TECHNISCHE UNIVERSITÄT DRESDEN
DRESDEN concept
Institut für Theoretische Informatik Lehrstuhl für Automatentheorie

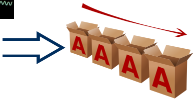
ANSWERING TEMPORAL CONJUNCTIVE QUERIES OVER DL ONTOLOGIES

Stefan Borgwardt

Dresden, 18.05.2015

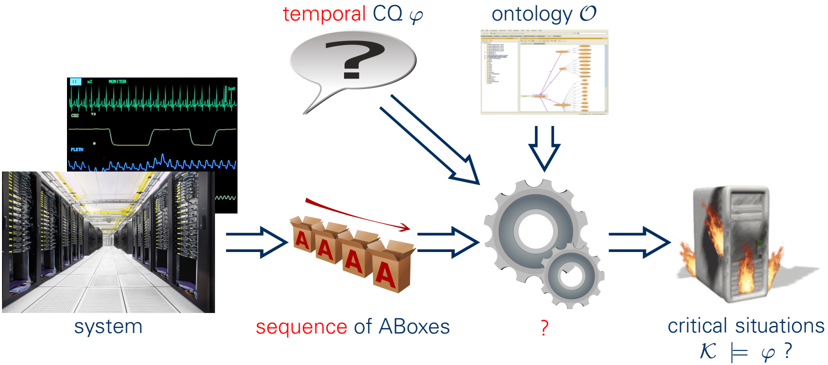
Temporal Query Answering

system



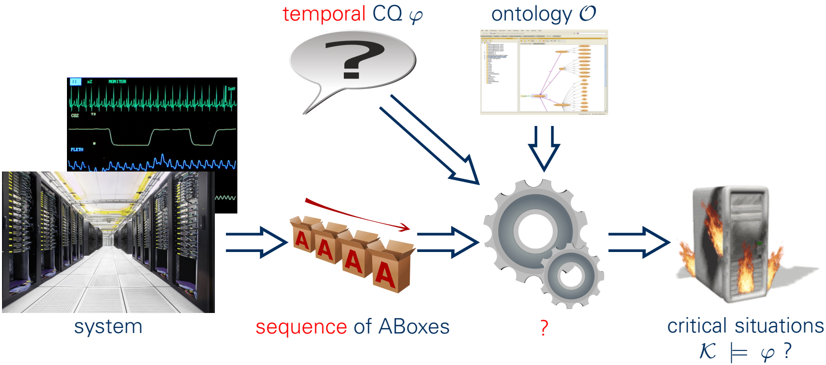
sequence of ABoxes

Temporal Query Answering



- temporal knowledge base (TKB) $\mathcal{K} = \langle (\mathcal{A}_i)_{0 \leq i \leq n}, \mathcal{O} \rangle$

Temporal Query Answering



- temporal knowledge base (TKB) $\mathcal{K} = \langle (\mathcal{A}_i)_{0 \leq i \leq n}, \mathcal{O} \rangle$
- sequences of interpretations $(\mathcal{I}_i)_{i \geq 0}$
  - shared domain
  - respect rigid names: $x^{\mathcal{I}_i} = x^{\mathcal{I}_j}$ for $x \in \mathsf{N_I} \cup \mathsf{N_{RC}} \cup \mathsf{N_{RR}}$

Temporal Conjunctive Queries (TCQs)

(Boolean) CQs + LTL:

$$\psi := \exists x_1 \ldots x_n . \alpha_1 \wedge \cdots \wedge \alpha_m$$

$$\varphi ::= \psi \mid \neg\varphi_1 \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \bigcirc\varphi_1 \mid \varphi_1 \,\mathsf{U}\, \varphi_2 \mid \bigcirc^{-}\varphi_1 \mid \varphi_1 \,\mathsf{S}\, \varphi_2$$

Temporal Conjunctive Queries (TCQs)

(Boolean) CQs + LTL:

$$\psi := \exists x_1 \dots x_n . \alpha_1 \wedge \dots \wedge \alpha_m$$

$$\varphi ::= \psi \mid \neg\varphi_1 \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \bigcirc\varphi_1 \mid \varphi_1 \, \mathsf{U} \, \varphi_2 \mid \bigcirc^{-}\varphi_1 \mid \varphi_1 \, \mathsf{S} \, \varphi_2$$

- propositional combination of CQs
- no temporal concepts, e.g.

$$\exists \mathsf{hasDisease}.\mathsf{ChickenPox} \sqsubseteq \Box\neg\exists \mathsf{hasDisease}.\mathsf{ChickenPox}$$

