Temporal Query Answering in DL-Lite over Inconsistent Data

Camille Bourgaux[1,2] & Anni-Yasmin Turhan[1]

[1]Technische Universität Dresden, [2]Télécom ParisTech

16th International Semantic Web Conference
October 24th 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...)

data (sensor values...)

System

Ex: cluster of servers

ABox (Data) $\mathcal{A}$

Specific knowledge (snapshot of the system)

adaptation

Ex: balance load

Conjunctive Query q

Description of the situation

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

situation recognition

Management component

Ex: list of servers almost overloaded

Ontology-based query answering

TBox $\mathcal{T}$

$\mathsf{WebServer} \sqsubseteq \mathsf{Server}$, *web servers are servers*

$\mathsf{AppServer} \sqsubseteq \mathsf{Server}$, *application servers are servers*

$\mathsf{WebServer} \sqsubseteq \neg\mathsf{AppServer}$ *web and application servers are disjoint*

ABox $\mathcal{A}$

$\mathsf{Execute}(a, b)$, $\mathsf{WebServer}(a)$

Conjunctive Query q

$\exists y \mathsf{Server}(x) \wedge \mathsf{Execute}(x, y)$ *retrieve servers that execute something*

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

Ontology-based query answering for situation recognition

Inconsistency-tolerant query answering

TBox $\mathcal{T}$

WebServer $\sqsubseteq$ Server,	*web servers are servers*
AppServer $\sqsubseteq$ Server,	*application servers are servers*
WebServer $\sqsubseteq \neg$AppServer	*web and application servers are disjoint*

ABox $\mathcal{A}$

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

Conjunctive Query q

$\exists y$Server(x) $\wedge$ Execute(x, y)	*retrieve servers that execute something*

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

Ontology-based query answering for situation recognition

Temporal query answering

TBox $\mathcal{T}$

$\mathsf{WebServer} \sqsubseteq \mathsf{Server}$, *web servers are servers*
$\mathsf{AppServer} \sqsubseteq \mathsf{Server}$, *application servers are servers*
$\mathsf{WebServer} \sqsubseteq \neg\mathsf{AppServer}$ *web and application servers are disjoint*

Sequence of ABoxes $(\mathcal{A}_i)_{1 \leq i \leq 2}$

$\mathcal{A}_1$: $\mathsf{Execute}(a, b)$, $\mathsf{WebServer}(a)$ *data at time point 1*
$\mathcal{A}_2$: $\mathsf{Execute}(a, c)$, $\mathsf{WebServer}(a)$ *data at time point 2*

Temporal Conjunctive Query ϕ

$\mathsf{Server}(x) \wedge \Box^{-} \exists y \mathsf{Execute}(x, y)$ *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

TBox $\mathcal{T}$	ABox $\mathcal{A}$	Repair 1	Repair 2
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)

Inconsistency-tolerant semantics for knowledge bases

Repair

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

AR semantics (ABox Repair)

AR answer $\Leftrightarrow$ answer in every repair

TBox $\mathcal{T}$

WebServer $\sqsubseteq$ Server,
AppServer $\sqsubseteq$ Server,
WebServer $\sqsubseteq$ $\neg$AppServer

ABox $\mathcal{A}$

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

Repair 1

WebServer(a),
Execute(a,b)

Repair 2

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

TBox $\mathcal{T}$

WebServer $\sqsubseteq$ Server,
AppServer $\sqsubseteq$ Server,
WebServer $\sqsubseteq$ $\neg$AppServer

ABox $\mathcal{A}$

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

Repair 1

WebServer(a),
Execute(a,b)

Repair 2

AppServer(a),
Execute(a,b)

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 $\Leftrightarrow$ answer in some repair

TBox $\mathcal{T}$	ABox $\mathcal{A}$	Repair 1	Repair 2
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 brave semantics

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

False under brave semantics

$$\text{IAR-answers} \subseteq \text{AR-answers} \subseteq \text{brave-answers}$$

- 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) $\langle \mathcal{T}, (\mathcal{A}_i)_{0 \leq i \leq n} \rangle$
  - 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:

$\phi_1 = \bigcirc^{-} \mathsf{AlmostOverloaded}(x)$: *was almost overloaded at previous time point*

$\phi_2 = \Diamond^{-}(\mathsf{Critical}(x) \wedge \bigcirc^{-} \Diamond^{-} \mathsf{Critical}(x))$: *has been in a critical situation twice*

