Answering Queries in Description Logics: Theory and Applications to Data Management

Diego Calvanese[1], Michael Zakharyaschev[2]

[1] KRDB Research Centre
Free University of Bozen-Bolzano

[2] Birbeck College, London

ESSLLI 2010, August 14–20, 2010
Copenhagen, Denmark

Overview of the Course

1. Introduction and background
    1. Ontology-based data management
    2. Brief introduction to computational complexity
    3. Query answering in databases
    4. Querying databases and ontologies
2. Lightweight description logics
    5. Introduction to description logics
    6. DLs for conceptual data modeling: the *DL-Lite* family
    7. The $\mathcal{EL}$ family of tractable description logics
3. Query answering in the *DL-Lite* family
    8. Query answering in description logics
    9. Lower bounds for more expressive description logics
    10. Query answering by rewriting
4. The combined approach to query answering
    11. Query answering in *DL-Lite*: data completion
    12. Query rewriting in $\mathcal{EL}$
5. Linking ontologies to relational data
    13. The impedance mismatch problem
    14. Query answering in Ontology-Based Data Access systems
6. Conclusions and references

Lecture 4:

The combined approach to query answering in *DL-Lite* and $\mathcal{EL}$

(A survey of query answering techniques for *DL-Lite* and $\mathcal{EL}$ logics)

Recommended reading

DL-Lite <u>available on the web</u>

(1) A. Artale, D. Calvanese, R. Kontchakov and M. Zakharyaschev.
The DL-Lite family and relations. JAIR, 36:1–69, 2009.

(2) R. Kontchakov, C. Lutz, D. Toman, F. Wolter and M. Zakharyaschev.
The combined approach to query answering in DL-Lite. Proceedings of KR 2010.

(3) R. Rosati and A. Almatelli. *Improving query answering over DL-Lite ontologies*.
Proceedings of KR 2010.

$\mathcal{EL}$

(4) C. Lutz, D. Toman, F. Wolter. *Conjunctive query answering in the description logic $\mathcal{EL}$ using a relational database system,* Proceedings of IJCAI 2009.

Acknowledgements: Roman Kontchakov, Carsten Lutz, Frank Wolter

Ontology-based data access: the story so far

- Next generation of information systems: instance data + ontologies

Reasoning problem: answering queries over knowledge & data

- **Instance queries** $q = C(x)$ over a TBox $\mathcal{T}$ and an Abox $\mathcal{A}$

  an ABox individual a is an answer iff $\mathcal{T}, \mathcal{A} \models C(a)$

Example $\mathcal{T} = \{\text{Boss} \sqsubseteq \text{Employee}\}$, $\mathcal{A} = \{\text{Boss(bob)}\}$, $q = \text{Emploee}(x)$

'list all employees'

Answer: $x = \text{bob}$ (not an answer over $\mathcal{A}$ alone)

$\mathcal{T}, \mathcal{A} \models C(a)$ iff there is no $\mathcal{I} \models \mathcal{T} \cup \mathcal{A}$ such that $\mathcal{I} \models \neg C(a)$

iff $\mathcal{T} \cup \mathcal{A} \cup \{\neg C(a)\}$ is not satisfiable

Instance checking is as complex as satisfiability checking

The story so far: more complex queries

- **Conjunctive queries** $\quad q = \exists \vec{y}\, \varphi(\vec{x}, \vec{y})$,

  where $\varphi(\vec{x}, \vec{y})$ is a conjunction of atoms $A(z)$, $R(z, z')$ with $z, z' \in \vec{x} \cup \vec{y}$

  $\vec{x}$ are the **answer variables**, $\vec{y}$ the **quantified variables**

  a tuple $\vec{a}$ of ABox individuals is an answer iff $\mathcal{I} \models \exists \vec{y}\, \varphi(\vec{a}, \vec{y})$ for every $\mathcal{I} \models \mathcal{T} \cup \mathcal{A}$

  usually **more complex** than satisfiability

  complexity of answering CQs without quantified variables?