Temporal Conjunctive Queries (TCQs)

(Boolean) CQs + LTL:

$$\psi := \exists x_1 \dots x_n . \alpha_1 \wedge \dots \wedge \alpha_m$$

$$\varphi ::= \psi \mid \neg\varphi_1 \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \bigcirc\varphi_1 \mid \varphi_1 \mathsf{U} \varphi_2 \mid \bigcirc^-\varphi_1 \mid \varphi_1 \mathsf{S} \varphi_2$$

- propositional combination of CQs
- no temporal concepts, e.g.

$$\exists \mathsf{hasDisease.ChickenPox} \sqsubseteq \Box\neg\exists \mathsf{hasDisease.ChickenPox}$$

$$\begin{aligned}\Diamond^- \big(&\exists y.\mathsf{hasDisease}(x, y) \wedge \mathsf{ChickenPox}(y) \vee \\ &\exists y.\mathsf{hasDisease}(x, y) \wedge \mathsf{Shingles}(y) \vee \\ &\exists y.\mathsf{hasAntibodies}(x, y) \wedge \mathsf{VZVAntibodies}(y)\big)\end{aligned}$$

TCQ Entailment in $\mathcal{EL}$ and $\mathcal{ALC}$

	data complexity			combined complexity		
	(i)	(ii)	(iii)	(i)	(ii)	(iii)
$\mathcal{EL}$	P	co-NP	co-NP	PSpace	PSpace	co-NExpTime
$\mathcal{ALC}$–$\mathcal{SHQ}$	co-NP	co-NP	co-NP	ExpTime	co-NExpTime	2-ExpTime

(i) no rigid concept or role names

(ii) rigid concept names, but no rigid role names

(iii) rigid role names (and also rigid concept names)

TCQ Entailment in $\mathcal{EL}$ and $\mathcal{ALC}$

	data complexity			combined complexity		
	(i)	(ii)	(iii)	(i)	(ii)	(iii)
$\mathcal{EL}$	P	co-NP	co-NP	PSpace	PSpace	co-NExpTime
$\mathcal{ALC}$–$\mathcal{SHQ}$	co-NP	co-NP	co-NP	ExpTime	co-NExpTime	2-ExpTime

(i) no rigid concept or role names

(ii) rigid concept names, but no rigid role names

(iii) rigid role names (and also rigid concept names)

- $\mathcal{EL}$: [B, Thost, 2015]
- $\mathcal{SHQ}$: only simple CQs [Baader, B, Lippmann, 2015]

TCQ Entailment in $\mathcal{EL}$ and $\mathcal{ALC}$

	data complexity			combined complexity		
	(i)	(ii)	(iii)	(i)	(ii)	(iii)
$\mathcal{EL}$	P	co-NP	co-NP	PSpace	PSpace	co-NExpTime
$\mathcal{ALC}$–$\mathcal{SHQ}$	co-NP	co-NP	co-NP	ExpTime	co-NExpTime	2-ExpTime

(i) no rigid concept or role names

(ii) rigid concept names, but no rigid role names

(iii) rigid role names (and also rigid concept names)

- $\mathcal{EL}$: [B, Thost, 2015]
- $\mathcal{SHQ}$: only simple CQs [Baader, B, Lippmann, 2015]
- techniques based on $\mathcal{ALC}$-LTL [Baader, Ghilardi, Lutz, 2008/2012]
- here: only upper bounds

Main Approach

Decide satisfiability of φ w.r.t. $\mathcal{K} = \langle(\mathcal{A}_i)_{0 \le i \le n}, \mathcal{O}\rangle$:

- replace CQs $\psi_1, \ldots, \psi_m$ by propositional variables $p_1, \ldots, p_m$
- find propositional LTL model of φ^{p}

$$\Diamond^-(p_1 \vee p_2 \vee p_3)$$

Main Approach

Decide satisfiability of φ w.r.t. $\mathcal{K} = \langle(\mathcal{A}_i)_{0 \leq i \leq n}, \mathcal{O}\rangle$:

- replace CQs $\psi_1, \ldots, \psi_m$ by propositional variables $p_1, \ldots, p_m$
- find propositional LTL model of φ^{p}

$\Diamond^-(p_1 \vee p_2 \vee p_3)$

set of possible worlds $\mathcal{S} = \{X_1, \ldots, X_k\} \subseteq 2^{\{p_1,\ldots,p_m\}}$
mapping $\iota\colon \{0, \ldots, n\} \rightarrow \{1, \ldots, k\}$

t-satisfiability:
Find LTL model $(w_i)_{i \geq 0}$ of φ^{p} such that

- $w_i \in \mathcal{S}$ for all $i \geq 0$, and
- $w_i = X_{\iota(i)}$ for all $i \in \{0, \ldots, n\}$.

