( a )), Execute(( a, b ))</td>
<td>WebServer(( a )), AppServer(( a )), Execute(( a, c ))</td>
</tr>
</tbody>
</table>

Case **without rigid predicates**: brave semantics

<table>
<thead>
<tr>
<th>( \mathcal{A}'_1 )</th>
<th>( \mathcal{A}'_2 )</th>
<th>( \mathcal{A}''_1 )</th>
<th>( \mathcal{A}''_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebServer(( a )), Execute(( a, b ))</td>
<td>WebServer(( a )), Execute(( a, c ))</td>
<td>Execute(( a, b ))</td>
<td>AppServer(( a )), Execute(( a, c ))</td>
</tr>
</tbody>
</table>

**True under brave semantics at time point 1**

\( \square \exists y \text{Execute}(a, y) \)

\( \square \text{Server}(a) \)

**False under brave semantics at time point 1**

\( \square \text{AppServer}(a) \)
Temporal query answering over inconsistent data

Example

\[ T \]
- WebServer \sqsubseteq Server,
- AppServer \sqsubseteq Server,
- WebServer \sqsubseteq \neg\text{AppServer}

\[ A_1 \text{ (time point 1)} \]
- WebServer(a), Execute(a, b)

\[ A_2 \text{ (time point 2)} \]
- WebServer(a), AppServer(a), Execute(a, c)

With AppServer rigid:

\[ A'_1 \]
- WebServer(a), Execute(a, b)

\[ A'_2 \]
- WebServer(a), Execute(a, c)

\[ A''_1 \]
- Execute(a, b)

\[ A''_2 \]
- AppServer(a), Execute(a, c)
Temporal query answering over inconsistent data

**Example**

\[ T \]

- WebServer \sqsubseteq Server,
- AppServer \sqsubseteq Server,
- WebServer \sqsubseteq \neg AppServer

\[ A_1 \text{ (time point 1)} \]

- WebServer(a),
- Execute(a, b)

\[ A_2 \text{ (time point 2)} \]

- WebServer(a),
- AppServer(a),
- Execute(a, c)

With AppServer rigid: AR semantics

\[ A'_1 \]

- WebServer(a),
- Execute(a, b)

\[ A'_2 \]

- WebServer(a),
- Execute(a, c)

\[ A''_1 \]

- Execute(a, b)

\[ A''_2 \]

- AppServer(a),
- Execute(a, c)

**True under AR semantics at time point 1**

- \[ \Box \exists y \text{Execute}(a, y) \]
- \[ \Box \text{Server}(a) \]

**False under AR semantics at time point 1**

- \[ \Box \neg \text{AppServer}(a) \]
Temporal query answering over inconsistent data

Example

<table>
<thead>
<tr>
<th>$T$</th>
<th>$A_1$ (time point 1)</th>
<th>$A_2$ (time point 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebServer ⊑ Server, AppServer ⊑ Server, WebServer ⊑ ¬AppServer</td>
<td>WebServer($a$), Execute($a, b$)</td>
<td>WebServer($a$), AppServer($a$), Execute($a, c$)</td>
</tr>
</tbody>
</table>

With AppServer rigid: IAR semantics

<table>
<thead>
<tr>
<th>$A'_1$</th>
<th>$A'_2$</th>
<th>$A''_1$</th>
<th>$A''_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebServer($a$), Execute($a, b$)</td>
<td>WebServer($a$), Execute($a, c$)</td>
<td>Execute($a, b$)</td>
<td>AppServer($a$), Execute($a, c$)</td>
</tr>
</tbody>
</table>

True under IAR semantics at time point 1

$\Box \exists y \text{Execute}(a, y)$

False under IAR semantics at time point 1

$\Box \text{Server}(a)$
$\Box \text{AppServer}(a)$
Temporal query answering over inconsistent data

Example

\( T \)
- WebServer ⊑ Server,
- AppServer ⊑ Server,
- WebServer ⊑ ¬AppServer

\( A_1 \) (time point 1)
- WebServer\((a)\), Execute\((a, b)\)