$\mathsf{Critical}(x) := \exists y \mathsf{Execute}(x, y) \wedge \mathsf{IncreasingWorkload}(y) \wedge \mathsf{AlmostOverloaded}(x)$

Outline

1. Introduction
2. Preliminary notions
3. Temporal query answering over inconsistent data
4. Complexity analysis
5. Conclusion and perspectives

Repair of a temporal knowledge base

"maximal subset of the data consistent with the TBox"

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

$\mathcal{T}$

WebServer $\sqsubseteq$ Server,
AppServer $\sqsubseteq$ Server,
WebServer $\sqsubseteq$ $\neg$AppServer

$\mathcal{A}_1$ (time point 1)

WebServer(a),
Execute(a, b)

$\mathcal{A}_2$ (time point 2)

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

$\mathcal{T}$

$\text{WebServer} \sqsubseteq \text{Server}$,
$\text{AppServer} \sqsubseteq \text{Server}$,
$\text{WebServer} \sqsubseteq \neg\text{AppServer}$

$\mathcal{A}_1$ (time point 1)

$\text{WebServer}(a)$,
$\text{Execute}(a, b)$

$\mathcal{A}_2$ (time point 2)

$\text{WebServer}(a)$,
$\text{AppServer}(a)$,
$\text{Execute}(a, c)$

First case: independent time points, no rigid predicates
$\rightarrow$ contradictions only between timed-assertions with the same time point

$\mathcal{T}$

WebServer $\sqsubseteq$ Server,
AppServer $\sqsubseteq$ Server,
WebServer $\sqsubseteq$ $\neg$AppServer

$\mathcal{A}_1$ (time point 1)

WebServer(a),
Execute(a, b)

$\mathcal{A}_2$ (time point 2)

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

First case: independent time points, no rigid predicates
$\rightarrow$ contradictions only between timed-assertions with the same time point
$\rightarrow$ repairs of TKB $=$ sequences of repairs of KBs

$\mathcal{A}'_1$

WebServer(a),
Execute(a, b)

$\mathcal{A}'_2$

WebServer(a),
Execute(a, c)

$\mathcal{A}''_1$

WebServer(a),
Execute(a, b)

$\mathcal{A}''_2$

AppServer(a),
Execute(a, c)

$\mathcal{T}$	$\mathcal{A}_1$ (time point 1)	$\mathcal{A}_2$ (time point 2)
WebServer $\sqsubseteq$ Server, AppServer $\sqsubseteq$ Server, WebServer $\sqsubseteq$ $\neg$AppServer	WebServer(a), Execute(a, b)	WebServer(a), AppServer(a), Execute(a, c)

Second case: some rigid predicates are not allowed to change over time

$\mathcal{T}$

WebServer $\sqsubseteq$ Server,
AppServer $\sqsubseteq$ Server,
WebServer $\sqsubseteq$ $\neg$AppServer

$\mathcal{A}_1$ (time point 1)

WebServer(a),
Execute(a, b)

$\mathcal{A}_2$ (time point 2)

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

Second case: some rigid predicates are not allowed to change over time
$\rightarrow$ contradictions between data at different time points possible

$\mathcal{T}$

WebServer $\sqsubseteq$ Server,
AppServer $\sqsubseteq$ Server,
WebServer $\sqsubseteq$ $\neg$AppServer

$\mathcal{A}_1$ (time point 1)

WebServer(a),
Execute(a, b)

$\mathcal{A}_2$ (time point 2)

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

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

$\mathcal{A}'_1$

WebServer(a),
Execute(a, b)

$\mathcal{A}'_2$

WebServer(a),
Execute(a, c)

$\mathcal{A}''_1$

Execute(a, b)

$\mathcal{A}''_2$

AppServer(a),
Execute(a, c)

AR, IAR and brave semantics defined in the natural way

AR semantics

AR answer ⇔ answer in every repair

IAR semantics

IAR answer ⇔ answer in the intersection of all repairs

Brave semantics

brave answer ⇔ answer in some repair

Temporal query answering over inconsistent data

Example

$\mathcal{T}$

WebServer $\sqsubseteq$ Server,
AppServer $\sqsubseteq$ Server,
WebServer $\sqsubseteq$ $\neg$AppServer

$\mathcal{A}_1$ (time point 1)

WebServer(a),
Execute(a, b)

$\mathcal{A}_2$ (time point 2)