Main Approach

Decide satisfiability of φ w.r.t. $\mathcal{K} = \langle (\mathcal{A}_i)_{0\leq i\leq n}, \mathcal{O}\rangle$:

- replace CQs $\psi_1, \dots, \psi_m$ by propositional variables $p_1, \dots, p_m$
- find propositional LTL model of φ^{p}

$$\Diamond^-(p_1 \vee p_2 \vee p_3)$$

set of possible worlds $\mathcal{S} = \{X_1, \dots, X_k\} \subseteq 2^{\{p_1,\dots,p_m\}}$
mapping $\iota\colon \{0, \dots, n\} \to \{1, \dots, k\}$

t-satisfiability:
Find LTL model $(w_i)_{i\geq 0}$ of φ^{p} such that

- $w_i \in \mathcal{S}$ for all $i \geq 0$, and
- $w_i = X_{\iota(i)}$ for all $i \in \{0, \dots, n\}$.

r-satisfiability:
Find models

- of γ_i and $\mathcal{O}$ ($1 \leq i \leq k$), and
- of $\gamma_{\iota(i)}$, $\mathcal{A}_i$, and $\mathcal{O}$ ($0 \leq i \leq n$)

that

- have the same domain and
- respect rigid names.

$$\gamma_i := \bigwedge_{p_j \in X_i} \psi_j \wedge \bigwedge_{p_j \in \overline{X_i}} \neg\psi_j$$

t-satisfiability

- guess complete sets of subformulae of φ^{p}
- check whether two consecutive sets are "compatible"
- ensure that the first $n + 1$ sets correspond to $X_{\iota(0)}, \ldots, X_{\iota(n)}$
- ensure that φ^{p} is satisfied at time point n

t-satisfiability

- guess complete sets of subformulae of φ^{p}
- check whether two consecutive sets are "compatible"
- ensure that the first $n+1$ sets correspond to $X_{\iota(0)}, \dots, X_{\iota(n)}$
- ensure that φ^{p} is satisfied at time point n
- Büchi automaton of exponential size ($n \cdot 2^{|\varphi|}$)
- ExpTime in combined complexity, P in data complexity

t-satisfiability

- guess complete sets of subformulae of φ^{p}
- check whether two consecutive sets are "compatible"
- ensure that the first $n + 1$ sets correspond to $X_{\iota(0)}, \dots, X_{\iota(n)}$
- ensure that φ^{p} is satisfied at time point n
- Büchi automaton of exponential size ($n \cdot 2^{|\varphi|}$)
- ExpTime in combined complexity, P in data complexity
- alternative: PSpace Turing machine

r-satisfiability in Case (i)

- no dependency between satisfiability tests for γ_i w.r.t. $\mathcal{O}$
- collect all satisfiable $X_i \subseteq \{p_1, \ldots, p_m\}$ into "maximal" set $\mathcal{S}$
- check satisfiability of $\gamma_{\iota(i)}$ w.r.t. $\langle \mathcal{A}_i, \mathcal{O} \rangle$

r-satisfiability in Case (i)

- no dependency between satisfiability tests for γ_i w.r.t. $\mathcal{O}$
- collect all satisfiable $X_i \subseteq \{p_1, \ldots, p_m\}$ into "maximal" set $\mathcal{S}$
- check satisfiability of $\gamma_{\iota(i)}$ w.r.t. $\langle \mathcal{A}_i, \mathcal{O} \rangle$

satisfiablity of $\gamma = \varrho_1 \wedge \cdots \wedge \varrho_\ell \wedge \neg\sigma_1 \wedge \cdots \wedge \neg\sigma_o$ w.r.t. $\langle \mathcal{A}, \mathcal{O} \rangle$

r-satisfiability in Case (i)

- no dependency between satisfiability tests for γ_i w.r.t. $\mathcal{O}$
- collect all satisfiable $X_i \subseteq \{p_1, \ldots, p_m\}$ into "maximal" set $\mathcal{S}$
- check satisfiability of $\gamma_{\iota(i)}$ w.r.t. $\langle \mathcal{A}_i, \mathcal{O} \rangle$

satisfiablity of $\gamma = \varrho_1 \wedge \cdots \wedge \varrho_\ell \wedge \neg\sigma_1 \wedge \cdots \wedge \neg\sigma_o$ w.r.t. $\langle \mathcal{A}, \mathcal{O} \rangle$

$$\langle \mathcal{A} \cup \mathcal{A}_{\varrho_1} \cup \cdots \cup \mathcal{A}_{\varrho_\ell}, \mathcal{O} \rangle \not\models \sigma_1 \vee \cdots \vee \sigma_o$$

r-satisfiability in Case (i)