\( A_2 \) (time point 2)
- WebServer\((a)\), AppServer\((a)\), Execute\((a, c)\)

With AppServer rigid: brave semantics

\( A'_1 \)
- WebServer\((a)\), Execute\((a, b)\)

\( A'_2 \)
- WebServer\((a)\), Execute\((a, c)\)

\( A''_1 \)
- Execute\((a, b)\)

\( A''_2 \)
- AppServer\((a)\), Execute\((a, c)\)

True under brave semantics at time point 1
- □∃y Execute\((a, y)\)
- □Server\((a)\)
- □AppServer\((a)\)

False under brave semantics at time point 1
Idea: combining known algorithms to perform inconsistency-tolerant TCQ answering

Without rigid predicates: combine algorithms

- classical TCQ answering + atemporal IAR query answering = IAR temporal query answering

- classical TCQ answering + atemporal AR query answering = sound approximation of AR answers and AR temporal query answering for restricted queries (without operators $\lor$, $\Diamond$, $\Diamond^-$, $U$, $S$)

- not true for brave
TCQ answering over $\text{DL-Lite}_\mathcal{R}$ ($\sim$ OWL 2 QL) TKBs

What we did
- complete the complexity picture for the classical semantics
- establish the complexity of inconsistency-tolerant TCQ answering
  - 3 inconsistency-tolerant semantics
  - 3 cases depending on the rigid predicates allowed
  - data and combined complexity
Complexity analysis
Completing the complexity picture for the classical semantics

Combined complexity of TCQ answering under classical semantics

- known result: \textbf{PSpace-complete} if \textit{negation} allowed in the query

- without negation: combined complexity drops to \textbf{NP-complete}

- cases with rigid predicates reduced to the case without rigid predicates by adding a set of assertions to every ABox
## Complexity analysis
Complexity of TCQ answering over DL-Lite\(\mathcal{R}\) TKB

<table>
<thead>
<tr>
<th></th>
<th>classical</th>
<th>AR</th>
<th>IAR</th>
<th>brave</th>
</tr>
</thead>
<tbody>
<tr>
<td>atemporal</td>
<td>in P</td>
<td>coNP-c</td>
<td>in P</td>
<td>in P</td>
</tr>
<tr>
<td>no rigid predicate</td>
<td>in P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rigid concepts only</td>
<td>in P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rigid concepts and roles</td>
<td>in P</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Data complexity

<table>
<thead>
<tr>
<th></th>
<th>classical</th>
<th>AR</th>
<th>IAR</th>
<th>brave</th>
</tr>
</thead>
<tbody>
<tr>
<td>atemporal</td>
<td>NP-c</td>
<td>(\Pi^p_2)-c</td>
<td>NP-c</td>
<td>NP-c</td>
</tr>
<tr>
<td>no rigid predicate</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rigid concepts only</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rigid concepts and roles</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Combined complexity
### Complexity analysis

#### Complexity of TCQ answering over DL-Lite\(^R\) TKB

<table>
<thead>
<tr>
<th></th>
<th>classical</th>
<th>AR</th>
<th>IAR</th>
<th>brave</th>
</tr>
</thead>
<tbody>
<tr>
<td>atemporal</td>
<td>in P</td>
<td>coNP-c</td>
<td>in P</td>
<td>in P</td>
</tr>
<tr>
<td>no rigid predicate</td>
<td>in P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rigid concepts only</td>
<td>in P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rigid concepts and roles</td>
<td>in P</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Data complexity

<table>
<thead>
<tr>
<th></th>
<th>classical</th>
<th>AR</th>
<th>IAR</th>
<th>brave</th>
</tr>
</thead>
<tbody>
<tr>
<td>atemporal</td>
<td>NP-c</td>
<td>(\Pi^p_2)-c</td>
<td>NP-c</td>
<td>NP-c</td>
</tr>
<tr>
<td>no rigid predicate</td>
<td>NP-c</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rigid concepts only</td>
<td>NP-c</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rigid concepts and roles</td>
<td>NP-c</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Combined complexity

