The DL-Lite Family of Languages
A FO Perspective

Alessandro Artale

KRDB Research Centre – Free University of Bozen-Bolzano

Joint work with *D. Calvanese, R. Kontchakov, M. Zakharyaschev*

TU Dresden. December 14–15, 2011

Recommended Readings

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

[2] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati.
DL-Lite: Tractable description logics for ontologies. Proceedings of AAAI 2005.

[3] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati.
Tractable reasoning and efficient query answering in DLs: The DL-Lite family.
Journal of Automated Reasoning, 39:385–429, 2007.

Outline

1. Ontology based data access
2. The *DL-Lite*-family of ontology languages:
   DL-Lite$_{bool}$, *DL-Lite*$_{horn}$, *DL-Lite*$_{core}$, *DL-Lite*$_{krom}$
3. Translation to the one-variable fragment of First-Order Logic
4. Answering UCQ
5. Conclusions

Ontologies in Computer Science

- Ontologies are formal specifications of a particular domain
- Used to represent information at the conceptual level in terms of classes/concepts/entities and relationships between them
- Typically expressed in logic:
  - First Order Logic
  - Description Logics: a specialized formalism (typically a fragment of FOL) for expressing knowledge in terms of classes and relationships

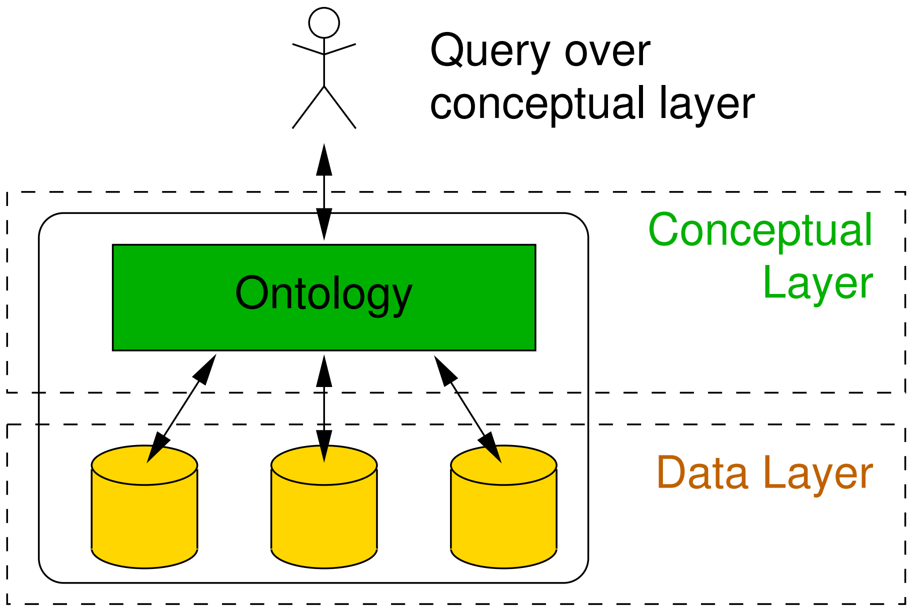
Ontologies in Computer Science

- Ontologies are formal specifications of a particular domain
- Used to represent information at the conceptual level in terms of classes/concepts/entities and relationships between them
- Typically expressed in logic:
  - First Order Logic
  - Description Logics: a specialized formalism (typically a fragment of FOL) for expressing knowledge in terms of classes and relationships
- Share strong similarities with other representation formalisms in Computer Science
  - Frame systems in Artificial Intelligence
  - ER diagrams in databases and information systems
  - UML class diagrams in software engineering
  - Constraints over a relational schema (inclusion and key dependencies)

Ontology based data access

Desiderata: achieve logical transparency in access to data:

- Hide to the user where and how data are stored
- Present to the user a conceptual view of the data
- Query the data sources through the conceptual model



As in Data Integration, but with a rich conceptual description as the global view

Description Logics: The *DL-Lite* family

The *DL-Lite* DLs provide an answer to our basic question: *For which ontology languages can we answer queries over an ontology efficiently (in data complexity)?*

- *DL-Lite* is a family of DLs optimized according to the tradeoff between expressive power and data complexity
- The *DL-Lite* family establishes the maximal subset of DLs constructs for which data complexity of query answering is LOGSPACE
  - Query answering techniques leverage on RDBMS technology (i.e. SQL)

Objectives of the Lecture

1. To show how the basic *DL-Lite* in [CDLLR,AAAI05; CDLLR,KR06] can be extended with full Booleans, cardinalities and role inclusion axioms obtaining the logic *DL-Lite*$_{bool}$ and three sublanguages: *DL-Lite*$_{krom}$, *DL-Lite*$_{core}$ and *DL-Lite*$_{horn}$
2. To characterize the first-order logic nature of class of *DL-Lite* DLs
3. To provide tight combined complexity results for reasoning in the new languages showing that:
   - Cardinalities are harmless;
   - Role inclusions, in most cases, destroy the nice computational behavior of *DL-Lite*!
4. To show the LOGSPACE data complexity result of answering positive existential queries in *DL-Lite*$_{horn}$.

The simplest *DL-Lite* Language: *DL-Lite*$_{core}$

DL-Lite$_{core}$ Ontology language:

- Concept Inclusions: $B_1 \sqsubseteq B_2$, $B_1 \sqsubseteq \neg B_2$ with:

$$B \longrightarrow A \mid \exists R \mid \bot$$
$$R \longrightarrow P \mid P^-$$

- ABox assertions: $A(c)$, $\neg A(c)$ $P(c,d)$, $\neg P(c,d)$ with c, d constants

The most expressive *DL-Lite* Language: *DL-Lite*$^{\mathcal{R},\mathcal{N}}_{bool}$

DL-Lite$^{\mathcal{R},\mathcal{N}}_{bool}$ Ontology language:

- Concept Inclusions: $C_1 \sqsubseteq C_2$, with:

$$\begin{array}{lcl} C & \longrightarrow & B \mid \neg C \mid C_1 \sqcap C_2 \\ B & \longrightarrow & A \mid \geq q\, R \mid \bot \\ R & \longrightarrow & P \mid P^- \end{array}$$

- Role Inclusions: $R_1 \sqsubseteq R_2$
- ABox assertions ($\mathcal{A}$): $A(c)$, $\neg A(c)$ $P(c, d)$, $\neg P(c, d)$, with c, d constants.

The most expressive *DL-Lite* Language: *DL-Lite*$^{\mathcal{R},\mathcal{N}}_{bool}$

DL-Lite$^{\mathcal{R},\mathcal{N}}_{bool}$ Ontology language:

- Concept Inclusions: $C_1 \sqsubseteq C_2$, with:

$$\begin{array}{lcl} C & \longrightarrow & B \mid \neg C \mid C_1 \sqcap C_2 \\ B & \longrightarrow & A \mid \ \geq q\, R \mid \bot \\ R & \longrightarrow & P \mid P^- \end{array}$$

- Role Inclusions: $R_1 \sqsubseteq R_2$
- ABox assertions ($\mathcal{A}$): $A(c)$, $\neg A(c)$ $P(c,d)$, $\neg P(c,d)$, with c, d constants.
- A TBox, $\mathcal{T}$, is a set of concept and role inclusions. A TBox, $\mathcal{T}$, is what we call an Ontology, $\mathcal{O}$.
- A Knowledge Base is a pair $\mathcal{K} = (\mathcal{T}, \mathcal{A})$

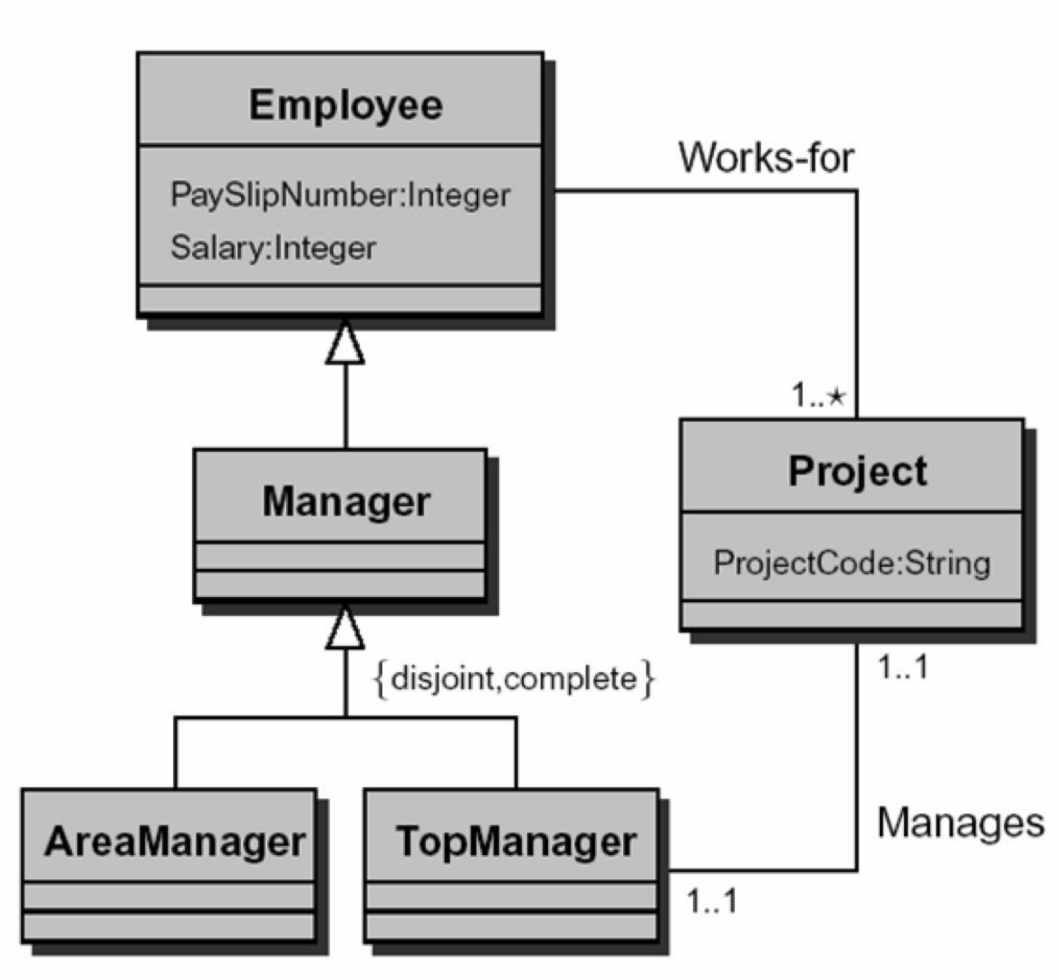
DL-Lite$^{\mathcal{R},\mathcal{N}}_{core}$, *DL-Lite*$^{\mathcal{R},\mathcal{N}}_{krom}$, *DL-Lite*$^{\mathcal{R},\mathcal{N}}_{horn}$