- no dependency between satisfiability tests for γ_i w.r.t. $\mathcal{O}$
- collect all satisfiable $X_i \subseteq \{p_1, \ldots, p_m\}$ into "maximal" set $\mathcal{S}$
- check satisfiability of $\gamma_{\iota(i)}$ w.r.t. $\langle \mathcal{A}_i, \mathcal{O} \rangle$

satisfiablity of $\gamma = \varrho_1 \wedge \cdots \wedge \varrho_\ell \wedge \neg\sigma_1 \wedge \cdots \wedge \neg\sigma_o$ w.r.t. $\langle \mathcal{A}, \mathcal{O} \rangle$

$$\langle \mathcal{A} \cup \mathcal{A}_{\varrho_1} \cup \cdots \cup \mathcal{A}_{\varrho_\ell}, \mathcal{O} \rangle \not\models \sigma_1 \vee \cdots \vee \sigma_o$$

ExpTime in combined complexity:

- compute $\mathcal{S}$: 2^m exponential tests
- enumerate all ι: $(2^m)^{n+1}$ many
- $n + 1$ exponential tests for r-satisfiability
- Büchi automaton for t-satisfiability: $p(n \cdot 2^{|\varphi|})$

r-satisfiability in Case (iii)

- create one copy $\mathcal{O}^{(i)}$ of $\mathcal{O}$ for each $X_i \in \mathcal{S}$
- non-rigid names are renamed: $A \rightsquigarrow A^{(i)}$
- $A^{(i)}$ represents the extension of A in the model for γ_i

r-satisfiability in Case (iii)

- create one copy $\mathcal{O}^{(i)}$ of $\mathcal{O}$ for each $X_i \in \mathcal{S}$
- non-rigid names are renamed: $A \rightsquigarrow A^{(i)}$
- $A^{(i)}$ represents the extension of A in the model for γ_i
- rigid names are shared among all copies
- additional copies for each combination of $\gamma_{\iota(i)}$ and $\mathcal{A}_i$

r-satisfiability in Case (iii)

- create one copy $\mathcal{O}^{(i)}$ of $\mathcal{O}$ for each $X_i \in \mathcal{S}$
- non-rigid names are renamed: $A \rightsquigarrow A^{(i)}$
- $A^{(i)}$ represents the extension of A in the model for γ_i
- rigid names are shared among all copies
- additional copies for each combination of $\gamma_{\iota(i)}$ and $\mathcal{A}_i$

satisfiability of $\gamma_{\mathcal{S},\iota} := \bigwedge_{i=1}^{k} \gamma_i^{(i)} \wedge \bigwedge_{i=0}^{n} \gamma_{\iota(i)}^{(k+i+1)}$

w.r.t. $\mathcal{A}_\iota := \bigcup_{i=0}^{n} \mathcal{A}_i^{(k+i+1)}$ and $\mathcal{O}_{\mathcal{S},\iota} := \bigcup_{i=1}^{k+n+1} \mathcal{O}^{(i)}$

r-satisfiability in Case (iii)

satisfiability of $\gamma_{\mathcal{S},\iota} := \bigwedge_{i=1}^{k} \gamma_i^{(i)} \wedge \bigwedge_{i=0}^{n} \gamma_{\iota(i)}^{(k+i+1)}$

w.r.t. $\mathcal{A}_\iota := \bigcup_{i=0}^{n} \mathcal{A}_i^{(k+i+1)}$ and $\mathcal{O}_{\mathcal{S},\iota} := \bigcup_{i=1}^{k+n+1} \mathcal{O}^{(i)}$

	combined complexity	data complexity
$\mathcal{S}$	enumerate all $\mathcal{S}$: 2^{2^m} many	
ι	enumerate all ι: $(2^m)^{n+1}$ many	
r-sat.	one double exponential test	
t-sat.	Büchi automaton: $p(n \cdot 2^{\lvert\varphi\rvert})$	
	2-ExpTime	

r-satisfiability in Case (iii)

satisfiability of $\gamma_{\mathcal{S},\iota} := \bigwedge_{i=1}^{k} \gamma_i^{(i)} \wedge \bigwedge_{i=0}^{n} \gamma_{\iota(i)}^{(k+i+1)}$