- **Positive existential queries** $\quad q = \exists \vec{y}\, \varphi(\vec{x}, \vec{y})$, $\quad \varphi$ may contain both $\wedge$ and $\vee$

  (but no $\neg$)

- **General FO queries** $\quad$ may contain $\wedge$, $\vee$, $\neg$, $\forall$, $\exists$

  no good: validity of FO formulas is **undecidable**

? description logics for which ontology-based query answering is

(1) as efficient as database query answering and

(2) based on relational database management systems

Answering CQs in *DL-Lite*$^{\mathcal{N}}_{bool}$: exercise

$\mathcal{T}$:

$$
\begin{array}{rclrcl}
\text{Research} & \sqsubseteq & \exists \text{worksIn}, & \exists \text{worksIn}^- & \sqsubseteq & \text{Project}, \\
\text{Project} & \sqsubseteq & \exists \text{manages}^-, & \exists \text{manages} & \sqsubseteq & \text{Academic} \sqcup \text{Visiting}, \\
\exists \text{teaches} & \sqsubseteq & \text{Academic} \sqcup \text{Research}, & \text{Academic} & \sqsubseteq & \exists \text{teaches} \sqcap \leq 1\, \text{teaches}, \\
\text{Research} \sqcap \text{Visiting} & \sqsubseteq & \bot, & \exists \text{writes} & \sqsubseteq & \text{Academic} \sqcup \text{Research},
\end{array}
$$

$$\mathcal{A} = \{\text{teaches}(a, b), \text{teaches}(a, c)\}$$

$$q = \exists y \, \big((\exists \text{teaches})(y) \land (\leq 1\, \text{teaches})(y)\big)$$

is there anybody who teaches precisely one module?

$$\mathcal{T}' = \mathcal{T} \cup \{\text{Visiting} \sqsubseteq \geq 2\, \text{writes}\}$$

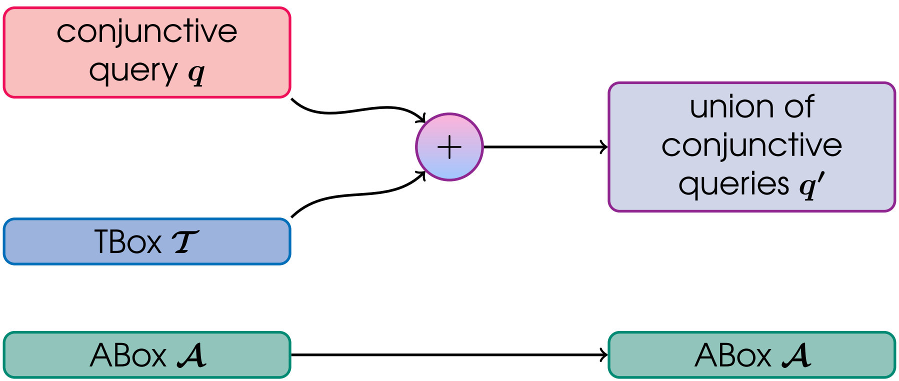
Disjunction is (NP-) **hard** even for data complexity

Only **Horn** logics can be suitable for ontology-based data access

Approach 1: query rewriting

Given a CQ $q(\vec{x})$ over $\mathcal{T}$, rewrite $q(\vec{x})$ into an FO query $q'(\vec{x})$ such that

for all $\mathcal{A}$ and $\vec{a}$, $\quad \mathcal{T}, \mathcal{A} \models q[\vec{a}] \quad$ iff $\quad \mathcal{A} \models q'[\vec{a}]$



'Maximal' DLs for which query answering is in **FO** (**=AC^0**) for data complexity:

DL-Lite$^{(\mathcal{H},\mathcal{N})}_{horn}$ under UNA and *DL-Lite*$^{\mathcal{H}}_{horn}$ without UNA

Query rewriting (cont.)