DL-Lite$^{\mathcal{R},\mathcal{N}}_{core}$ Ontology language: $B_1 \sqsubseteq B_2, \quad B_1 \sqsubseteq \neg B_2$

DL-Lite$^{\mathcal{R},\mathcal{N}}_{krom}$ Ontology language: $B_1 \sqsubseteq B_2, \quad B_1 \sqsubseteq \neg B_2, \quad \neg B_1 \sqsubseteq B_2$

DL-Lite$^{\mathcal{R},\mathcal{N}}_{horn}$ Ontology language: $\bigsqcap_k B_k \sqsubseteq B$

DL-Lite – Example



$$
\begin{array}{rcl}
\mathsf{Manager} & \sqsubseteq & \mathsf{Employee} \\
\mathsf{AreaManager} & \sqsubseteq & \mathsf{Manager} \\
\mathsf{TopManager} & \sqsubseteq & \mathsf{Manager} \\
\mathsf{AreaManager} \sqcap \mathsf{TopManager} & \sqsubseteq & \bot \\
\exists \mathsf{WorksFor} & \sqsubseteq & \mathsf{Employee} \\
\exists \mathsf{WorksFor}^- & \sqsubseteq & \mathsf{Project} \\
\mathsf{Project} & \sqsubseteq & \exists \mathsf{WorksFor}^- \\
\geq 2\, \mathsf{Manages} & \sqsubseteq & \bot \\
\geq 2\, \mathsf{Manages}^- & \sqsubseteq & \bot \\
 & \vdots &
\end{array}
$$

See [Artale et. al.,ER07] for more details on the correspondence between *DL-Lite* and conceptual data models

Semantics of *DL-Lite*

Construct	Syntax	Example	Semantics
atomic concept	A	Doctor	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
atomic role	P	child	$P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
inverse role	P^-	child^-	$\{(d,e) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (e,d) \in P^{\mathcal{I}}\}$
empty concept	$\bot$	$\bot$	$\emptyset$
conjunction	$C_1 \sqcap C_2$	Doctor $\sqcap$ Male	$C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$
negation	$\neg C$	$\neg$(Doctor $\sqcap$ Male)	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
cardinalities	$\geq nR$	$\geq 2\ \text{child}^-$	$\{d \in \Delta^{\mathcal{I}} \mid \sharp\{e \in \Delta^{\mathcal{I}} \mid (d,e) \in R^{\mathcal{I}}\} \geq n\}$
inclusion asser.	$Cl \sqsubseteq Cr$	Father $\sqsubseteq\ \geq 1$ child	$Cl^{\mathcal{I}} \subseteq Cr^{\mathcal{I}}$
memb. asser.	$A(a)$	Father(bob)	$a^{\mathcal{I}} \in A^{\mathcal{I}}$
memb. asser.	$P(a,b)$	child(bob, ann)	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}}$

Relevant reasoning tasks

We are interested in:

1. Checking the consistency of the ontology (**Schema Consistency**)
2. Checking the consistency of single classes in the ontology (**Class Consistency**)
3. Checking whether new constraints hold in the ontology (e.g. discovering new ISA – **Class Subsumption**)
4. Checking the consistency of the data wrt the ontology
5. Answering queries expressed over the ontology by means of the underlying data

FO Translation

DL-Lite$_{bool}^{\mathcal{N}}$ is NP-complete – Upper Bound

- Class consistency for *DL-Lite*$_{bool}^{\mathcal{N}}$ can be reduced to formula satisfiability for the one-variable fragment $\mathcal{QL}^1$ of first-order logic without equality and functions.
- Formula satisfiability for the one-variable fragment $\mathcal{QL}^1$ is known to be NP-complete [BGG:97].
- First we present a lengthy yet quite 'natural' and 'transparent' reduction $\cdot^{\dagger}$;
- Then we shall see that this reduction can be substantially optimised to a LOGSPACE reduction $\cdot^{\ddagger}$.

DL-Lite$^{\mathcal{N}}_{bool}$ is NP-complete – Upper Bound – Translating C

- Inductive translation of Concepts, C^*:

$$
\begin{aligned}
(\bot)^* &= \bot \\
(A)^* &= A(x) \\
(\neg C)^* &= \neg C^*(x) \\
(C_1 \sqcap C_2)^* &= C_1^*(x) \wedge C_2^*(x) \\
(\geq q\, R)^* &= E_q R(x)
\end{aligned}
$$

DL-Lite$^{\mathcal{N}}_{bool}$ is NP-complete – Upper Bound – Translation $\mathcal{K}^\dagger$

- Translation of $\mathcal{K}$=(TBox,ABox): The lengthy translation $\mathcal{K}^\dagger$.

$$\mathcal{K}^\dagger \;=\; \Big[\mathcal{T}^* \;\wedge\; \bigwedge_{R \in role^{\pm}(\mathcal{K})} \big(\varepsilon(R) \wedge \delta(R)\big)\Big] \;\wedge\; \Big[\mathcal{A}^\dagger \;\wedge\; \bigwedge_{R \in role^{\pm}(\mathcal{K})} R^\dagger\Big]$$

DL-Lite$^{\mathcal{N}}_{bool}$ is NP-complete – Upper Bound – Translation $\mathcal{K}^\dagger$

- Translation of $\mathcal{K}$=(TBox,ABox): The lengthy translation $\mathcal{K}^\dagger$.

$$\mathcal{K}^\dagger \;=\; \Big[\mathcal{T}^* \;\wedge\; \bigwedge_{R \in role^{\pm}(\mathcal{K})} \big(\varepsilon(R) \wedge \delta(R)\big)\Big] \;\wedge\; \Big[\mathcal{A}^\dagger \;\wedge\; \bigwedge_{R \in role^{\pm}(\mathcal{K})} R^\dagger\Big]$$

$$\mathcal{T}^* \;=\; \bigwedge_{C_1 \sqsubseteq C_2 \in \mathcal{T}} \forall x \, \big(C_1^*(x) \rightarrow C_2^*(x)\big)$$

DL-Lite$_{bool}^{\mathcal{N}}$ is NP-complete – Upper Bound – Translation $\mathcal{K}^\dagger$

- Translation of $\mathcal{K}$=(TBox,ABox): The lengthy translation $\mathcal{K}^\dagger$.

$$\mathcal{K}^\dagger \;=\; \Big[\mathcal{T}^* \;\wedge\; \bigwedge_{R \in \textit{role}^\pm(\mathcal{K})} \big(\varepsilon(R) \wedge \delta(R)\big)\Big] \;\wedge\; \Big[\mathcal{A}^\dagger \;\wedge\; \bigwedge_{R \in \textit{role}^\pm(\mathcal{K})} R^\dagger\Big]$$

$$\mathcal{T}^* \;=\; \bigwedge_{C_1 \sqsubseteq C_2 \in \mathcal{T}} \forall x \,\big(C_1^*(x) \rightarrow C_2^*(x)\big)$$

$$\mathcal{A}^\dagger \;=\; \bigwedge_{A(a_i) \in \mathcal{A}} A(a_i) \;\wedge\; \bigwedge_{P(a_i,a_j) \in \mathcal{A}} P a_i a_j$$

$$\bigwedge_{\neg A(a_i) \in \mathcal{A}} \neg A(a_i) \;\wedge\; \bigwedge_{\neg P(a_i,a_j) \in \mathcal{A}} \neg P a_i a_j$$

DL-Lite$^{\mathcal{N}}_{bool}$ is NP-complete – Upper Bound – Translation $\mathcal{K}^{\dagger}$

- Translation of $\mathcal{K}$=(TBox,ABox): The lengthy translation $\mathcal{K}^{\dagger}$.