<table>
<thead>
<tr>
<th></th>
<th>classical</th>
<th>AR</th>
<th>IAR</th>
<th>brave</th>
</tr>
</thead>
</table>
Complexity analysis
Complexity of TCQ answering over DL-Lite\textsubscript{R} TKB

<table>
<thead>
<tr>
<th></th>
<th>classical</th>
<th>AR</th>
<th>IAR</th>
<th>brave</th>
</tr>
</thead>
<tbody>
<tr>
<td>atemporal</td>
<td>in P</td>
<td>coNP-c</td>
<td>in P</td>
<td>in P</td>
</tr>
<tr>
<td>no rigid predicate</td>
<td>in P</td>
<td>coNP-c</td>
<td>in P</td>
<td>in P</td>
</tr>
<tr>
<td>rigid concepts only</td>
<td>in P</td>
<td>coNP-c</td>
<td>in P</td>
<td>NP-c</td>
</tr>
<tr>
<td>rigid concepts and roles</td>
<td>in P</td>
<td>coNP-c</td>
<td>in P</td>
<td>NP-c</td>
</tr>
</tbody>
</table>

Data complexity

<table>
<thead>
<tr>
<th></th>
<th>classical</th>
<th>AR</th>
<th>IAR</th>
<th>brave</th>
</tr>
</thead>
<tbody>
<tr>
<td>atemporal</td>
<td>NP-c</td>
<td>(\Pi^p_2)-c</td>
<td>NP-c</td>
<td>NP-c</td>
</tr>
<tr>
<td>no rigid predicate</td>
<td>NP-c</td>
<td>(\Pi^p_2)-c</td>
<td>NP-c</td>
<td>NP-c</td>
</tr>
<tr>
<td>rigid concepts only</td>
<td>NP-c</td>
<td>(\Pi^p_2)-c</td>
<td>NP-c</td>
<td>NP-c</td>
</tr>
<tr>
<td>rigid concepts and roles</td>
<td>NP-c</td>
<td>(\Pi^p_2)-c</td>
<td>NP-c</td>
<td>NP-c</td>
</tr>
</tbody>
</table>

Combined complexity
Conclusion and perspectives

Contributions

- Inconsistency-tolerant temporal query answering
  - AR, IAR, brave semantics extended to TCQ answering
  - practical algorithms for some cases (without rigid predicates)
    - IAR
    - AR for restricted queries
    - sound approximation of AR for general queries

- Complexity analysis
  - temporal dimension does not increase combined complexity
  - higher data complexity in only two cases out of nine

Future work

- Identify more cases where known algorithms can be used
- Practical algorithms for other cases
- Investigate $\mathcal{EL}_\bot$ ($\sim$ OWL 2 EL)

Thanks for your attention.

Questions?
Temporal query answering

- Temporal knowledge base (TKB) \( \langle \mathcal{T}, (A_i)_{0 \leq i \leq n} \rangle \)
  - global ontology: domain knowledge holds eternally
  - sequence of datasets: data at different time points

- **Rigid** predicates: interpretations are not allowed to change over time

- Temporal conjunctive query (TCQ): CQs + LTL operators
  - conjunctive queries are TCQs
  - if \( \phi_1, \phi_2 \) are TCQs, so are
    - \( \phi_1 \land \phi_2 \) (and) and \( \phi_1 \lor \phi_2 \) (or)
    - \( \bigcirc \phi_1 \) (next) and \( \bigcirc \neg \phi_1 \) (previous)
    - \( \bullet \phi_1 \) (weak next) and \( \bullet \neg \phi_1 \) (weak previous)
    - \( \Box \phi_1 \) (always) and \( \Box \neg \phi_1 \) (always in the past)
    - \( \Diamond \phi_1 \) (eventually) and \( \Diamond \neg \phi_1 \) (some time in the past)
    - \( \phi_1 U \phi_2 \) (until) and \( \phi_1 S \phi_2 \) (since)