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

$\mathcal{T}$

WebServer $\sqsubseteq$ Server,
AppServer $\sqsubseteq$ Server,
WebServer $\sqsubseteq$ ¬AppServer

$\mathcal{A}_1$ (time point 1)

WebServer(a),
Execute(a, b)

$\mathcal{A}_2$ (time point 2)

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

Case without rigid predicates:

$\mathcal{A}'_1$

WebServer(a),
Execute(a, b)

$\mathcal{A}'_2$

WebServer(a),
Execute(a, c)

$\mathcal{A}''_1$

WebServer(a),
Execute(a, b)

$\mathcal{A}''_2$

AppServer(a),
Execute(a, c)

Temporal query answering over inconsistent data

Example

$\mathcal{T}$

WebServer $\sqsubseteq$ Server,
AppServer $\sqsubseteq$ Server,
WebServer $\sqsubseteq \neg$AppServer

$\mathcal{A}_1$ (time point 1)

WebServer(a),
Execute(a, b)

$\mathcal{A}_2$ (time point 2)

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

Case without rigid predicates: AR semantics

$\mathcal{A}'_1$

WebServer(a),
Execute(a, b)

$\mathcal{A}'_2$

WebServer(a),
Execute(a, c)

$\mathcal{A}''_1$

WebServer(a),
Execute(a, b)

$\mathcal{A}''_2$

AppServer(a),
Execute(a, c)

True under AR semantics at time point 1

$\Box\exists y$Execute(a, y)
$\Box$Server(a)

False under AR semantics at time point 1

$\Box$AppServer(a)

$\mathcal{T}$

WebServer $\sqsubseteq$ Server,
AppServer $\sqsubseteq$ Server,
WebServer $\sqsubseteq$ $\neg$AppServer

$\mathcal{A}_1$ (time point 1)

WebServer(a),
Execute(a, b)

$\mathcal{A}_2$ (time point 2)

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

Case without rigid predicates: IAR semantics

$\mathcal{A}'_1$

WebServer(a),
Execute(a, b)

$\mathcal{A}'_2$

WebServer(a),
Execute(a, c)

$\mathcal{A}''_1$

WebServer(a),
Execute(a, b)

$\mathcal{A}''_2$

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)

$\mathcal{T}$

WebServer $\sqsubseteq$ Server,
AppServer $\sqsubseteq$ Server,
WebServer $\sqsubseteq$ $\neg$AppServer

$\mathcal{A}_1$ (time point 1)

WebServer(a),
Execute(a, b)

$\mathcal{A}_2$ (time point 2)

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

Case without rigid predicates: brave semantics

$\mathcal{A}'_1$

WebServer(a),
Execute(a, b)

$\mathcal{A}'_2$

WebServer(a),
Execute(a, c)

$\mathcal{A}''_1$

WebServer(a),
Execute(a, b)

$\mathcal{A}''_2$

AppServer(a),
Execute(a, c)

True under brave semantics at time point 1

$\Box\exists y$Execute(a, y)
$\Box$Server(a)

False under brave semantics at time point 1

$\Box$AppServer(a)

$\mathcal{T}$

WebServer $\sqsubseteq$ Server,
AppServer $\sqsubseteq$ Server,
WebServer $\sqsubseteq$ $\neg$AppServer

$\mathcal{A}_1$ (time point 1)

WebServer(a),
Execute(a, b)

$\mathcal{A}_2$ (time point 2)

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

With AppServer rigid:

$\mathcal{A}'_1$

WebServer(a),
Execute(a, b)

$\mathcal{A}'_2$

WebServer(a),
Execute(a, c)

$\mathcal{A}''_1$

Execute(a, b)

$\mathcal{A}''_2$

AppServer(a),
Execute(a, c)

$\mathcal{T}$

WebServer $\sqsubseteq$ Server,
AppServer $\sqsubseteq$ Server,
WebServer $\sqsubseteq$ $\neg$AppServer

$\mathcal{A}_1$ (time point 1)

WebServer(a),
Execute(a, b)

$\mathcal{A}_2$ (time point 2)

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

With AppServer rigid: AR semantics

$\mathcal{A}'_1$

WebServer(a),
Execute(a, b)

$\mathcal{A}'_2$

WebServer(a),
Execute(a, c)

$\mathcal{A}''_1$

Execute(a, b)

$\mathcal{A}''_2$

AppServer(a),
Execute(a, c)