$$\mathcal{K}^{\dagger} = \left[\mathcal{T}^{*} \wedge \bigwedge_{R\in \mathit{role}^{\pm}(\mathcal{K})}\big(\varepsilon(R) \wedge \delta(R)\big)\right] \wedge \left[\mathcal{A}^{\dagger} \wedge \bigwedge_{R\in \mathit{role}^{\pm}(\mathcal{K})} R^{\dagger}\right]$$

$$\mathcal{T}^{*} = \bigwedge_{C_1\sqsubseteq C_2\in\mathcal{T}} \forall x \left(C_1^{*}(x) \rightarrow C_2^{*}(x)\right)$$

$$\mathcal{A}^{\dagger} = \bigwedge_{A(a_i)\in\mathcal{A}} A(a_i) \wedge \bigwedge_{P(a_i,a_j)\in\mathcal{A}} Pa_ia_j$$

$$\bigwedge_{\neg A(a_i)\in\mathcal{A}} \neg A(a_i) \wedge \bigwedge_{\neg P(a_i,a_j)\in\mathcal{A}} \neg Pa_ia_j$$

$$R^{\dagger} = \bigwedge_{q=1}^{q_{\mathcal{T}}} \bigwedge_{\substack{a, a_{j_1}, \ldots, a_{j_q} \in \mathit{ob}(\mathcal{A}) \\ j_i \neq j_{i'} \text{ for } i \neq i'}} \left(\bigwedge_{i=1}^{q} Raa_{j_i} \rightarrow E_qR(a)\right) \wedge$$

$$\bigwedge_{a_i,a_j\in \mathit{ob}(\mathcal{A})}\left(Ra_ia_j \rightarrow \mathit{inv}(R)a_ja_i\right)$$

($q_{\mathcal{T}}$ is the maximum cardinality number in $\mathcal{T}$)

DL-Lite$^{\mathcal{N}}_{bool}$ is NP-complete – Upper Bound – Translation $\mathcal{K}^\dagger$ (cont.)

- Translation of $\mathcal{K}$=(TBox,ABox): The lengthy translation $\mathcal{K}^\dagger$.

$$\mathcal{K}^\dagger \;=\; \Big[\mathcal{T}^* \wedge \bigwedge_{R \in role^{\pm}(\mathcal{K})} \big(\varepsilon(R) \wedge \delta(R)\big)\Big] \wedge \Big[\mathcal{A}^\dagger \wedge \bigwedge_{R \in role^{\pm}(\mathcal{K})} R^\dagger\Big]$$

$DL\text{-}Lite^{\mathcal{N}}_{bool}$ is NP-complete – Upper Bound – Translation $\mathcal{K}^\dagger$ (cont.)

- Translation of $\mathcal{K}$=(TBox,ABox): The lengthy translation $\mathcal{K}^\dagger$.

$$\mathcal{K}^\dagger = \Big[\mathcal{T}^* \wedge \bigwedge_{R \in role^\pm(\mathcal{K})} \big(\varepsilon(R) \wedge \delta(R)\big)\Big] \wedge \Big[\mathcal{A}^\dagger \wedge \bigwedge_{R \in role^\pm(\mathcal{K})} R^\dagger\Big]$$

$$\varepsilon(R) = \forall x \left(E_1 R(x) \rightarrow inv(E_1 R(dr))\right)$$

DL-Lite$^{\mathcal{N}}_{bool}$ is NP-complete – Upper Bound – Translation $\mathcal{K}^{\dagger}$ (cont.)

- Translation of $\mathcal{K}$=(TBox,ABox): The lengthy translation $\mathcal{K}^{\dagger}$.

$$\mathcal{K}^{\dagger} = \Big[\mathcal{T}^{*} \wedge \bigwedge_{R \in role^{\pm}(\mathcal{K})} \big(\varepsilon(R) \wedge \delta(R)\big)\Big] \wedge \Big[\mathcal{A}^{\dagger} \wedge \bigwedge_{R \in role^{\pm}(\mathcal{K})} R^{\dagger}\Big]$$

$$\varepsilon(R) = \forall x \,\big(E_1R(x) \to inv(E_1R(dr))\big)$$

$$\delta(R) = \bigwedge_{q=1}^{q_{\mathcal{T}}-1} \forall x \,\big(E_{q+1}R(x) \to E_qR(x)\big)$$

DL-Lite$_{bool}^{\mathcal{N}}$ is NP-complete – FO Model

$\mathcal{T} = \{A \equiv \exists P^-,\ A \sqsubseteq\ \geq 2\, P,\ \top \sqsubseteq\ \leq 1\, P^-,\ \exists P \sqsubseteq A\}$, $\mathcal{A} = \{A(a),\ P(a, a')\}$

$\mathcal{T}^* = \big(A(x) \leftrightarrow E_1 P^-(x)\big) \wedge \big(A(x) \rightarrow E_2 P(x)\big) \wedge \neg E_2 P^-(x) \wedge \big(E_1 P(x) \rightarrow A(x)\big)$

$A^{\mathfrak{M}} = \{a, \qquad\qquad \}$ $\qquad$ $E_1 P^{-\mathfrak{M}} = \{ \qquad\qquad \}$

$E_1 P^{\mathfrak{M}} = \{$

$DL\text{-}Lite^{\mathcal{N}}_{bool}$ is NP-complete – FO Model

$\mathcal{T} = \{A \equiv \exists P^-,\ A \sqsubseteq\ \geq 2\,P,\ \top \sqsubseteq\ \leq 1\,P^-,\ \exists P \sqsubseteq A\}$, $\mathcal{A} = \{A(a),\ P(a,a')\}$

$\mathcal{T}^* = \big(A(x) \leftrightarrow E_1P^-(x)\big) \wedge \big(A(x) \rightarrow E_2P(x)\big) \wedge \neg E_2P^-(x) \wedge \big(E_1P(x) \rightarrow A(x)\big)$

$R^\dagger:\ Paa' \qquad \Rightarrow\ P^-a'a,\ E_1P(a)\ \Rightarrow\ E_1P^-(a')$

$A^{\mathfrak{M}} = \{a, \qquad\qquad \}$ $\qquad E_1P^{-\mathfrak{M}} = \{ \qquad\qquad \}$

$E_1P^{\mathfrak{M}} = \{$

DL-Lite$^{\mathcal{N}}_{bool}$ is NP-complete – FO Model

$\mathcal{T} = \{A \equiv \exists P^-,\ A \sqsubseteq\ \geq 2\, P,\ \top \sqsubseteq\ \leq 1\, P^-,\ \exists P \sqsubseteq A\}$, $\mathcal{A} = \{A(a),\ P(a, a')\}$

$$\mathcal{T}^* = \big(A(x) \leftrightarrow E_1P^-(x)\big) \wedge \big(A(x) \to E_2P(x)\big) \wedge \neg E_2P^-(x) \wedge \big(E_1P(x) \to A(x)\big)$$

$$R^\dagger :\ Paa' \quad \Rightarrow \quad P^-a'a,\ E_1P(a) \ \Rightarrow\ E_1P^-(a')$$

$$\varepsilon(R) :\ E_1P(a),\ E_1P^-(a') \quad \Rightarrow \quad E_1P^-(dp^-),\ E_1P(dp)$$

$A^{\mathfrak{M}} = \{a, \qquad\qquad \}$ $\qquad\qquad$ $E_1P^{-\mathfrak{M}} = \{a', \qquad\qquad \}$

$E_1P^{\mathfrak{M}} = \{a,$

DL-Lite$^{\mathcal{N}}_{bool}$ is NP-complete – FO Model

$\mathcal{T} = \{A \equiv \exists P^-,\ A \sqsubseteq\ \geq 2\,P,\ \top \sqsubseteq\ \leq 1\,P^-,\ \exists P \sqsubseteq A\}$, $\mathcal{A} = \{A(a),\ P(a,a')\}$

$\mathcal{T}^* = \big(A(x) \leftrightarrow E_1P^-(x)\big) \wedge \big(A(x) \rightarrow E_2P(x)\big) \wedge \neg E_2P^-(x) \wedge \big(E_1P(x) \rightarrow A(x)\big)$

$R^\dagger:\ Paa' \quad \Rightarrow \quad P^-a'a,\ E_1P(a) \ \Rightarrow\ E_1P^-(a')$

$\varepsilon(R):\ E_1P(a),\ E_1P^-(a') \quad \Rightarrow \quad E_1P^-(dp^-),\ E_1P(dp)$

$\mathcal{T}^*:\ E_1P(x) \rightarrow A(x) \quad \Rightarrow \quad A(dp)$

$A^{\mathfrak{M}} = \{a, \qquad \}$ $\qquad E_1P^{-\mathfrak{M}} = \{a', dp^-, \qquad \}$

$E_1P^{\mathfrak{M}} = \{a, dp,$

DL-Lite$^{\mathcal{N}}_{bool}$ is NP-complete – FO Model

$\mathcal{T} = \{A \equiv \exists P^-,\ A \sqsubseteq\ \geq 2\, P,\ \top \sqsubseteq\ \leq 1\, P^-,\ \exists P \sqsubseteq A\}$, $\mathcal{A} = \{A(a),\ P(a, a')\}$