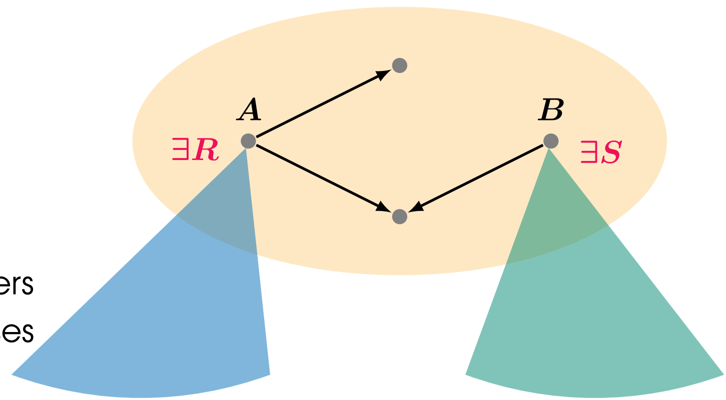
Want: all tuples $\vec{a}$ of individuals in $\mathcal{A}$ such that $\mathcal{I}_{\mathcal{K}} \models q(\vec{a})$

where $\mathcal{I}_{\mathcal{K}}$ is the **canonical model** of $\mathcal{K} = (\mathcal{T}, \mathcal{A})$

Can: query the **ABox** $\mathcal{A}$ (using an RDBMS)

To construct the canonical model $\mathcal{I}_{\mathcal{K}}$:

1. take the ABox
2. apply TBox axioms to ABox
3. satisfy the existential quantifiers by introducing 'fresh' witnesses

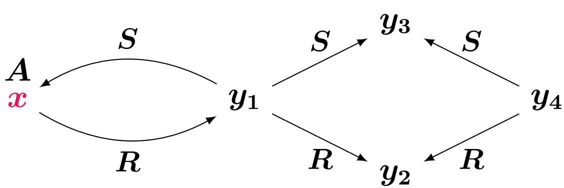


q' should incorporate $\mathcal{T}$

Query rewriting: exercise

Compute the rewriting q' for the following CQ and TBox:

$q(x) =$



$$\mathcal{T} = \{B \sqsubseteq \exists R, \quad B \sqsubseteq \exists S, \quad \exists R \sqsubseteq A\}$$

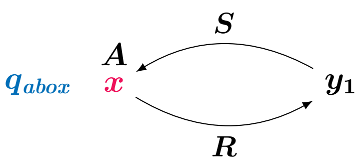
or

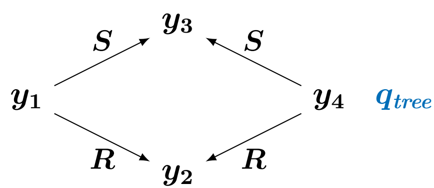
$$q(x) = \exists y_1, y_2, y_3, y_4 \; \big[A(x) \land R(x, y_1) \land S(y_1, x) \land R(y_1, y_2) \land S(y_1, y_3) \land R(y_4, y_2) \land S(y_4, y_3)\big]$$

Hint: Consider all possible locations for y_1, y_2, y_3, y_4 in the canonical model (in ABox or the tree part)

Exercise (cont.)

Suppose y_1 is in the ABox, while y_2, y_3, y_4 are in the tree part

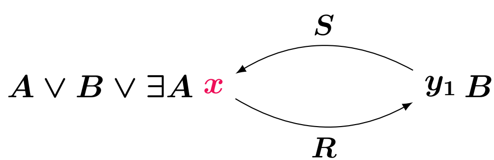




$$\mathcal{T} = \{B \sqsubseteq \exists R, \quad B \sqsubseteq \exists S, \quad \exists R \sqsubseteq A\}$$

- Which concepts at y_1 can ensure that there is a match for q_{tree} in the canonical model?
- Which concepts at x can ensure A?

rewritten query for this partition:



take disjunction of such queries for **all** partitions

Query rewriting: summary

- **Off-the-shelf RDBMSs** can be used for CQ answering in *DL-Lite*

  working systems available (Quonto, Requiem, Presto)

- **Experimental results:** not scalable for large *DL-Lite*$_{core}$ ontologies

  complexity paradox?