True under AR semantics at time point 1

$\Box\exists y$Execute(a, y)
$\Box$Server(a)

False under AR semantics at time point 1

$\Box$AppServer(a)

$\mathcal{T}$

WebServer $\sqsubseteq$ Server,
AppServer $\sqsubseteq$ Server,
WebServer $\sqsubseteq$ $\neg$AppServer

$\mathcal{A}_1$ (time point 1)

WebServer(a),
Execute(a, b)

$\mathcal{A}_2$ (time point 2)

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

With AppServer rigid: IAR semantics

$\mathcal{A}'_1$

WebServer(a),
Execute(a, b)

$\mathcal{A}'_2$

WebServer(a),
Execute(a, c)

$\mathcal{A}''_1$

Execute(a, b)

$\mathcal{A}''_2$

AppServer(a),
Execute(a, c)

True under IAR semantics at time point 1

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

False under IAR semantics at time point 1

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

$\mathcal{T}$

WebServer $\sqsubseteq$ Server,
AppServer $\sqsubseteq$ Server,
WebServer $\sqsubseteq \neg$AppServer

$\mathcal{A}_1$ (time point 1)

WebServer(a),
Execute(a, b)

$\mathcal{A}_2$ (time point 2)

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

With AppServer rigid: brave semantics

$\mathcal{A}'_1$

WebServer(a),
Execute(a, b)

$\mathcal{A}'_2$

WebServer(a),
Execute(a, c)

$\mathcal{A}''_1$

Execute(a, b)

$\mathcal{A}''_2$

AppServer(a),
Execute(a, c)

True under brave semantics at time point 1

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

False under brave semantics at time point 1

Inconsistency-tolerant temporal query answering in DL-Lite

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 $\vee$, $\Diamond$, $\Diamond^{-}$, U, S)
- not true for brave

Complexity analysis

- TCQ answering over 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

Combined complexity of TCQ answering under classical semantics

- known result: PSpace-complete if negation allowed in the query
- without negation: combined complexity drops to NP-complete
- cases with rigid predicates reduced to the case without rigid predicates by adding a set of assertions to every ABox

	classical	AR	IAR	brave
atemporal	in P	coNP-c	in P	in P
no rigid predicate	in P			
rigid concepts only	in P			
rigid concepts and roles	in P			

Data complexity

	classical	AR	IAR	brave
atemporal	NP-c	Π_2^p-c	NP-c	NP-c
no rigid predicate				
rigid concepts only				
rigid concepts and roles				

Combined complexity

	classical	AR	IAR	brave
atemporal	in P	coNP-c	in P	in P
no rigid predicate	in P			
rigid concepts only	in P			
rigid concepts and roles	in P			

Data complexity

	classical	AR	IAR	brave
atemporal	NP-c	Π_2^p-c	NP-c	NP-c
no rigid predicate	NP-c			
rigid concepts only	NP-c			
rigid concepts and roles	NP-c			

Combined complexity

	classical	AR	IAR	brave
atemporal	in P	coNP-c	in P	in P
no rigid predicate	in P	coNP-c	in P	in P
rigid concepts only	in P	coNP-c	in P	NP-c
rigid concepts and roles	in P	coNP-c	in P	NP-c

Data complexity

	classical	AR	IAR	brave
atemporal	NP-c	Π_2^p-c	NP-c	NP-c
no rigid predicate	NP-c	Π_2^p-c	NP-c	NP-c
rigid concepts only	NP-c	Π_2^p-c	NP-c	NP-c
rigid concepts and roles	NP-c	Π_2^p-c	NP-c	NP-c

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) $\langle \mathcal{T}, (\mathcal{A}_i)_{0 \leq i \leq n} \rangle$
  - 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 ϕ_1, ϕ_2 are TCQs, so are
    - $\phi_1 \wedge \phi_2$ (and) and $\phi_1 \vee \phi_2$ (or)
    - $\bigcirc\phi_1$ (next) and $\bigcirc^-\phi_1$ (previous)
    - $\bullet\phi_1$ (weak next) and $\bullet^-\phi_1$ (weak previous)
    - $\Box\phi_1$ (always) and $\Box^-\phi_1$ (always in the past)
    - $\Diamond\phi_1$ (eventually) and $\Diamond^-\phi_1$ (some time in the past)
    - $\phi_1 \mathsf{U} \phi_2$ (until) and $\phi_1 \mathsf{S} \phi_2$ (since)