$\mathcal{T}^* = \big(A(x) \leftrightarrow E_1 P^-(x)\big) \wedge \big(A(x) \rightarrow E_2 P(x)\big) \wedge \neg E_2 P^-(x) \wedge \big(E_1 P(x) \rightarrow A(x)\big)$

$$R^\dagger :\ Paa' \quad \Rightarrow \quad P^- a' a,\ E_1 P(a)\ \Rightarrow\ E_1 P^-(a')$$

$$\varepsilon(R) :\ E_1 P(a),\ E_1 P^-(a') \quad \Rightarrow \quad E_1 P^-(dp^-),\ E_1 P(dp)$$

$$\mathcal{T}^* :\ E_1 P(x) \rightarrow A(x) \quad \Rightarrow \quad A(dp)$$

$$\mathcal{T}^* :\ A(x) \leftrightarrow E_1 P^-(x) \quad \Rightarrow \quad E_1 P^-(a),\ E_1 P^-(dp),\ A(a'),\ A(dp^-)$$

$A^{\mathfrak{M}} = \{a, dp, \qquad \}$ $\qquad E_1 P^{-\mathfrak{M}} = \{a', dp^-, \qquad \}$

$E_1 P^{\mathfrak{M}} = \{a, dp,$

DL-Lite$^{\mathcal{N}}_{bool}$ is NP-complete – FO Model

$\mathcal{T} = \{A \equiv \exists P^-,\ A \sqsubseteq\ \geq 2\,P,\ \top \sqsubseteq\ \leq 1\,P^-,\ \exists P \sqsubseteq A\}$, $\mathcal{A} = \{A(a),\ P(a,a')\}$

$\mathcal{T}^* = \big(A(x) \leftrightarrow E_1P^-(x)\big) \wedge \big(A(x) \rightarrow E_2P(x)\big) \wedge \neg E_2P^-(x) \wedge \big(E_1P(x) \rightarrow A(x)\big)$

$$R^\dagger :\ Paa' \quad\Rightarrow\quad P^-a'a,\ E_1P(a) \ \Rightarrow\ E_1P^-(a')$$

$$\varepsilon(R) :\ E_1P(a),\ E_1P^-(a') \quad\Rightarrow\quad E_1P^-(dp^-),\ E_1P(dp)$$

$$\mathcal{T}^* :\ E_1P(x) \rightarrow A(x) \quad\Rightarrow\quad A(dp)$$

$$\mathcal{T}^* :\ A(x) \leftrightarrow E_1P^-(x) \quad\Rightarrow\quad E_1P^-(a),\ E_1P^-(dp),\ A(a'),\ A(dp^-)$$

$$\mathcal{T}^* :\ A(x) \rightarrow E_2P(x) \quad\Rightarrow\quad E_2P^{\mathfrak{M}} = A^{\mathfrak{M}}$$

$A^{\mathfrak{M}} = \{a, dp, a', dp^-\}$ $\qquad$ $E_1P^{-\mathfrak{M}} = \{a', dp^-, a, dp\}$

$E_1P^{\mathfrak{M}} = \{a, dp,$

$DL\text{-}Lite^{\mathcal{N}}_{bool}$ is NP-complete – FO Model

$\mathcal{T} = \{A \equiv \exists P^-,\ A \sqsubseteq\ \geq 2\,P,\ \top \sqsubseteq\ \leq 1\,P^-,\ \exists P \sqsubseteq A\}$, $\mathcal{A} = \{A(a),\ P(a,a')\}$

$\mathcal{T}^* = \big(A(x) \leftrightarrow E_1P^-(x)\big) \wedge \big(A(x) \rightarrow E_2P(x)\big) \wedge \neg E_2P^-(x) \wedge \big(E_1P(x) \rightarrow A(x)\big)$

$$
\begin{array}{lll}
R^\dagger:\ Paa' & \Rightarrow & P^-a'a,\ E_1P(a)\ \Rightarrow\ E_1P^-(a') \\
\varepsilon(R):\ E_1P(a),\ E_1P^-(a') & \Rightarrow & E_1P^-(dp^-),\ E_1P(dp) \\
\mathcal{T}^*:\ E_1P(x) \rightarrow A(x) & \Rightarrow & A(dp) \\
\mathcal{T}^*:\ A(x) \leftrightarrow E_1P^-(x) & \Rightarrow & E_1P^-(a),\ E_1P^-(dp),\ A(a'),\ A(dp^-) \\
\mathcal{T}^*:\ A(x) \rightarrow E_2P(x) & \Rightarrow & E_2P^{\mathfrak{M}} = A^{\mathfrak{M}} \\
\delta(R):\ E_2P(x) \rightarrow E_1P(x) & \Rightarrow & E_1P^{\mathfrak{M}} = E_2P^{\mathfrak{M}}
\end{array}
$$

$A^{\mathfrak{M}} = \{a, dp, a', dp^-\} = E_2P^{\mathfrak{M}}$ $\quad$ $E_1P^{-\mathfrak{M}} = \{a', dp^-, a, dp\}$

$E_1P^{\mathfrak{M}} = \{a, dp,$

DL-Lite$^{\mathcal{N}}_{bool}$ is NP-complete – FO Model

$\mathcal{T} = \{A \equiv \exists P^-,\ A \sqsubseteq\ \geq 2\,P,\ \top \sqsubseteq\ \leq 1\,P^-,\ \exists P \sqsubseteq A\}$, $\mathcal{A} = \{A(a),\ P(a,a')\}$

$\mathcal{T}^* = \big(A(x) \leftrightarrow E_1P^-(x)\big) \wedge \big(A(x) \rightarrow E_2P(x)\big) \wedge \neg E_2P^-(x) \wedge \big(E_1P(x) \rightarrow A(x)\big)$

$$
\begin{array}{lcl}
R^\dagger :\ Paa' & \Rightarrow & P^-a'a,\ E_1P(a)\ \Rightarrow\ E_1P^-(a') \\
\varepsilon(R) :\ E_1P(a),\ E_1P^-(a') & \Rightarrow & E_1P^-(dp^-),\ E_1P(dp) \\
\mathcal{T}^* :\ E_1P(x) \rightarrow A(x) & \Rightarrow & A(dp) \\
\mathcal{T}^* :\ A(x) \leftrightarrow E_1P^-(x) & \Rightarrow & E_1P^-(a),\ E_1P^-(dp),\ A(a'),\ A(dp^-) \\
\mathcal{T}^* :\ A(x) \rightarrow E_2P(x) & \Rightarrow & E_2P^{\mathfrak{M}} = A^{\mathfrak{M}} \\
\delta(R) :\ E_2P(x) \rightarrow E_1P(x) & \Rightarrow & E_1P^{\mathfrak{M}} = E_2P^{\mathfrak{M}}
\end{array}
$$

$A^{\mathfrak{M}} = \{a, dp, a', dp^-\} = E_2P^{\mathfrak{M}} = D$, $\quad E_1P^{-\mathfrak{M}} = \{a', dp^-, a, dp\} = D$,
$E_1P^{\mathfrak{M}} = \{a, dp, a', dp^-\} = D.$

DL-Lite$^{\mathcal{N}}_{bool}$ is NP-complete – Upper Bound – Lemma 1

Lemma 1. A *DL-Lite*$^{\mathcal{N}}_{bool}$ KB $\mathcal{K}$ is satisfiable iff the $\mathcal{QL}^1$-sentence $\mathcal{K}^\dagger$ is satisfiable.

($\Leftarrow$) Starting from a model $\mathfrak{M} = (D, \cdot^{\mathfrak{M}})$, with $D = ob(\mathcal{A}) \cup dr(\mathcal{K})$, of $\mathcal{K}^\dagger$, we construct an interpretation $\mathcal{I}$ for *DL-Lite*$^{\mathcal{N}}_{bool}$ based on some domain $\Delta \supseteq D$ inductively defined as:

$$\Delta = \bigcup_{m=0}^{\infty} W_m, \qquad \text{where} \quad W_0 = D = \textit{constants in } \mathcal{K}^\dagger$$

DL-Lite$^{\mathcal{N}}_{bool}$ is NP-complete – Upper Bound – Lemma 1

Lemma 1. A *DL-Lite*$^{\mathcal{N}}_{bool}$ KB $\mathcal{K}$ is satisfiable iff the $\mathcal{QL}^1$-sentence $\mathcal{K}^\dagger$ is satisfiable.

($\Leftarrow$) Starting from a model $\mathfrak{M} = (D, \cdot^{\mathfrak{M}})$, with $D = ob(\mathcal{A}) \cup dr(\mathcal{K})$, of $\mathcal{K}^\dagger$, we construct an interpretation $\mathcal{I}$ for *DL-Lite*$^{\mathcal{N}}_{bool}$ based on some domain $\Delta \supseteq D$ inductively defined as:

$$\Delta = \bigcup_{m=0}^{\infty} W_m, \qquad \text{where} \quad W_0 = D = \textit{constants in } \mathcal{K}^\dagger$$

Each set W_{m+1}, for $m \geq 0$, is constructed by adding to W_m some new elements, w', that are fresh copies of certain elements, d, from $W_0 = D$, i.e., $cp(w') = d$.

$$A^{\mathcal{I}} = \{w \in \Delta \mid \mathfrak{M} \models A[cp(w)]\}$$

$$P_k^{\mathcal{I}} = \bigcup_{m=0}^{\infty} P_k^m, \qquad \text{where } P_k^m \subseteq W_m \times W_m$$

For the basis of induction we set, for each role P_k:

$$P^0 = \{(a_i^{\mathfrak{M}}, a_j^{\mathfrak{M}}) \in W_0 \times W_0 \mid \mathfrak{M} \models Pa_ia_j\}.$$

DL-Lite$^{\mathcal{N}}_{bool}$ is NP-complete – Upper Bound – Lemma 1 (cont.)