  Reason: q over $(\mathcal{T}, \mathcal{A})$ $\rightsquigarrow_{\mathcal{T}}$ q' over $\mathcal{A}$ with $|q'| = O(|\mathcal{T}| \cdot |q|)^{|q|}$

  is it optimal?

- Is data complexity a proper measure?

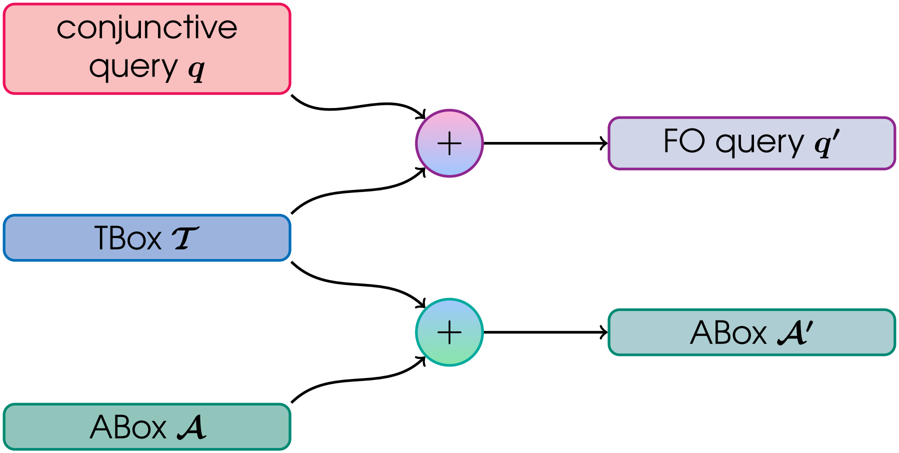
  (in RDBMSs, typical queries are relatively small...)

  Take the structure of $\mathcal{A}$, $\mathcal{T}$, q into account? Bounded treewidth? ...

- The rewriting approach is **not** applicable to other tractable DLs, e.g., $\mathcal{EL}$

  why?

Approach 2: data completion

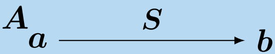


- Extend ABox to the canonical model of $(\mathcal{T}, \mathcal{A})$
- Encode it as a **finite** structure $\mathcal{A}'$
- Rewrite q into q' to ensure that the answers to q' over $\mathcal{A}'$ are correct

$\rightsquigarrow$ **combined approach**

Compact canonical models (example)

ABox $\mathcal{A}$

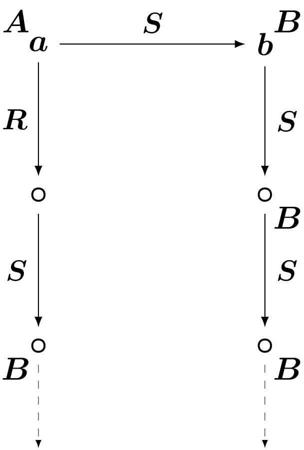


TBox $\mathcal{T}$

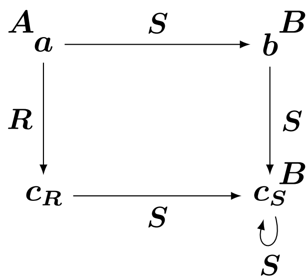
$A \sqsubseteq \exists R, \qquad \exists S^- \sqsubseteq B,$

$\exists R^- \sqsubseteq \exists S, \quad \exists S^- \sqsubseteq \exists S$

Canonical model $\mathcal{I}_\mathcal{K}$



'Compact' canonical model $\mathcal{C}_\mathcal{K}$



$\mathcal{I}_\mathcal{K}$ is obtained by 'unravelling' $\mathcal{C}_\mathcal{K}$

Does $\mathcal{C}_\mathcal{K}$ give correct answers to queries?