w.r.t. $\mathcal{A}_\iota := \bigcup_{i=0}^{n} \mathcal{A}_i^{(k+i+1)}$ and $\mathcal{O}_{\mathcal{S},\iota} := \bigcup_{i=1}^{k+n+1} \mathcal{O}^{(i)}$

	combined complexity	data complexity
$\mathcal{S}$	enumerate all $\mathcal{S}$: 2^{2^m} many	enumerate all $\mathcal{S}$: constant
ι	enumerate all ι: $(2^m)^{n+1}$ many	guess ι of size $(n+1) \cdot m$
r-sat.	one double exponential test	one polynomial test: linearly many disjuncts of constant size
t-sat.	Büchi automaton: $p(n \cdot 2^{\lvert\varphi\rvert})$	Büchi automaton: $p(n)$
	2-ExpTime	co-NP

r-satisfiability in Case (ii)

	data complexity			combined complexity		
	(i)	(ii)	(iii)	(i)	(ii)	(iii)
$\mathcal{EL}$	P	co-NP	co-NP	PSpace	PSpace	co-NExpTime
$\mathcal{ALC}$–$\mathcal{SHQ}$	co-NP	co-NP	co-NP	ExpTime	co-NExpTime	2-ExpTime

r-satisfiability in Case (ii)

	data complexity			combined complexity		
	(i)	(ii)	(iii)	(i)	(ii)	(iii)
$\mathcal{EL}$	P	co-NP	co-NP	PSpace	PSpace	co-NExpTime
$\mathcal{ALC}$–$\mathcal{SHQ}$	co-NP	co-NP	co-NP	ExpTime	co-NExpTime	2-ExpTime

- guess possible types $\mathcal{D} \subseteq 2^{\mathsf{N}_{\mathsf{RC}}(\mathcal{O})}$ for domain elements
- guess mapping $\tau\colon \mathsf{N}_\mathsf{I}(\mathcal{K}) \to \mathcal{D}$ (types for named domain elements)

r-satisfiability in Case (ii)

	data complexity			combined complexity		
	(i)	(ii)	(iii)	(i)	(ii)	(iii)
$\mathcal{EL}$	P	co-NP	co-NP	PSpace	PSpace	co-NExpTime
$\mathcal{ALC}$–$\mathcal{SHQ}$	co-NP	co-NP	co-NP	ExpTime	co-NExpTime	2-ExpTime

- guess possible types $\mathcal{D} \subseteq 2^{\mathsf{N_{RC}}(\mathcal{O})}$ for domain elements
- guess mapping $\tau\colon \mathsf{N_I}(\mathcal{K}) \to \mathcal{D}$ (types for named domain elements)
- satisfiability of γ_i w.r.t. $\langle \mathcal{A}_\tau, \mathcal{O} \rangle$ (respecting $\mathcal{D}$)
- satisfiability of $\gamma_{\iota(i)}$ w.r.t. $\langle \mathcal{A}_i \cup \mathcal{A}_\tau, \mathcal{O} \rangle$ (respecting $\mathcal{D}$)
- each test is possible in exponential time

Other DLs

	data complexity (i)	data complexity (ii)	data complexity (iii)	combined complexity (i)	combined complexity (ii)	combined complexity (iii)
$\mathcal{EL}$	P	co-NP	co-NP	PSpace	PSpace	co-NExpTime
$\mathcal{ALC}$–$\mathcal{SHQ}$	co-NP	co-NP	?	ExpTime	co-NExpTime	2-ExpTime
DL-Lite$_{core}$–*DL-Lite*$^{\mathcal{H}}_{horn}$	ALogTime	ALogTime	ALogTime	PSpace	PSpace	PSpace
DL-Lite$_{krom}$–*DL-Lite*$^{\mathcal{H}}_{bool}$	co-NP	co-NP	?	ExpTime	co-NExpTime	2-ExpTime
$\mathcal{SHOQ}$/$\mathcal{SHOI}$	≥co-NP	?	?	2-ExpTime	2-ExpTime	2-ExpTime
$\mathcal{SHIQ}$	co-NP	co-NP	?	2-ExpTime	2-ExpTime	2-ExpTime
$\mathcal{ALCO}$–$\mathcal{ALCHOQ}$	co-NP	co-NP	?	≥co-NExpTime	?	2-ExpTime
$\mathcal{S}$–$\mathcal{SOQ}$	co-NP	co-NP	?	≥co-NExpTime	?	2-ExpTime
$\mathcal{ALCOIQ}$	decidable	decidable	decidable	decidable	decidable	decidable

Thank you!