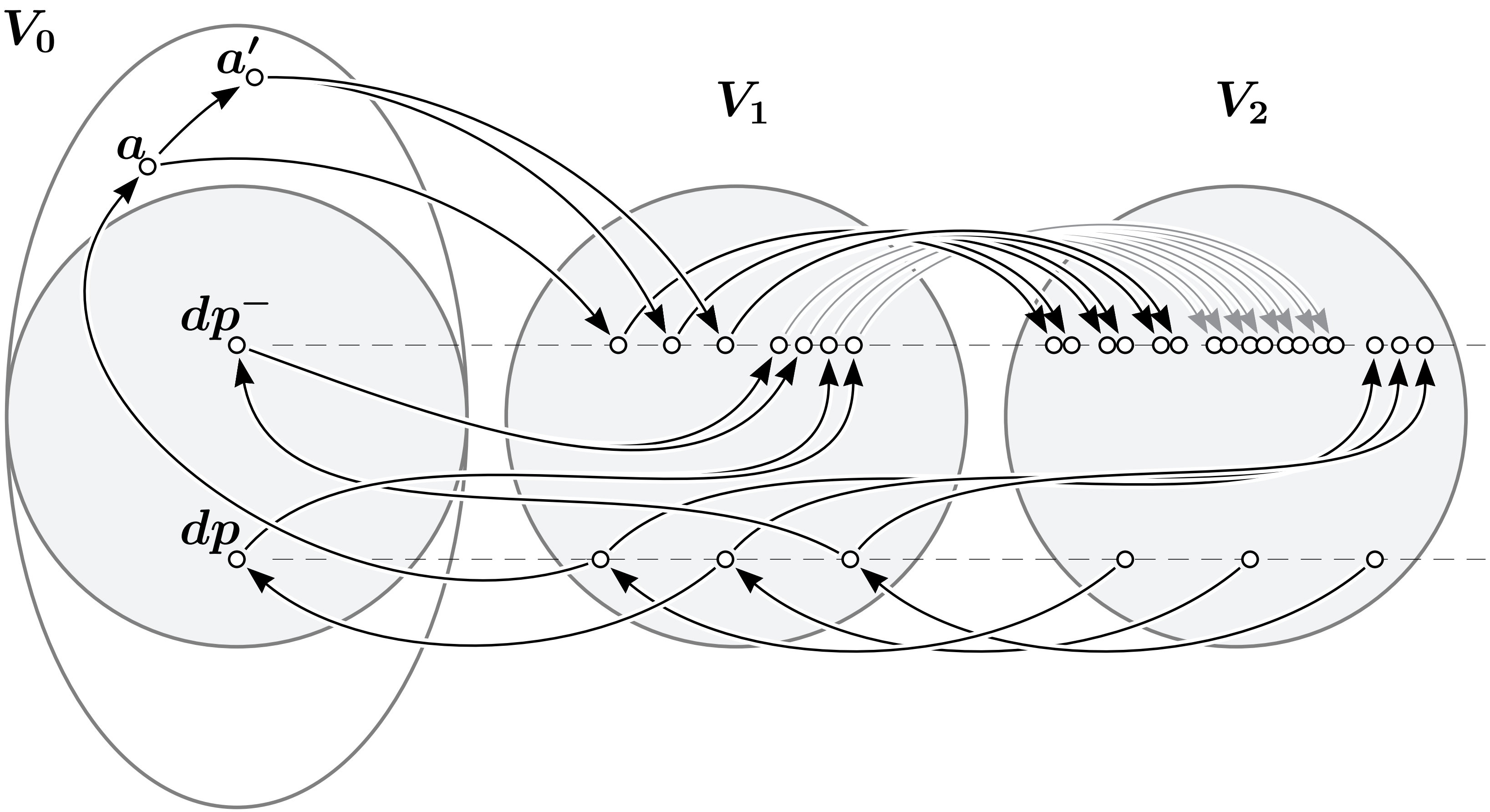


Figure 1: Unravelling model $\mathfrak{M}$ (first three steps).

DL-Lite$^{\mathcal{N}}_{bool}$ is NP-complete – Upper Bound – Translation $\mathcal{K}^\ddagger$

- Translation of $\mathcal{K}$=(TBox,ABox): The lengthy vs. short translation $\mathcal{K}^\ddagger$.

$$\mathcal{K}^\dagger = \Big[\mathcal{T}^* \wedge \bigwedge_{R \in \textit{role}^\pm(\mathcal{K})} \big(\varepsilon(R) \wedge \delta(R)\big)\Big] \wedge \Big[\mathcal{A}^\dagger \wedge \bigwedge_{R \in \textit{role}^\pm(\mathcal{K})} R^\dagger\Big]$$

$$\mathcal{K}^\ddagger = \Big[\mathcal{T}^* \wedge \bigwedge_{R \in \textit{role}^\pm(\mathcal{K})} \big(\varepsilon(R) \wedge \delta^\ddagger(R)\big)\Big] \wedge \mathcal{A}^\ddagger \wedge \mathcal{A}^\perp$$

$DL\text{-}Lite^{\mathcal{N}}_{bool}$ is NP-complete – Upper Bound – Translation $\mathcal{K}^{\ddagger}$

- Translation of $\mathcal{K}$=(TBox,ABox): The lengthy vs. short translation $\mathcal{K}^{\ddagger}$.

$$\mathcal{K}^{\dagger} = \Big[\mathcal{T}^{*} \wedge \bigwedge_{R \in role^{\pm}(\mathcal{K})} \big(\varepsilon(R) \wedge \delta(R)\big)\Big] \wedge \Big[\mathcal{A}^{\dagger} \wedge \bigwedge_{R \in role^{\pm}(\mathcal{K})} R^{\dagger}\Big]$$

$$\mathcal{K}^{\ddagger} = \Big[\mathcal{T}^{*} \wedge \bigwedge_{R \in role^{\pm}(\mathcal{K})} \big(\varepsilon(R) \wedge \delta^{\ddagger}(R)\big)\Big] \wedge \mathcal{A}^{\ddagger} \wedge \mathcal{A}^{\perp}$$

$$\delta^{\ddagger}_{R}(x) = \bigwedge_{\substack{q,q' \in Q^{R}_{\mathcal{T}},\ q' > q \\ q' > q'' > q \text{ for no } q'' \in Q^{R}_{\mathcal{T}}}} \big(E_{q'}R(x) \rightarrow E_{q}R(x)\big)$$

DL-Lite$^{\mathcal{N}}_{bool}$ is NP-complete – Upper Bound – Translation $\mathcal{K}^{\ddagger}$

- Translation of $\mathcal{K}$=(TBox,ABox): The lengthy vs. short translation $\mathcal{K}^{\ddagger}$.

$$\mathcal{K}^{\dagger} = \left[\mathcal{T}^{*} \wedge \bigwedge_{R \in \textit{role}^{\pm}(\mathcal{K})} \big(\varepsilon(R) \wedge \delta(R)\big)\right] \wedge \left[\mathcal{A}^{\dagger} \wedge \bigwedge_{R \in \textit{role}^{\pm}(\mathcal{K})} R^{\dagger}\right]$$

$$\mathcal{K}^{\ddagger} = \left[\mathcal{T}^{*} \wedge \bigwedge_{R \in \textit{role}^{\pm}(\mathcal{K})} \big(\varepsilon(R) \wedge \delta^{\ddagger}(R)\big)\right] \wedge \mathcal{A}^{\ddagger} \wedge \mathcal{A}^{\perp}$$

$$\delta^{\ddagger}_{R}(x) = \bigwedge_{\substack{q,q' \in Q^{R}_{\mathcal{T}},\ q'>q \\ q'>q''>q \text{ for no } q'' \in Q^{R}_{\mathcal{T}}}} \big(E_{q'}R(x) \rightarrow E_{q}R(x)\big)$$

$$\mathcal{A}^{\ddagger} = \bigwedge_{A(a) \in \mathcal{A}} A(a) \quad \wedge \quad \bigwedge_{\neg A(a) \in \mathcal{A}} \neg A(a) \quad \wedge$$

$$\bigwedge_{a \in ob(\mathcal{A})} \bigwedge_{\substack{R \in \textit{role}^{\pm}(\mathcal{K}) \\ \exists a' \in ob(\mathcal{A})\ R(a,a') \in \mathcal{A}}} E_{q_{R,a}} R(a)$$

DL-Lite$^{\mathcal{N}}_{bool}$ is NP-complete – Upper Bound – Translation $\mathcal{K}^{\ddagger}$

- Translation of $\mathcal{K}$=(TBox,ABox): The lengthy vs. short translation $\mathcal{K}^{\ddagger}$.