Constructing $\mathcal{C}_{\mathcal{K}}$

Compact canonical interpretation $\mathcal{C}_{\mathcal{K}}$:

$\Delta^{\mathcal{C}_{\mathcal{K}}} = \mathsf{Ind}(\mathcal{A}) \cup \{c_R \mid R \text{ is generating in } \mathcal{K}\}$ $\qquad$ c_R is a **witness for R**

$a \rightsquigarrow c_{R_1} \rightsquigarrow \cdots \rightsquigarrow c_{R_n}$ $\qquad$ R_n is **generating**

$\mathcal{K} \models \exists R_1(a)$ but $R_1(a,b) \notin \mathcal{A}$ for all $b \in \mathsf{Ind}(\mathcal{A})$

$\mathcal{T} \models \exists R_i^- \sqsubseteq \exists R_{i+1}$ and $R_i^- \neq R_{i+1}$

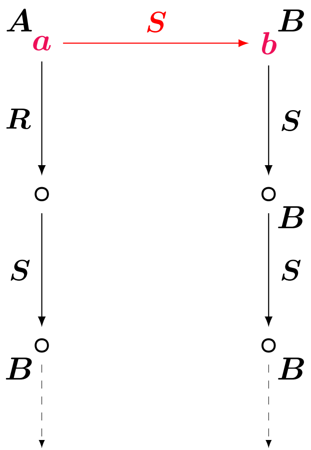
$A^{\mathcal{C}_{\mathcal{K}}} = \{a \mid \mathcal{K} \models A(a)\} \cup \{c_R \mid \mathcal{T} \models \exists R^- \sqsubseteq A\}$ $\qquad$ (A a concept name)

$P^{\mathcal{C}_{\mathcal{K}}} = \{(a,b) \mid P(a,b) \in \mathcal{A}\} \cup \{(d, c_P) \mid d \rightsquigarrow c_P\} \cup \{(c_{P^-}, d) \mid d \rightsquigarrow c_{P^-}\}$

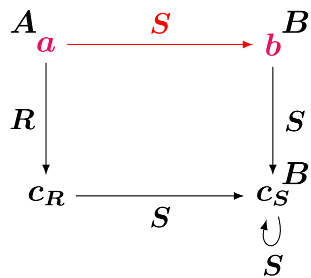
(P a role name)

Querying $\mathcal{C}_\mathcal{K}$

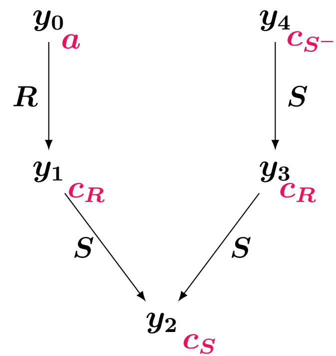
$\mathcal{I}_\mathcal{K}$



$\mathcal{C}_\mathcal{K}$



q



What is the answer to q over $\mathcal{I}_\mathcal{K}$?

What is the answer to q over $\mathcal{C}_\mathcal{K}$?

Find an **FO expressible condition** for such situations

Tree witnesses

Given $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, q and $R(x, y) \in q$,

one can compute (in polynomial time) a partial function

$$f_{R(x,y)} : terms(q) \longrightarrow \{c_S \mid S \text{ used in } \mathcal{K}\} \cup \{\varepsilon\}$$

such that

- if $f_{R(x,y)}$ does not exist then y cannot mapped to c_R
- if y is mapped to c_R in $\mathcal{C}_\mathcal{K}$ and $f_{R(x,y)}(z)$ is defined then
  - if $f_{R(x,y)}(z) = \varepsilon$ then we must have $x = z$
  - otherwise z must be mapped to $f_{R(x,y)}(z)$

In the previous example, $f_{R(y_1,y_2)}(y_3) = \varepsilon$

$f_{R(y,y)}$ does not exists

Query rewriting for *DL-Lite*$^{\mathcal{N}}_{horn}$ (1)

rewrite a given CQ $q = \exists \vec{u}\, \varphi$ into an FO query $q^\dagger$ such that

- answers to q over $\mathcal{I}_\mathcal{K}$ = answers to $q^\dagger$ over $\mathcal{C}_\mathcal{K}$
- $|q^\dagger| = O(|q| \cdot |\mathcal{T}|)$

$$q^\dagger = \exists \vec{u}\, (\varphi \land \varphi_1 \land \varphi_2 \land \varphi_3)$$

$$\varphi_1 = \bigwedge_{v \notin \vec{u}} \ \bigwedge_{R \text{ a role in } \mathcal{T}} (v \neq c_R)$$