$$\mathcal{K}^{\dagger} = \Big[\mathcal{T}^* \wedge \bigwedge_{R \in \mathit{role}^{\pm}(\mathcal{K})} \big(\varepsilon(R) \wedge \delta(R)\big)\Big] \wedge \Big[\mathcal{A}^{\dagger} \wedge \bigwedge_{R \in \mathit{role}^{\pm}(\mathcal{K})} R^{\dagger}\Big]$$

$$\mathcal{K}^{\ddagger} = \Big[\mathcal{T}^* \wedge \bigwedge_{R \in \mathit{role}^{\pm}(\mathcal{K})} \big(\varepsilon(R) \wedge \delta^{\ddagger}(R)\big)\Big] \wedge \mathcal{A}^{\ddagger} \wedge \mathcal{A}^{\perp}$$

$$\delta^{\ddagger}_R(x) = \bigwedge_{\substack{q,q' \in Q^R_{\mathcal{T}},\ q' > q \\ q' > q'' > q \text{ for no } q'' \in Q^R_{\mathcal{T}}}} \big(E_{q'}R(x) \rightarrow E_q R(x)\big)$$

$$\mathcal{A}^{\ddagger} = \bigwedge_{A(a) \in \mathcal{A}} A(a) \quad \wedge \quad \bigwedge_{\neg A(a) \in \mathcal{A}} \neg A(a) \quad \wedge$$

$$\bigwedge_{a \in ob(\mathcal{A})} \bigwedge_{\substack{R \in \mathit{role}^{\pm}(\mathcal{K}) \\ \exists a' \in ob(\mathcal{A})\ R(a,a') \in \mathcal{A}}} E_{q_{R,a}} R(a)$$

$$\mathcal{A}^{\perp} = \perp \ \text{ if } \ P(a_i, a_j) \in \mathcal{A} \text{ and } \neg P(a_i, a_j) \in \mathcal{A}$$

DL-Lite$^{\mathcal{N}}_{bool}$ is NP-complete – Upper Bound – Translation $\mathcal{K}^{\ddagger}$

- Translation of $\mathcal{K}$=(TBox,ABox): The lengthy vs. short translation $\mathcal{K}^{\ddagger}$.

$$\mathcal{K}^{\dagger} = \Big[\mathcal{T}^{*} \wedge \bigwedge_{R \in \textit{role}^{\pm}(\mathcal{K})} \big(\varepsilon(R) \wedge \delta(R)\big)\Big] \wedge \Big[\mathcal{A}^{\dagger} \wedge \bigwedge_{R \in \textit{role}^{\pm}(\mathcal{K})} R^{\dagger}\Big]$$

$$\mathcal{K}^{\ddagger} = \Big[\mathcal{T}^{*} \wedge \bigwedge_{R \in \textit{role}^{\pm}(\mathcal{K})} \big(\varepsilon(R) \wedge \delta^{\ddagger}(R)\big)\Big] \wedge \mathcal{A}^{\ddagger} \wedge \mathcal{A}^{\perp}$$

$$\delta^{\ddagger}_{R}(x) = \bigwedge_{\substack{q,q' \in Q^{R}_{\mathcal{T}},\ q' > q \\ q' > q'' > q \text{ for no } q'' \in Q^{R}_{\mathcal{T}}}} \big(E_{q'}R(x) \rightarrow E_{q}R(x)\big)$$

$$\mathcal{A}^{\ddagger} = \bigwedge_{A(a) \in \mathcal{A}} A(a) \wedge \bigwedge_{\neg A(a) \in \mathcal{A}} \neg A(a) \wedge$$

$$\bigwedge_{a \in \textit{ob}(\mathcal{A})} \bigwedge_{\substack{R \in \textit{role}^{\pm}(\mathcal{K}) \\ \exists a' \in \textit{ob}(\mathcal{A})\ R(a,a') \in \mathcal{A}}} E_{q_{R,a}} R(a)$$

$$\mathcal{A}^{\perp} = \perp \text{ if } P(a_i, a_j) \in \mathcal{A} \text{ and } \neg P(a_i, a_j) \in \mathcal{A}$$

$\mathcal{K}^{\ddagger}$ can be computed in LOGSPACE.

Satifiability Checking – Combined Complexity Results

Theorem

1. The satisfiability problem for *DL-Lite*$^{\mathcal{N}}_{bool}$, *DL-Lite*$^{\mathcal{F}}_{bool}$ and *DL-Lite*$_{bool}$ knowledge bases is NP-complete.

Satifiability Checking – Combined Complexity Results

Theorem

1. The satisfiability problem for $\textit{DL-Lite}^{\mathcal{N}}_{bool}$, $\textit{DL-Lite}^{\mathcal{F}}_{bool}$ and $\textit{DL-Lite}_{bool}$ knowledge bases is NP-complete.
2. The satisfiability problem for $\textit{DL-Lite}^{\mathcal{N}}_{krom}$, $\textit{DL-Lite}^{\mathcal{F}}_{krom}$, $\textit{DL-Lite}_{krom}$, as well as $\textit{DL-Lite}^{\mathcal{N}}_{core}$, $\textit{DL-Lite}^{\mathcal{F}}_{core}$ and $\textit{DL-Lite}_{core}$ knowledge bases, is NLOGSPACE-complete.

Satifiability Checking – Combined Complexity Results

Theorem

1. The satisfiability problem for $DL\text{-}Lite^{\mathcal{N}}_{bool}$, $DL\text{-}Lite^{\mathcal{F}}_{bool}$ and $DL\text{-}Lite_{bool}$ knowledge bases is NP-complete.
2. The satisfiability problem for $DL\text{-}Lite^{\mathcal{N}}_{krom}$, $DL\text{-}Lite^{\mathcal{F}}_{krom}$, $DL\text{-}Lite_{krom}$, as well as $DL\text{-}Lite^{\mathcal{N}}_{core}$, $DL\text{-}Lite^{\mathcal{F}}_{core}$ and $DL\text{-}Lite_{core}$ knowledge bases, is NLOGSPACE-complete.
3. The satisfiability problem for $DL\text{-}Lite^{\mathcal{N}}_{horn}$, $DL\text{-}Lite^{\mathcal{F}}_{horn}$ and $DL\text{-}Lite_{horn}$ knowledge bases is P-complete.

Sub-Roles
Vs
Cardinality Constraints

DL-Lite$_{horn}^{\mathcal{R},\mathcal{F}}$ is EXPTIME-complete

DL-Lite$_{horn}^{\mathcal{R},\mathcal{F}}$ Ontology language:

- Role Inclusions: $R_1 \sqsubseteq R_2$,
- Functionality Axioms: $\geq 2\ R \sqsubseteq \bot$
- Concept Inclusions: $\bigsqcap_k B_k \sqsubseteq B$, with:

$$B \longrightarrow A \mid \exists R \mid \bot$$

$$R \longrightarrow P \mid P^-$$

- ABox assertions: $A(c)$, $\neg A(c)$ $P(c,d)$, $\neg P(c,d)$ with c, d constants

DL-Lite$_{horn}^{\mathcal{R},\mathcal{F}}$ is ExpTime-complete

DL-Lite$_{horn}^{\mathcal{R},\mathcal{F}}$ Ontology language:

- Role Inclusions: $R_1 \sqsubseteq R_2$,
- Functionality Axioms: $\geq 2\ R \sqsubseteq \bot$
- Concept Inclusions: $\bigsqcap_k B_k \sqsubseteq B$, with:

$$B \longrightarrow A \mid \exists R \mid \bot$$
$$R \longrightarrow P \mid P^-$$

- ABox assertions: $A(c)$, $\neg A(c)$ $P(c, d)$, $\neg P(c, d)$ with c, d constants
- Upper Bound: *DL-Lite*$_{horn}^{\mathcal{R},\mathcal{F}}$ is a sub-language of $\mathcal{SHIQ}$ which is ExpTime-complete.
- Lower Bound: *DL-Lite*$_{horn}^{\mathcal{R},\mathcal{F}}$ KBs can encode the behaviour of polynomial-space-bounded alternating Turing machines (ATMs).

DL-Lite$^{\mathcal{R},\mathcal{F}}_{core}$ is EXPTIME-complete

The only difference between *DL-Lite*$^{\mathcal{R},\mathcal{F}}_{core}$ and *DL-Lite*$^{\mathcal{R},\mathcal{F}}_{horn}$ is the possibility to express conjunction on the left of axioms in *DL-Lite*$^{\mathcal{R},\mathcal{F}}_{horn}$.

Elimination of axioms of the form $A_1 \sqcap A_2 \sqsubseteq C$. Define a new KB $\mathcal{K}'$ by replacing this axiom in $\mathcal{K}$ with the following set of new axioms, where $R_1, R_2, R_3, R_{12}, R_{23}$ are fresh role names:

$$A_1 \sqsubseteq \exists R_1 \qquad A_2 \sqsubseteq \exists R_2, \tag{1}$$

$$R_1 \sqsubseteq R_{12}, \qquad R_2 \sqsubseteq R_{12}, \tag{2}$$

$$\geq 2\, R_{12} \sqsubseteq \bot, \tag{3}$$

$$\exists R_1^- \sqsubseteq \exists R_3^-, \tag{4}$$

$$\exists R_3 \sqsubseteq C, \tag{5}$$

$$R_3 \sqsubseteq R_{23}, \qquad R_2 \sqsubseteq R_{23}, \tag{6}$$

$$\geq 2\, R_{23}^- \sqsubseteq \bot. \tag{7}$$

Cardinality + Sub-Roles: Regaining Tractability

- TBox assertions: $C_1 \sqsubseteq C_2$, $R_1 \sqsubseteq R_2$

Definition 1 (Relaxing cardinality constraints) *Given a TBox $\mathcal{T}$ and a role $R \in role^{\pm}(\mathcal{T})$, we define the following parameters:*

$$\begin{aligned}
\textit{ubound}(R, \mathcal{T}) &= \min\big(\{\infty\} \cup \{q \mid C \sqsubseteq (\leq q\, R) \in \mathcal{T}\}\big)\\
\textit{lbound}(R, \mathcal{T}) &= \max\big(\{0\} \cup \{q \mid C \sqsubseteq (\geq q\, R) \in \mathcal{T}\}\big)\\
\textit{rank}(R, \mathcal{T}) &= \max\big(\textit{lbound}(R, \mathcal{T}), \textstyle\sum_{R' \in \textit{dsub}_{\mathcal{T}}(R)} \textit{rank}(R', \mathcal{T})\big)\\
\textit{rank}(R, \mathcal{A}) &= \max\big(\{0\} \cup \{n \mid R_i(a, a_j) \in \mathcal{A}, R_i \sqsubseteq^*_{\mathcal{T}} R,\\
&\qquad \textit{for distinct } a_1, \ldots, a_n\}\big)
\end{aligned}$$

(inter1) If R has a proper sub-role in $\mathcal{T}$ then the TBox contains no at-most cardinality restrictions on R.

(inter2) If R has a proper sub-role in $\mathcal{T}$ then

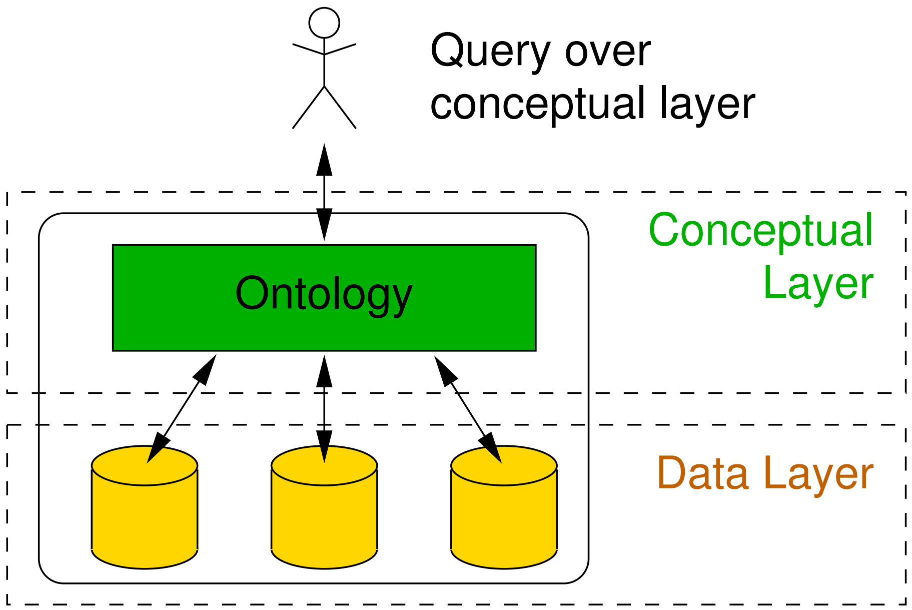
$$\textit{ubound}(R, \mathcal{T}) \geq \textit{rank}(R, \mathcal{T}) + \max\{1, \textit{rank}(R, \mathcal{A})\}$$

Query Language

Ontology based data access

Desiderata: achieve logical transparency in access to data:

- Hide to the user where and how data are stored
- Present to the user a conceptual view of the data
- Query the data sources through the conceptual model



As in Data Integration, but with a rich conceptual description as the global view

Query Language

We consider positive existential queries—extending UCQs with unrestricted interaction of conjunction and disjunction—over the terms of the ontology:

$$
\begin{array}{lcl}
t & ::= & y_i \mid a_i \\
q & ::= & A_k(t) \mid P_k(t_1, t_2) \mid q_1 \wedge q_2 \mid q_1 \vee q_2 \mid \exists y_i\, q
\end{array}
$$

Example:

$$
\begin{array}{l}
q(x) = \{\, x \mid \exists y, p.\ \mathsf{Employee}(x) \wedge \mathsf{WorksFor}(x, p) \wedge \mathsf{Project}(p) \wedge \\
\qquad \mathsf{Boss}(y, x) \wedge \mathsf{Employee}(y) \wedge \mathsf{WorksFor}(y, p) \,\}
\end{array}
$$

Query Answering

Since we work under the Open World Semantics then Query answering over an ontology $\mathcal{O}$ wrt an ABox $\mathcal{A}$ amounts to computing certain answers:

$$cert(q, \mathcal{O}, \mathcal{A}) = \{ \vec{t} \mid \vec{t} \in q^{\mathcal{I}} \text{ for every } \mathcal{I} \in \textit{mod}(\mathcal{O}, \mathcal{A}) \}$$

i.e., the tuples that are answers to q in all models of the ABox $\mathcal{A}$ w.r.t. the ontology $\mathcal{O}$.

Query Answering

Since we work under the Open World Semantics then Query answering over an ontology $\mathcal{O}$ wrt an ABox $\mathcal{A}$ amounts to computing certain answers:

$$cert(q, \mathcal{O}, \mathcal{A}) \;=\; \{\, \vec{t} \mid \vec{t} \in q^{\mathcal{I}} \text{ for every } \mathcal{I} \in \textit{mod}(\mathcal{O}, \mathcal{A}) \,\}$$

i.e., the tuples that are answers to q in all models of the ABox $\mathcal{A}$ w.r.t. the ontology $\mathcal{O}$.

Computing certain answers is a form of logical implication:

$$\vec{t} \in cert(q, \mathcal{O}, \mathcal{A}) \qquad \text{iff} \qquad (\mathcal{O}, \mathcal{A}) \models q(\vec{t})$$

Data complexity Vs. Combined complexity

When considering a setting where the size of the data largely dominates the size of the conceptual layer $\leadsto$ We look at data complexity

When both the size of the ontology and the size of the underlying data are comparable $\leadsto$ We look at combined complexity

Data complexity Vs. Combined complexity

When considering a setting where the size of the data largely dominates the size of the conceptual layer $\leadsto$ We look at data complexity

When both the size of the ontology and the size of the underlying data are comparable $\leadsto$ We look at combined complexity

Basic questions:

- How complex becomes reasoning over both an ontology and a data source? (both data and combined complexity)
  - In particular, for which ontology language can we answer queries to DB sources through an ontology efficiently? (data complexity)

Data Complexity of Query Answering in *DL-Lite*$^{\mathcal{N}}_{horn}$ is in LOGSPACE

Given a positive existential query, $q(\vec{x})$, and a *DL-Lite*$^{\mathcal{N}}_{horn}$ KB, $\mathcal{T}$, then:

1. First, we construct a single, but possibly infinite, model $\mathcal{I}_0$ which provides all answers to all positive existential queries with respect to the *DL-Lite*$^{\mathcal{N}}_{horn}$ KB, $\mathcal{K}$:
   - The canonical model $\mathcal{I}_0$ is obtained starting from the minimal Herbrand model for $\mathcal{K}^{\ddagger}$.
2. Second, to find all answers to a given query it is enough to consider some finite part of $\mathcal{I}_0$ the size of which does not depend on the given ABox.
   - Assume that, in the query $\mathbf{q}(\vec{x}) = \exists \vec{y}\, \varphi(\vec{x}, \vec{y})$, we have $\vec{y} = y_1, \ldots, y_k$, then, to check whether $\mathcal{I}_0 \models \mathbf{q}(\vec{a})$ it suffices to consider only the points of depth $\leq m_0$ in $\Delta^{\mathcal{I}_0}$, with $m_0 = k + |\mathit{role}^{\pm}(\mathcal{T})|$.

Data Complexity of Query Answering in *DL-Lite*$^{\mathcal{N}}_{horn}$ is in LOGSPACE

- The LOGSPACE query answering algorithm will consider then all assignments in this finite part of $\mathcal{I}_0$ to the variables $\vec{x}$, $\vec{y}$, compute the corresponding types (the concepts that contain these elements), and, finally, encode the problem '$\mathcal{K} \models \mathbf{q}(\vec{a})$?' as a model checking problem for the first-order formula $\varphi_{\mathcal{T},\mathbf{q}}(\vec{x})$:
  - $\varphi_{\mathcal{T},\mathbf{q}}(\vec{x})$ depends on $\mathcal{T}$ and $\mathbf{q}$ but not on $\mathcal{A}$,
  - $\mathcal{A} \models \varphi_{\mathcal{T},\mathbf{q}}(\vec{a})$ iff $\mathcal{I}_0 \models \mathbf{q}(\vec{a})$.