'all answer variables must get ABox values'

NB. if φ_1 is replaced with $\varphi_1' = \bigwedge_{v \notin \vec{u}} \neg \mathrm{aux}(v)$, where aux is a new relation containing all c_R,

then $|q^\dagger| = O(|q|)$

Query rewriting for *DL-Lite*$^{\mathcal{N}}_{horn}$ (2)

$$\varphi_2 = \bigwedge_{\substack{R(x,y)\in q\\ f_{R(x,y)} \text{ does not exist}}} (y \neq c_R)$$

if no tree witness exists then y cannot be mapped to a non-ABox element

$$\varphi_3 = \bigwedge_{\substack{R(x,y)\in q\\ f_{R(x,y)} \text{ exists}}} \Big((y = c_R) \quad \to \bigwedge_{f_{R(x,y)}(z)=\varepsilon} (z = x)\Big)$$

Exercises

Exercise 1: compute q' for the exercise on page 13

$$\varphi_1 = \varphi_2 = \top$$
$$\varphi_3 = (y_2 = c_S) \rightarrow (y_1 = y_3)$$

Exercise 2: Use the rewriting and combined approaches for the following KB and query:

$\mathcal{T}$:

$\mathsf{Student} \sqsubseteq \exists\mathsf{hasTutor}$

$\exists\mathsf{teaches}^- \sqsubseteq \mathsf{Student}$

$\mathsf{Professor} \sqsubseteq \exists\mathsf{teaches}$

$\exists\mathsf{hasTutor}^- \sqsubseteq \mathsf{Professor}$

$\mathcal{A}$: $\{\mathsf{Student}(a),\ \mathsf{Professor}(b),\ \mathsf{teaches}(b,a)\}$

$$q(x) = \mathsf{teaches}(x,y), \mathsf{hasTutor}(y,z), \mathsf{hasTutor}(u,z)$$

Query answering in *DL-Lite*$_{horn}^{(\mathcal{HN})}$

what can we do with **role inclusions**?

Reduce **positive existential queries** over *DL-Lite*$_{horn}^{(\mathcal{HN})}$ KBs to

unions of (**exponentially many**) CQs over *DL-Lite*$_{horn}^{\mathcal{N}}$ KBs

Step 1. *DL-Lite*$_{horn}^{(\mathcal{HN})}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ $\rightsquigarrow$ *DL-Lite*$_{horn}^{\mathcal{N}}$ KB $\mathcal{K} = (\mathcal{T}_h, \mathcal{A})$

by replacing all $R \sqsubseteq^* S$ with $\exists R \sqsubseteq \exists S$ ($\sqsubseteq^*$ is the transitive closure of $\sqsubseteq$)

Step 2. Positive existential q over $\mathcal{K}$ $\rightsquigarrow$ union of CQs q_h over $\mathcal{C}_{\mathcal{K}_h}$:

– replace each $R(t, t')$ in q with $\bigvee_{S \sqsubseteq^* R} S(t, t')$

– convert result into disjunctive normal form (exponential blowup)

$\leq r^{|q|}$ conjuncts, where r is the depth of $\sqsubseteq^*$

$$\mathcal{K} \models q(\vec{a}) \quad \text{iff} \quad \mathcal{C}_{\mathcal{K}_h} \models q_h$$

is there a polynomial rewriting?

Other applications

- $\mathcal{C}_\mathcal{K}$ can be constructed by first-order queries $\rightsquigarrow$
  pure polynomial rewriting for $\textit{DL-Lite}^{(\mathcal{N})}_{core}$
- without the UNA, the technique is applicable to query answering in $\textit{DL-Lite}^{(\mathcal{HF})}_{horn}$
  (which is P-complete for data complexity)
- experiments show that the approach is **competitive**
  with executing the **original query** over the data
  (the formulas φ_1–φ_3 introduce additional selection conditions on top of the original query)

Open questions

– is the exponential blowup unavoidable for role inclusions?