Data Complexity of Query Answering in *DL-Lite*$^{\mathcal{N}}_{horn}$ is in LogSpace

Formulas $\psi_B(x)$, for $B \in$ *Bcon*$(\mathcal{K})$, describe the types of the elements of *ob*$(\mathcal{A})$ in the model $\mathcal{I}_0$:

$$\mathcal{A} \models \psi_B[a_i] \text{ iff } a_i^{\mathcal{I}_0} \in B^{\mathcal{I}_0}, \text{ for } a_i \in ob(\mathcal{A})$$

$$\psi_B^0(x) = \begin{cases} A(x), & \text{if } B = A, \\ E_q R^{\mathcal{T}}(x), & \text{if } B = \geq q\, R \end{cases}$$

$$\psi_B^i(x) = \psi_B^0(x) \vee \bigvee_{B_1 \sqcap \cdots \sqcap B_k \sqsubseteq B \in \text{ext}(\mathcal{T})} \left(\psi_{B_1}^{i-1}(x) \wedge \cdots \wedge \psi_{B_k}^{i-1}(x)\right), \quad \text{for } i \geq 1$$

$$E_q R^{\mathcal{T}}(x) = \exists y_1 \ldots \exists y_q \left(\bigwedge_{1 \leq i < j \leq q} (y_i \neq y_j) \wedge \bigwedge_{1 \leq i \leq q} R(x, y_i) \right)$$

Data Complexity of Query Answering in *DL-Lite*$^{\mathcal{N}}_{horn}$ is in LOGSPACE

Formulas $\theta_{B,dr}$, for $B \in$ *Bcon*$(\mathcal{K})$ and $dr \in$ *dr*$(\mathcal{T})$, describe the types of elements *dr*$(\mathcal{T})$ in the model $\mathcal{I}_0$:

$$\mathcal{A} \models \theta_{B,dr} \text{ iff } w \in B^{\mathcal{I}_0}, \text{ for } w \in \Delta^{\mathcal{I}_0} \text{ with } \mathit{cp}(w) = dr.$$

For each $B \in$ *Bcon*$(\mathcal{K})$ and each $dr \in$ *dr*$(\mathcal{K})$, we inductively define a sequence $\theta^0_{B,dr}, \theta^1_{B,dr}, \ldots$ by taking:

$$\theta^0_{B,dr} = \top, \textit{ if } B = \exists R, \textit{ and } \theta^0_{B,dr} = \bot, \textit{ otherwise}$$

$$\theta^i_{B,dr} = \theta^{i-1}_{B,dr} \vee \bigvee_{B_1 \sqcap \cdots \sqcap B_k \sqsubseteq B \in \mathsf{ext}(\mathcal{T})} \left(\theta^{i-1}_{B_1,dr} \wedge \cdots \wedge \theta^{i-1}_{B_k,dr}\right), \text{ for } i \geq 1$$

Data Complexity of Query Answering in *DL-Lite*$^{\mathcal{N}}_{horn}$ is in LOGSPACE

$$\text{path} = \{(R_i, R_j) \mid \mathcal{T} \models \exists \mathit{inv}(R_i) \sqsubseteq \geq q\, R_j\}$$

$$\Sigma_{\mathcal{T},m_0} = \{\varepsilon\} \cup \mathit{role}^{\pm}(\mathcal{K}) \cup \{(R_1, \ldots, R_n) \mid 2 \leq n \leq m_0, (R_j, R_{j+1}) \in \text{path}\}$$

- Every existential variable, y_i, is evaluated using $\sigma \in \Sigma_{\mathcal{T},m_0}$;
- If $\sigma = \epsilon$, then y_i is assigned to an ABox element;
- $\sigma \neq \epsilon$, then y_i is evaluated as w using the pair (a, σ) s.t. $a^{\mathcal{I}_0}$ is the root of the tree $\mathfrak{T}_a$ containing w, and σ is the sequence of roles on the path from $a^{\mathcal{I}_0}$ to w.

For $\sigma \neq \epsilon$ s.t. $\sigma = (R_i, \ldots)$, this formula ensures that path σ exists in $\mathcal{I}_0$:

$$\eta^{\sigma}(a) \quad = \quad \bigvee_{q \in Q^{R_i}_{\mathcal{T}}} \left(\psi_{\geq q\, R_i}(a) \wedge \neg \psi^{0}_{\geq q\, R_i}(a)\right)$$

Data Complexity of Query Answering in *DL-Lite*$^{\mathcal{N}}_{horn}$ is in LOGSPACE

Let $\mathbf{q}(\vec{x}) = \exists y_1, \ldots, y_k . \varphi(\vec{x}, y_1, \ldots, y_k)$, then for every $\vec{\sigma} \in \Sigma^k_{\mathcal{T},m_0}$, concept name A and role name R, we define:

$$A^{\vec{\sigma}}(t) = \begin{cases} \psi_A(t), & \text{if } t^{\vec{\sigma}} = \varepsilon, \\ \theta_{A,\mathit{inv}(ds)}, & \text{if } t^{\vec{\sigma}} = \sigma'.[S], \text{ for some } \sigma' \in \Sigma_{\mathcal{T},m_0}, \end{cases}$$

$$R^{\vec{\sigma}}(t_1, t_2) = \begin{cases} R^{\mathcal{T}}(t_1, t_2), & \text{if } t_1^{\vec{\sigma}} = t_2^{\vec{\sigma}} = \varepsilon, \\ (t_1 = t_2), & \text{if } t_1^{\vec{\sigma}} \xrightarrow{R} t_2^{\vec{\sigma}} \text{ and either } t_1^{\vec{\sigma}} \neq \varepsilon \text{ or } t_2^{\vec{\sigma}} \neq \varepsilon, \\ \bot, & \text{otherwise.} \end{cases}$$

The first-order rewriting of $q(\vec{x})$ is then:

$$\varphi_{\mathcal{T},\mathbf{q}}(\vec{x}) = \exists \vec{y} \bigvee_{\vec{\sigma} \in \Sigma^k_{\mathcal{T},m_0}} (\varphi^{\vec{\sigma}}(\vec{x}, \vec{y}) \wedge \eta^{\vec{\sigma}}(\vec{y}))$$

language	complexity		
	combined	data	
	satisfiability	inst. checking	query answering
DL-Lite$_{core}$	NLogSpace $\geq$ [Log]	in LogSpace	in LogSpace
DL-Lite$^{\mathcal{F}}_{core}$	NLogSpace	in LogSpace	in LogSpace
DL-Lite$^{\mathcal{N}}_{core}$	NLogSpace	in LogSpace	in LogSpace
DL-Lite$^{\mathcal{R}}_{core}$	NLogSpace	in LogSpace	in LogSpace
DL-Lite$^{\mathcal{R},\mathcal{F}}_{core}$	ExpTime $\geq$	P $\geq$	P
DL-Lite$^{\mathcal{R},\mathcal{N}}_{core}$	ExpTime	coNP $\geq$	coNP
DL-Lite$_{krom}$	NLogSpace	in LogSpace	coNP $\geq$ [B]
DL-Lite$^{\mathcal{F}}_{krom}$	NLogSpace	in LogSpace	coNP
DL-Lite$^{\mathcal{N}}_{krom}$	NLogSpace $\leq$	in LogSpace	coNP
DL-Lite$^{\mathcal{R}}_{krom}$	NLogSpace $\leq$	in LogSpace	coNP
DL-Lite$^{\mathcal{R},\mathcal{F}}_{krom}$	ExpTime	coNP $\geq$	coNP
DL-Lite$^{\mathcal{R},\mathcal{N}}_{krom}$	ExpTime	coNP	coNP

language	complexity		
	combined	data	
	satisfiability	inst. checking	query answering
$DL\text{-}Lite_{horn}$	P ≥ [Log]	in LOGSPACE	in LOGSPACE
$DL\text{-}Lite_{horn}^{\mathcal{F}}$	P	in LOGSPACE	in LOGSPACE
$DL\text{-}Lite_{horn}^{\mathcal{N}}$	P ≤	in LOGSPACE	in LOGSPACE ≤
$DL\text{-}Lite_{horn}^{\mathcal{R}}$	P ≤	in LOGSPACE	in LOGSPACE ≤ [C]
$DL\text{-}Lite_{horn}^{\mathcal{R},\mathcal{F}}$	EXPTIME	P	P ≤ [D]
$DL\text{-}Lite_{horn}^{\mathcal{R},\mathcal{N}}$	EXPTIME	coNP	coNP
$DL\text{-}Lite_{bool}$	NP ≥ [Log]	in LOGSPACE	coNP
$DL\text{-}Lite_{bool}^{\mathcal{F}}$	NP	in LOGSPACE	coNP
$DL\text{-}Lite_{bool}^{\mathcal{N}}$	NP ≤	in LOGSPACE ≤	coNP
$DL\text{-}Lite_{bool}^{\mathcal{R}}$	NP ≤	in LOGSPACE ≤	coNP
$DL\text{-}Lite_{bool}^{\mathcal{R},\mathcal{F}}$	EXPTIME	coNP	coNP
$DL\text{-}Lite_{bool}^{\mathcal{R},\mathcal{N}}$	EXPTIME	coNP	coNP ≤ [E]

Conclusions & Ongoing Work

- Ontologies based data access is an important problem we have to consider
- Expressive power of ontology languages heavily influences (data) complexity of query answering
- Reasonable expressiveness in the ontology and efficiency of query answering can be reconciled $\rightsquigarrow$ *DL-Lite*-family

Conclusions & Ongoing Work

- Ontologies based data access is an important problem we have to consider
- Expressive power of ontology languages heavily influences (data) complexity of query answering
- Reasonable expressiveness in the ontology and efficiency of query answering can be reconciled $\leadsto$ *DL-Lite*-family

- The *DL-Lite* DLs do not enjoy the finite model property: What if we want to restrict the attention to finite models only?
- Developing efficient algorithms for answering positive existential queries for the LOGSPACE languages that rely on relational database techniques (i.e., SQL).