– is the exponential blowup unavoidable for positive existential queries?

– for which DLs pure rewriting can be polynomial?

Query rewriting in $\mathcal{EL}$

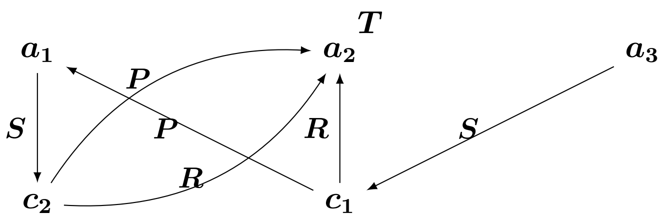
The query rewriting approach **cannot** work for $\mathcal{EL}$ because already

> instance checking in $\mathcal{EL}$ is **PTime-complete** w.r.t. data complexity

Lower bound: by reduction of PTime-complete entailment for Horn CNF

E.g., $\varphi = (a_1 \wedge a_2 \rightarrow a_3) \wedge (a_2 \rightarrow a_1) \wedge a_2$ is encoded by the ABox

$\mathcal{A}_\varphi$



and the (φ-independent) TBox $\mathcal{T}$: $\mathcal{T} = \{\exists S.(\exists P.T \sqcap \exists R.T) \sqsubseteq T\}$

$$\varphi \models a_i \quad \text{iff} \quad (\mathcal{T}, \mathcal{A}_\varphi) \models T(a_i)$$

Compact canonical models for $\mathcal{EL}$

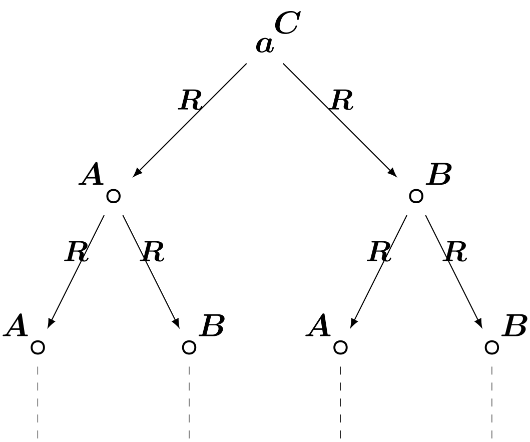
ABox $\mathcal{A}$

C a

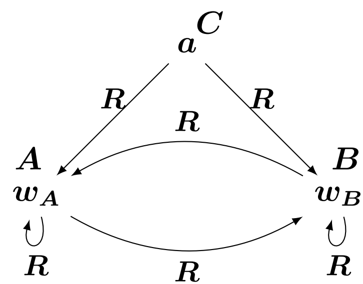
TBox $\mathcal{T}$

$\top \sqsubseteq \exists R.A, \qquad \top \sqsubseteq \exists R.B$

Canonical model $\mathcal{I}_\mathcal{K}$



Compact canonical model $\mathcal{C}_\mathcal{K}$



$\mathcal{I}_\mathcal{K}$ is obtained by unravelling $\mathcal{C}_\mathcal{K}$

Difference from *DL-Lite*: multiple R-successors of non-ABox points

Constructing $\mathcal{C}_\mathcal{K}$

Compact canonical interpretation $\mathcal{C}_\mathcal{K}$:

$$\mathsf{Con}(\mathcal{K}) = \text{ the set of all concepts in } \mathcal{K}$$

$$\Delta^{\mathcal{C}_\mathcal{K}} = \mathsf{Ind}(\mathcal{A}) \cup \{w_C \mid C \in \mathsf{Con}(\mathcal{K})\}$$ w_C is a **witness for C**

$$A^{\mathcal{C}_\mathcal{K}} = \{a \mid \mathcal{K} \models A(a)\} \cup \{w_C \mid \mathcal{T} \models C \sqsubseteq A\}$$ (A a concept name)

$$\begin{aligned} P^{\mathcal{C}_\mathcal{K}} = {} & \{(a,b) \mid P(a,b) \in \mathcal{A}\} \cup \\ & \{(a, w_C) \mid \mathcal{K} \models \exists P.C(a)\} \cup \\ & \{(w_C, w_D) \mid \mathcal{T} \models C \sqsubseteq \exists P.D\} \end{aligned}$$ (P a role name)

Query rewriting for $\mathcal{EL}$

rewrite a given CQ $q = \exists \vec{u}\, \varphi$ into an FO query $q^\dagger$ such that

- answers to q over $\mathcal{I}_\mathcal{K}$ = answers to q^* over $\mathcal{C}_\mathcal{K}$
- $|q^*| = O(|q| \cdot |\mathcal{T}|)$

$$\boxed{q^\dagger = \exists \vec{u}\, (\varphi \land \varphi_1 \land \varphi_2 \land \varphi_3)}$$

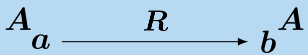
φ_1: answer variables and variables in cycles in q must be mapped to ABox

φ_2: if $x_1 \xrightarrow{R} x_2 \xleftarrow{R} x_3$ in q and x_2 is mapped outside the ABox then $x_1 = x_3$

φ_3: if $x_1 \xrightarrow{R} x_2 \xleftarrow{S} x_3$ in q and $R \neq S$ then x_2 must be mapped to ABox

Query rewriting for $\mathcal{EL}$: example 1

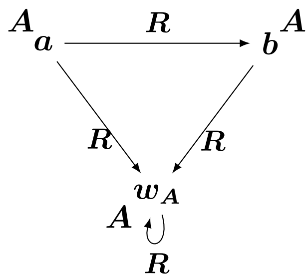
ABox $\mathcal{A}$



TBox $\mathcal{T}$

$A \sqsubseteq \exists R.A$

$\mathcal{C}_{\mathcal{K}}$

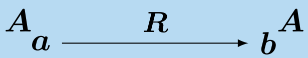


$q(x) \ = \ \exists y\ \big[R(x,y) \land R(y,y)\big]$ answers $x = a, \quad x = b$

$q^*(x) \ = \ \exists y\ \big[R(x,y) \land R(y,y) \land \text{ABox}(x) \land \text{ABox}(y)\big]$ no answer

Query rewriting for $\mathcal{EL}$: example 2

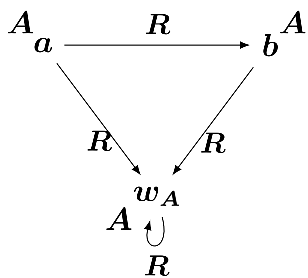
ABox $\mathcal{A}$



TBox $\mathcal{T}$

$A \sqsubseteq \exists R.A$

$\mathcal{C}_\mathcal{K}$



$$q(x, x') \;=\; \exists y \left[R(x, y) \wedge R(x', y) \wedge R(x, x')\right]$$ answers $x = a, \quad x' = b$

$$q^*(x) \;=\; \exists y \left[R(x, y) \wedge R(x', y) \wedge R(x, x') \wedge \mathsf{ABox}(x) \wedge \mathsf{ABox}(x') \wedge (\neg\mathsf{ABox}(y) \rightarrow x = x')\right]$$ no answer

Discussion

Horn-$\mathcal{SHIQ}$ T. Eiter, G. Gottlob, M. Ortiz, M. Šimkus (2008):

answering CQs in Horn-$\mathcal{SHIQ}$ is

- **ExpTime-complete** w.r.t. combined complexity, and
- **PTime-complete** w.r.t. data complexity

(no experimental data yet)

Combined technique for Horn-$\mathcal{SHIQ}$?

Other formalisms? E.g., the TGD and EGD fragment of FOL $(\varphi \longrightarrow \exists \vec{y} \psi)$

Datalog rewritings? E.g., $\mathcal{ELHIO}^{\neg}$ H. Peŕez-Urbina, B. Motik, I. Horrocks (2009)

What is the proper complexity measure? E.g., can we have sameAs ?

CWA or OWA? E.g., datalog$^{\pm}$ A. Calì, G. Gottlob, T. Lukasiewicz (2009)